1990

# A Dynamic scheduling monitor for a manufacturing process

Gregory Weilnau

Rochester Institute of Technology
School of Computer Science and Technology

# A Dynamic Scheduling Monitor for A Manufacturing Process

by

Gregory P. Weilnau

A thesis, submitted to
The Faculty of the School of Computer Science and Technology,
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science.

Approved by: ______________________________

Professor James Heliotis

______________________________

Professor Guy Johnson

______________________________

Professor Peter G. Anderson

February 13, 1990

Title of Thesis: A DYNAMIC SCHEDULING MONITOR FOR A MANUFACTURING PROCESS

I ______________________ hereby (grant/deny) permission to the Wallace Memorial Library, of RIT, to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

or

I ______________________ prefer to be contacted each time a request for reproduction is made. I can be reached at the following address:

5699 BUFFALO ROAD

CHURCHVILLE, NY 14428

Date: 2/14/90

# Dedication

To my loving wife Patricia, who has given me boundless encouragement over these past five years of graduate school.

# Acknowledgements

I express my gratitude to the following people for their assistance towards the completion of this thesis.

Professor Jim Heliotis and Professor Guy Johnson for their encouragement and guidance.

Bruce Erway and Richard Beswick for their encouragement and proofreading skills.

Joe LaFrance and Patrick Furchill for their timely program evaluations.

# Abstract

Dynamic scheduling is the ability to create an effective schedule based on the current state of a manufacturing factory. I have developed such a scheduler for an injection molding department. The molding department provides the "factory" for which the schedule is created. In order to test my scheduling algorithm, I have developed three modules that form a complete factory model. A user interface allows access to the factory system. A factory database system permits the user to create and maintain the data that is required. The schedule and run module creates a schedule and then, simulates the schedule being run in the factory. The run component also keeps the factory performance data updated in order that subsequent schedules can be calculated based on the current abilities of the factory. The system is implemented on an IBM AT class machine utilizing the C++ language.



## Keywords and Phrases

scheduling, dynamic scheduling, simulation, modeling


## Computing Review Subject Codes

I.6.3 Simulation and Modeling, Applications

J.6 Computer-Aided Manufacturing

# Table of Contents

# Tables and Figures

# Introduction and Background

Dynamic scheduling involves taking the current state of the factory and creating a "best fit" plan to produce a given product requirement in an efficient manner. The plan takes into account any existing inventory of parts so that a minimum of on hand stock is required. This benefit of reduced inventory provides the cost savings that will allow the effective implementation of a dynamic scheduler.

This thesis involves not only the scheduling algorithm, but also the implementation of a factory model with which the algorithm is tested. The factory that is simulated is an automotive injection molding department. It utilizes mold machines and mold tools to produce the various component parts that are assembled into a saleable product. A factory database was designed to maintain all the necessary data on each mold machine, mold tool and product model. The current status of each machine, tool and component part is maintained in a factory status database. A product is comprised of various molded and purchased components that are assembled into automotive emission control devices. The emphasis of this thesis is strictly on the molded components produced in the injection molding area.

The "factory" is comprised of 33 mold machines of three different capacities. A mold machine is a flexible piece of equipment that when tooled with a mold can produce a large quantity of parts in a relatively short period of time. The mold machine capacity is measured by the amount of force that the mold clamp can exert on a mold. The mold machine sizes that are available are 75, 220 and 375 tons. The injection molding process involves heating raw plastic into a liquid state and injecting it at high pressure into a closed mold. The mold is held closed until sufficient cooling time has passed.

When the mold opens the parts are ejected into a collection device, then they are placed in inventory or proceed directly to the assembly area.

A mold is sized by how much clamp force it will require to adequately produce the product. The projected surface area of the molded part is used to calculate the required clamp force. Projected surface area is the area of the molded part that is perpendicular to the clamp force direction. The required clamp force varies proportionally to the projected surface area. A mold must also be physically sized to fit into the appropriate machine. A typical mold for a 75 ton machine would be about one foot square. For a 375 ton machine a mold could be as large as two feet square. The mold tool is typically multi-cavity so that for every machine cycle several parts can be produced. A mold is basically constructed like a large block of steel that requires the use of a hoist when a mold is changed in a machine. Setup of a mold is an important consideration when determining a schedule, since it involves changing molds in a machine before part production can begin. When a mold is changed, it must be brought up to a proper operating temperature of about 150 degrees F. This changeover process is time consuming and has a dramatic effect on the factory output. Some changeovers are less time consuming since they only involve changing small features on a part. These changeovers can be done with the mold in the machine and can be accomplished quickly.

Repair times also contribute significantly to ability to meet a specific schedule. The number of simultaneous repairs and setups are limited to the number of skilled personnel available at any given time. The simulator allows the user to change the number of people available so that studies can be done to determine the appropriate labor force required.

The scheduling algorithm presented here accounts for both setup and repair times when calculating a schedule. The algorithm is designed to utilize the fewest number of machines and molds. This keeps labor requirements at a minimum since the number of simultaneous repairs and setups occurring will be the fewest required. The algorithm can be stated in three basic steps.

1) Break the model based schedule into a component based schedule.

2) Assign the part based schedule into molds with adequate time windows.

3) Schedule the molds into machines with adequate time windows.

## Previous Work

This section contains a synopsis of each of four algorithms that I found during the library search. These descriptions contain the basic assumptions and requirements of each procedure.

The paper presented by Dessouky, Dessouky and Dessouky titled "A Case Study in Parallel Unrelated Machine Scheduling: A Heuristic Approach" lists four goals for their system.

1) To expedite weekly scheduling, presently performed manually.

2) To facilitate sensitivity analysis and schedule revision.

3) To test the impact of expanding the system by adding more products, product families, and lines.

4) To permit synchronizing the scheduling of the packaging operation with that of preceding operations.

The manufacturing process that is studied is a dry foods production and packaging system. The algorithm that is developed utilizes a dual decision path in that the scheduling is chosen in one of two ways. The decision of which method to use is made based on the ability to produce the weekly

requirements within the week time frame. Method one is used if there is enough capacity to produce the volume in the week. This minimizes the number of hours required to produce the weekly volume. If it cannot be done within the week, the second method is used to schedule the maximum amount of product in the given week.

In the paper titled "Knowledge based routing and sequencing for Discrete Part Production" Ben-Arieh and Moodie describe an expert system approach. This concept utilizes real time data generated by the production machinery to assist in planning the production run. The main objective of the system is to maximize the production of assembled parts. This system is aimed at complex manufacturing systems where optimal real-time solutions are not possible. They show that the routing for fabrication and assembly of components is really a tree structure. This leads them to describe three algorithms used to make the routing decisions.

1) Breaking any assembly tree into a list of due dates for the components

2) The search algorithm that considers the due date of the component in question and all the future possible routes for that component.

3) Introducing new components is accomplished by an algorithm that looks at the system for similar components.

"Computer Integrated Manufacturing System Control: A Data Flow Approach" by Lewis,Barash and Solberg considers the data flow control architecture as method of handling the scheduling problem. Essentially they assign unique number to each task that defines the routing required for that given task. When a process becomes available it picks the next task from the queue of tasks waiting for that process. A task is considered to be waiting

for a process if the corresponding bit in the unique number is set. When the task is completed the process will clear the bit to indicate the next process required and return the task to the appropriate queue. The authors emphasize that there is no central control system nor the need for one. Each process is responsible for obtaining the necessary materials and performing any necessary bookkeeping.

Data is supplied that indicates an improved machine utilization with this approach. A side effect is that increased material handling is incurred due to what I believe is the lack of a central control.

Maimon and Gershwin present a real-time algorithm in their paper titled "Dynamic Scheduling and Routing for Flexible Manufacturing Systems That Have Unreliable Machines". Utilizing a feedback law, their algorithm can compensate for a machine that becomes unavailable because of downtime. One of the keys to the success of their approach is the use of flexible machines. This provides a larger number of routings for any given part. Another key is the assumption that no time is lost for set up.

They focus on two type of manufacturing scenarios, one being a robotic cell assembling printed circuit boards and the second dealing with a machining center for metal part fabrication. The scheduling problem for both of these is the same, selecting an appropriate loading schedule along with determining an effective routing.

# Functional Description

The implementation of this thesis involves three major components. These three components exist as independently executable modules that access the same factory database. These modules are referred to as the User Interface (UI), Engineering Manager (EM) and Schedule and Run (SR).

The UI allows the user to have complete access to the other two modules, along with allowing the user to perform schedule entry and factory status information maintenance.

The EM allows the user to create and maintain the complete factory specification database. The specification database contains static factory data.

The SR component calculates how the user entered model schedule will be produced on the various mold machines and mold tools in the factory. A factory simulation is provided in the SR program to provide testing capabilities.

The program is menu oriented utilizing single key commands. This provides an efficient package that is easy to use. The system is composed of the three executable files, nine factory database files and five schedule output files. A common thread between the modules is the database structure. In the next section, I describe the database files that are used for this system along with how the basic data entry menus operate. Following that, each executable module is presented with the output file structure explanation completing the section.

## Factory Database Description

The database for the factory is stored in nine files that are described in TABLE 1.

| File Name | Data Description | Created by | Accessed by |
|---|---|---|---|
| machspec.dat | machine number<br>machine size<br>(5) prioritized molds<br>service date | EM.EXE | UI,EM,SR |
| moldspec.dat | mold tool number<br>mold service date<br>machine size required<br>number of cavities<br>(12) avail. part numbers<br>base cycletime | EM.EXE | UI,EM,SR |
| partspec.dat | model number<br>(15) part numbers | EM.EXE | UI,EM,SR |
| machhist.log | machine number<br>downtime %<br>runtime %<br>average setup hours<br>average repair hours<br>mean time between failure | EM.EXE | UI,EM,SR |
| moldhist.log | mold tool number<br>downtime %<br>runtime %<br>acceptance %<br>average setup hours<br>average repair hours<br>mean time between failure | EM.EXE | UI,EM,SR |
| machstat.log | machine number<br>current mold tool number<br>status | UI.EXE | UI,SR |
| moldstat.log | mold tool number<br>current location<br>status | UI.EXE | UI,SR |
| partstat.log | part number<br>quantity | UI.EXE | UI,SR |
| schedule.log | model number<br>quantity<br>ship date | UI.EXE | UI,SR |

TABLE 1: Factory Database File Description

The data description column indicates only the user entered data. Some records have additional information that is assigned at the time of record creation. This additional information is not user accessible through the program other than in hardcopy printouts. The first item in the data description column for each file is the record key. All these files are kept in ASCII form in the same directory as the executable files. Whenever these files are read into memory a backup file is created with .bak extension. This backup procedure provides some protection against data corruption problems. The user should keep backup database files on a separate disk to maximize data protection. Example factory database file are located in Appendix One.

## Basic Data Entry Menu System

The basic menu format looks similar to the following example for all the data entry routines:

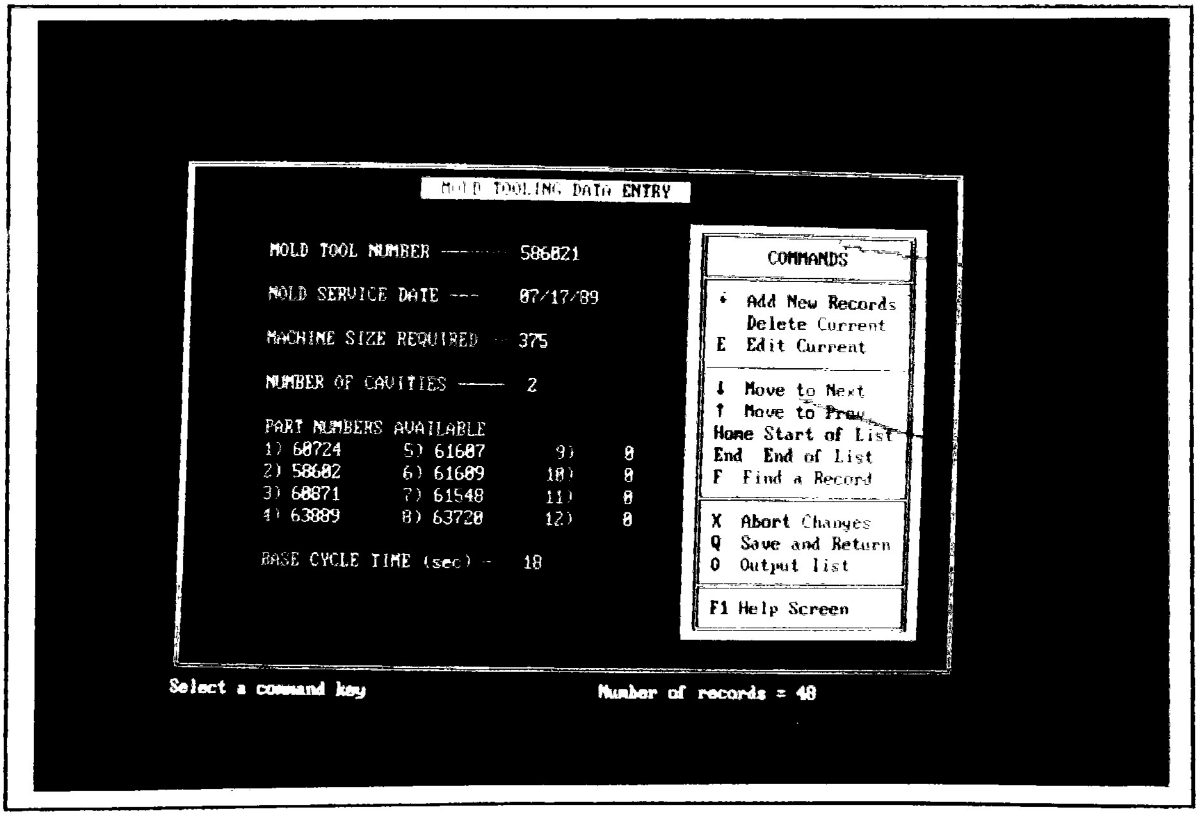


Figure 1: Sample Data Entry Menu with Command Box

This particular example is used for mold tool data entry in the EM.EXE module. The data entry menus use a common command structure that is summarized in the following instruction section:

(+) Add a new record to the list. Duplicates are prevented if the data field is the record key. When adding a new record, all fields are cleared and the cursor is positioned in the place where the data is required. Enter the data required in the format shown. The format screen is shown in Figure 2. The message bar at the bottom of the screen gives the entry commands. If <enter> is pressed when the field contains the correct information, the cursor will move to the next field requiring data. If a default value is available, it will be displayed in the field position and can be accepted by pressing enter. Once all fields are completed, pressing ESC or PgUp will accept the record. The screen will then display this new record as the current record. The entry of a new record can be aborted prior to completing all fields by pressing ESC.

(-) Delete the current record from the list. The user is prompted for verification of the deletion prior to execution of the command. The user must enter n/N to abort deletion.

(E) Edit the current record. Only certain items in the record are available for editing. In general the record key cannot be edited. Fields available for editing will have the cursor displayed at them one at a time. To accept the current value, just press <Enter>. To abort prior to editing all fields, press ESC. Once all fields have been edited, press ESC or PgUp to accept the edited record.

(X) Abort all modifications from this session. List reverts back to previous state. The list is not saved to file.

(Q) Saves all modifications to the file and exit the current menu.

(O) Output the list to the stdprn device. Output is formatted to use an 80 column printer. No special printer commands are issued.

(F1) Displays the help screen

(F) The user can enter a specific record key and move the list pointer to that record if it exists. An error message is displayed if the record cannot be found.

(CUP/CDN) Move the record pointer to the previous/next record. The list is circular so the previous record to the first is the last.

(HOME) Move the record pointer to the first record.

(END) Move the record pointer to the last record in the list.

Exceptions to this generalized command interpretation do exist. The exceptions are noted in the module descriptions where they arise. All input is checked for correctness in syntax and range checked where required. If improper information is entered, an audible alarm sounds and the data field is cleared alerting the user to try again. If duplicate information is entered for the record key, the user is notified of the duplication and is prevented from creating a duplicate record. When a data entry command is issued by the user (+ or E), the command box disappears and a format screen is shown to guide the user as to the type of data required.

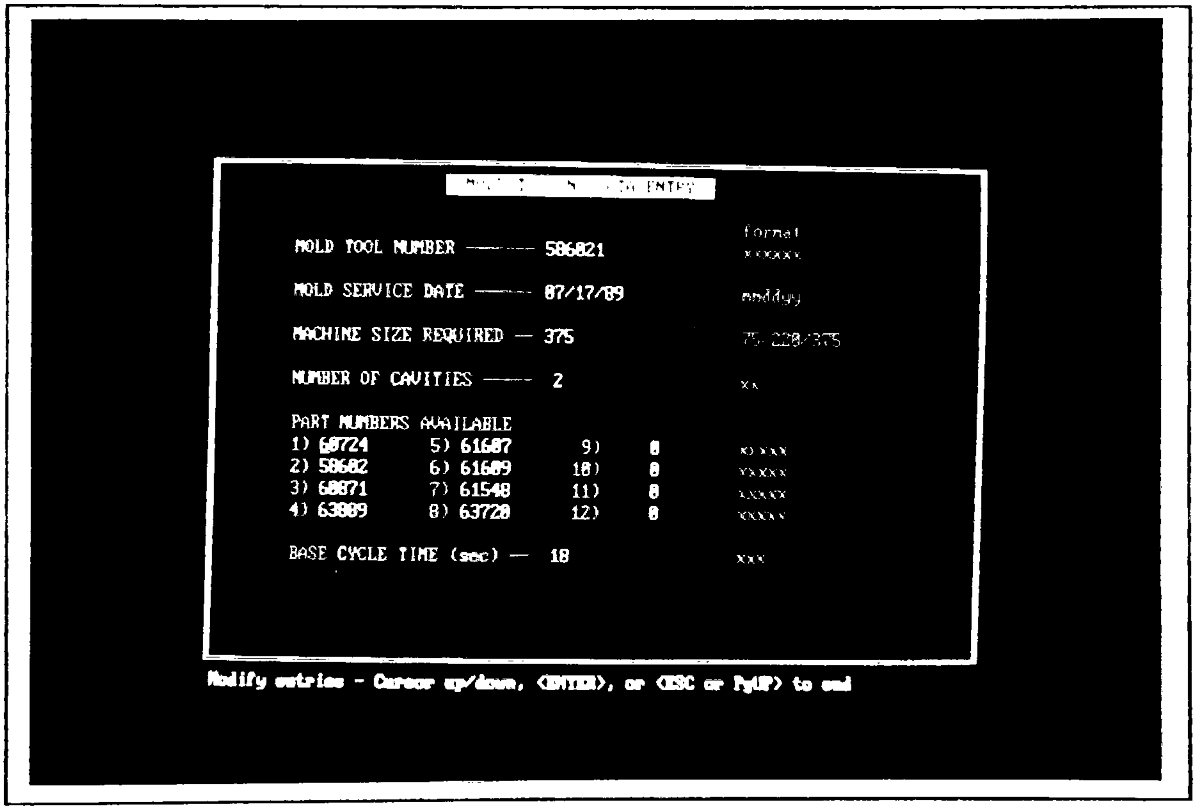


Figure 2: Sample Data Entry Menu with Format Display

The format display indicates the correct form for data entry. "xxxxxx" indicates that a maximum of six digits of the integer type is required. "xx.xx" indicates a two place real value in expected. The date format of mmddyy requires a two digit month value with two digit day and two digit year.The date entered as 122589 will display as 12/25/89. Year values less

than 80 indicate the 21st century. If the field has a default value it will be displayed when the cursor is at the field.

A message facility utilizes the last row on the screen to communicate to the user what type of command is expected. Normal communication is occurring when this message has a blue background. Unrecognized commands cause a non fatal error message on a blue background. Serious errors are displayed on red background. Any error will cause the speaker to sound. When an error message is displayed, it will explain the next command required in order to continue.

## System Command Structure

The menu command tree that is used for this system is shown in FIGURE 3:

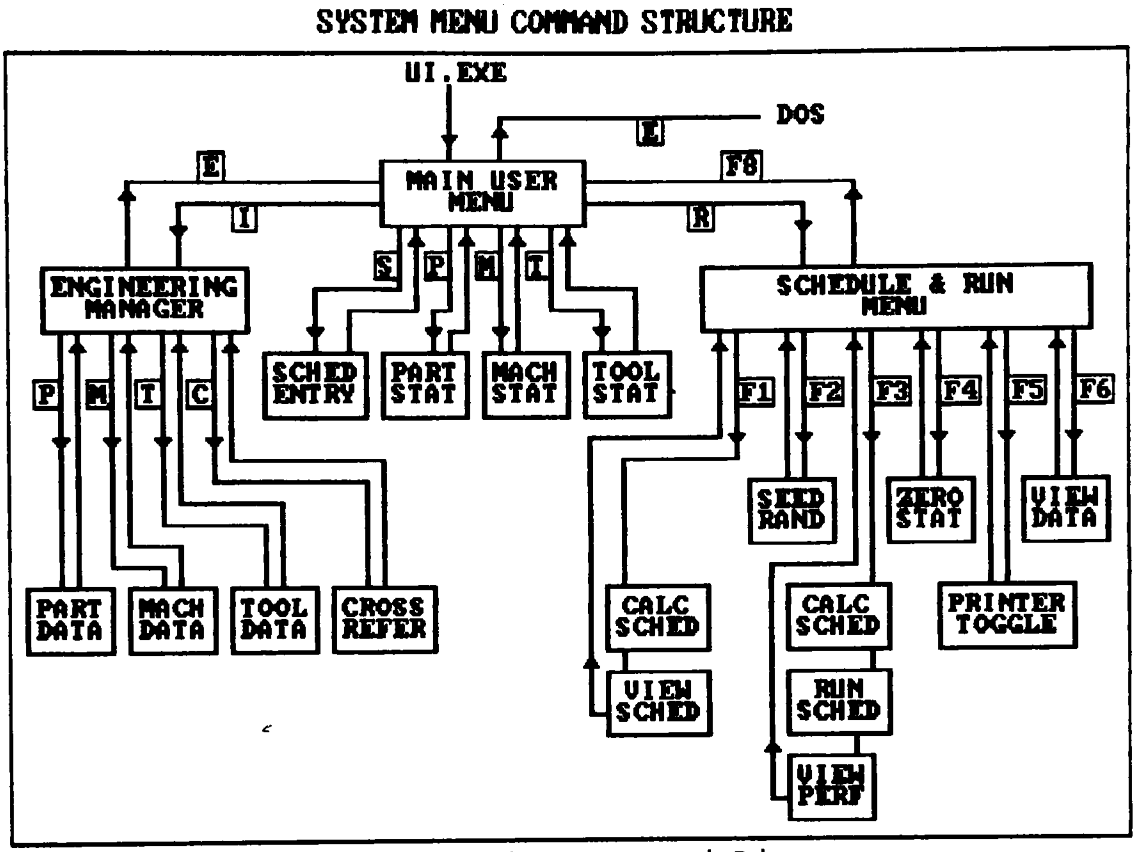


Figure 3: System Command Diagram

This tree shows the command relationship between the main modules and the commands available from each module's main menu. Commands are enclosed in the boxes next to the path lines. The data functions of the EM and the stat and entry functions of the UI, all use the basic data entry menu that was previously described.

In the following sections, each of the module's main menu is described in detail. Any exceptions to the general command structure previously outlined are noted in the explanation for each module.

## The User Interface

When UI.EXE is executed, the user is presented with the opening menu shown in Figure 4:

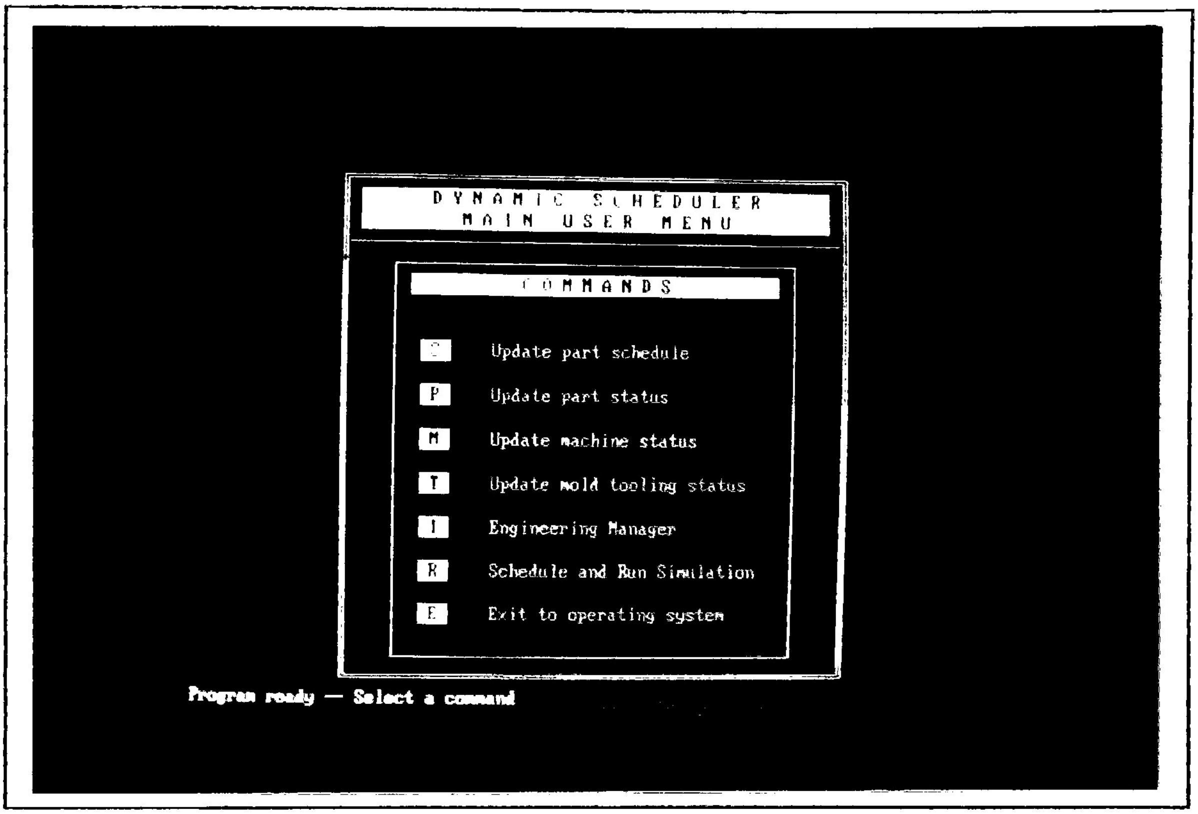


Figure 4: User Interface Main Command Menu

Command S allows the user to input a schedule that contains the three pieces of information indicated for schedule.log in TABLE 1. The model number is verified against the existing part list database, the quantity is checked to be greater than zero and the date input is verified to be an actual date. The generalized command structure holds true for the schedule entry with one exception. When entering a schedule, duplicate model numbers are allowed providing for multiple daily requirements of a given model.

The Commands M, P and T all operate in almost the same fashion as the generalized command structure with several subtle differences. These commands maintain status records for the main data types: Machine, Mold and Part.

Each of these types must have a status record that corresponds to a each specification/history record pair. For this reason the addition, editing and deletion operations are handled differently than in the specification records. The addition feature (+) adds new status records automatically based on the currently available specification records. Deletion (-) is only allowed if no specification record can be found. Editing is limited to status information only, record keys cannot be altered. A create function (c), creates a new set of default status records, overwriting any old ones, after giving the user an audiovisual warning and an opportunity to abort the creation.

By entering the I command, the user is placed in the EM module. The EM module is described in the next section.

The R command executes the SR module for access to the scheduler and simulation menu. The SR module is described in the section following the EM section.

The Engineering Manager

With the execution of the EM module the user is presented with the menu in Figure 5:

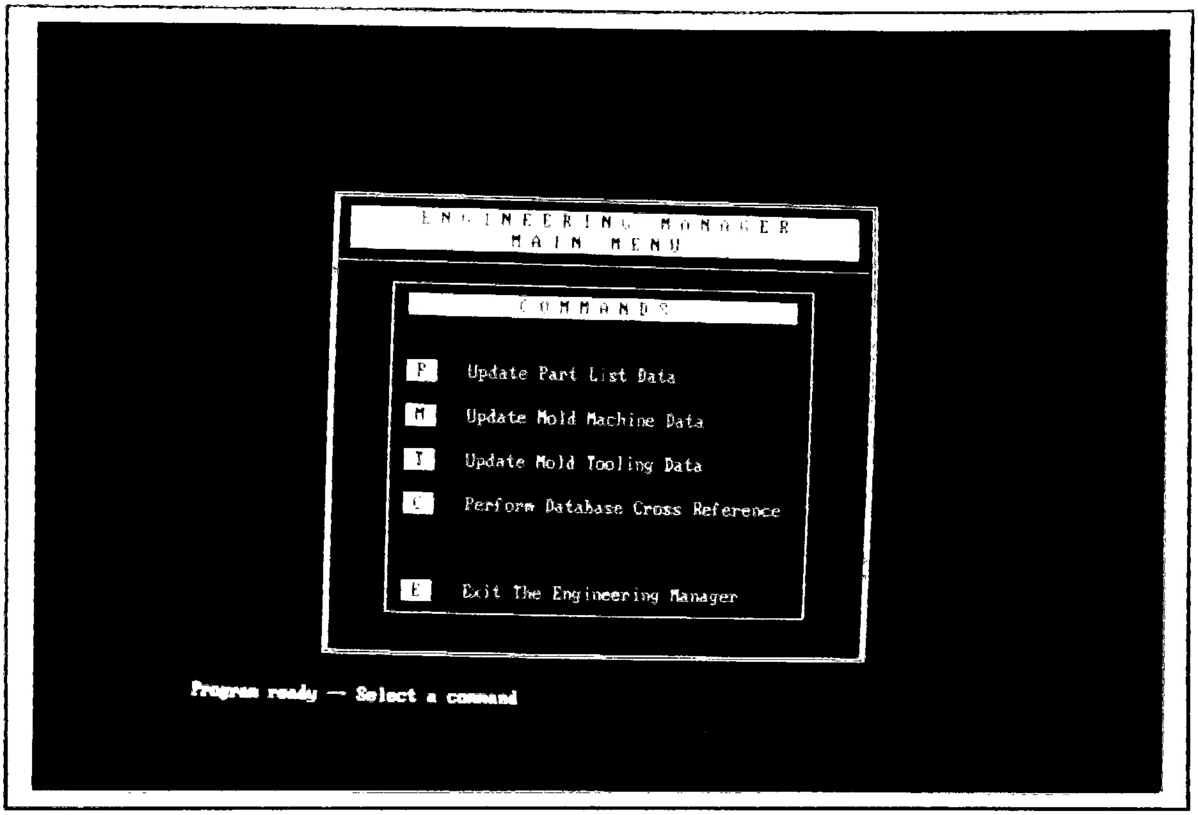


Figure 5: Engineering Manager Main Command Menu

The M, P and T commands perform to the general command structure outlined above. These commands maintain the specification/history record files.

The M and T commands have an additional embedded menu that controls the creation of the history records. The history records for the molds and machines are only available to the user when a new tool or machine is added to the database. Therefore, the user has only one opportunity at entering the data required for the history files. This was done because the factory will generate all the data contained in the history records once the system is operational. The initial values are all defaulted to values that the user can

accept with a single keystroke. These initial values are estimates of how the equipment will function when operational. The history menus have two commands for the user: (A) to accept the default values or (E) to allow the user to edit the performance values. The editing of a history record is accomplished in the same manner as the other data entry routines. When editing a history record, the machine or mold number cannot be changed.

The C command provides a cross reference check of the three databases created by the EM module. These checks are done externally to the menu structure to allow the user freedom in entry order of the data. If a discrepancy is found during the check, an error report is sent to the stdprn device. The checks that are performed are:

Check model part numbers to mold part numbers. This ensures that any part number has a mold that can produce it.

Check that machine prioritized molds actually exist in the mold database.

Check that machine prioritized molds have size requirements that match machine capacity.

By using the C command, the user can be assured that the factory database is specified correctly if no discrepancies are found.

## The Schedule and Run module

The SR module utilizes all the data entered through the other two modules to calculate a schedule and simulate the execution of the schedule in the factory. When the SR module is executed all factory data files are loaded and cross checked to ensure no records are missing. If missing records are found, the user is notified which records are suspected of being missing. The user can then go back to the appropriate menu and correct the problem. The SR module is controlled by the user through the menu shown in Figure 6:

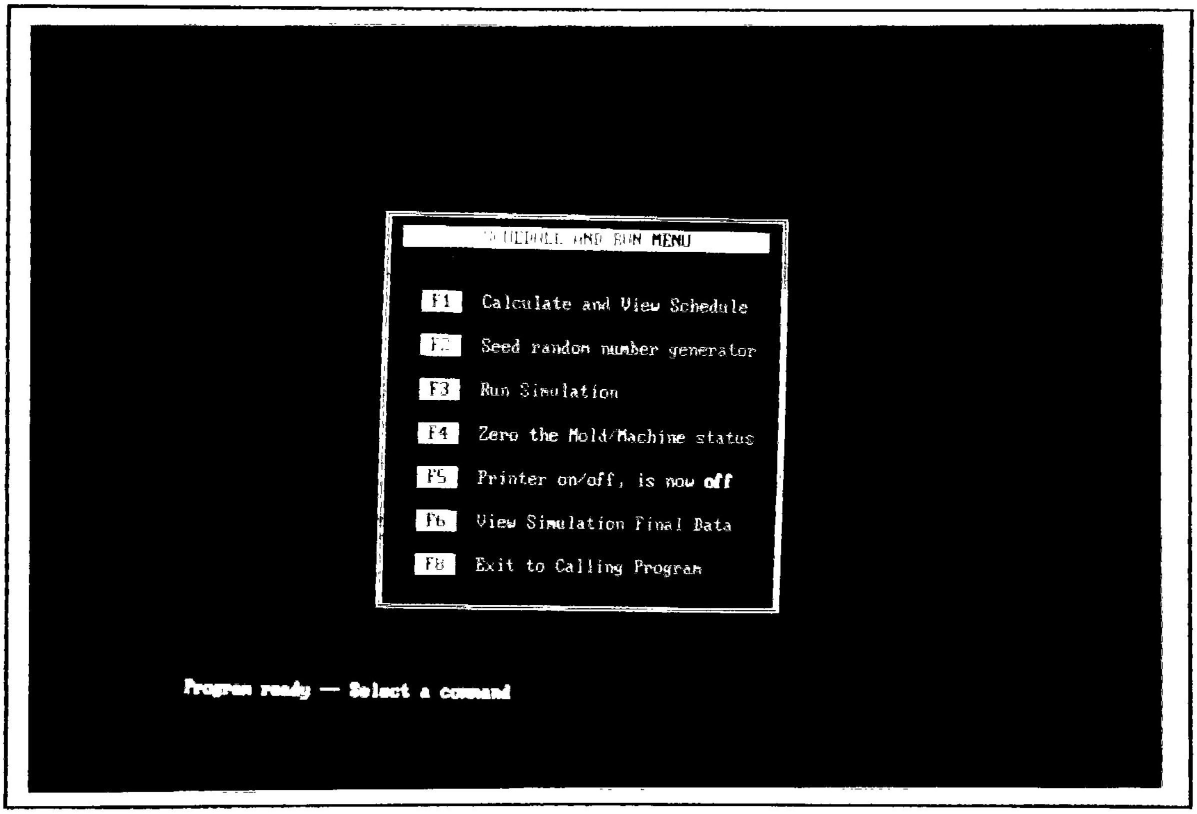


Figure 6: Schedule and Run Main Command Menu

The commands are entered through the function keys. This basic change from the other modules provides the user with the indication that different operations are occurring. An expanded explanation of the function key commands is on the next page.

F1 The schedule is calculated based on the current status of the factory and the requirements of the model schedule. Once the schedule is calculated, the user is placed into a schedule viewer that allows the scanning of each machine and it's mold schedule. A printout is available from inside the schedule viewer by pressing F4. Scanning in the schedule viewer is done with up and down cursor keys. The viewer is exited by pressing the ESC key.

F2 A screen prompt appears requesting the entry of a seed value to be used in seeding the random number generator used during the simulation.

F3 If a schedule was not calculated using F1, then one is calculated now. No schedule viewing is done through this command. The simulation starts immediately after the schedule is saved to machine.sch. Upon exit from the simulation, the user is presented with a Performance Data Menu that has the following commands:

F1 Places the user in the performance viewing mode. The performance viewer allows scanning each machine and mold used for the simulation to view the performance data gathered. Figure 9 shows the format used for this viewer.

F3 Creates a performance printout for the entire factory. This printout includes the information available in the viewer. An example performance printout is included in Appendix Four.

F5 Saves the machine performance data to the machine and mold history records. This is not done automatically since repeated simulations with the same database would not be possible with a variable factory database.

ESC Exits the performance data menu.

When exiting this command, the resulting schedule is saved in results.sch. Appendix Three includes a sample results.sch file.

F4 Allows the operator to zero the factory. This is equivalent to removing all molds from the machines and placing them on the rack. Each mold and machine has its status set to IDLE. The inventory for each part is also zeroed. This allows a simulation to be run over and over to check consistency.

F5 Toggles the use of the printer by the simulator on or off. The simulator uses the printer to output schedule shipment information. The output can be activated by using this command. The default is off. Appendix Four includes an example of schedule shipment information.

F6 Displays the final data for simulation to the screen. The items displayed include actual time used to run the simulation along with the time frame that was simulated. Total parts produced and the schedule attainment are displayed. 100% attainment means that all parts where available on time. A sample final performance screen is in Appendix Four.

F8 Exits to the calling program.

The five output files are created by the SR module each containing a particular level of the schedule. The files are described in table 2:

| File Name | data description | created by |
|---|---|---|
| jobbase.sch | component based schedule | SR.EXE |
| moldbase.sch | mold usage schedule | SR.EXE |
| machbase.sch | machine usage schedule | SR.EXE |
| machine.sch | output form of the schedule | SR.EXE |
| results.sch | machine.sch after the simulation | SR.EXE |

Table 2: Scheduler Output File Descriptions

The *base.sch are created on each pass of the three step scheduling process. All the information included in these files is also contained in the machine.sch and results.sch. The machine.sch shows the schedule that is calculated by the scheduler. Each record contains fields that indicated how many parts are shipped and when the parts are shipped. The results.sch shows the result of running the schedule through the factory. The fields representing the shipping data contain 0 in machine.sch while in results.sch they contain the actual data. These last two files allow a simple method of validating the simulator. Both machine.sch and results.sch are included in Appendix Three.

## The Simulator

While the simulation is running, two basic display modes are available. The primary display is called the global display. The global display shows a set of colored blocks on the screen. Each block represents a machine in the factory, with the color being used to describe the current status of the machineand mold combination. Four pieces of data are shown in the machine block:machine number, machine size icon, current mold number, and quantity left to produce for the current mold. As the simulation continues, the block video parameters and contents are modified to reflect the current state of the machine mold pair. The status of machine and mold are independent so combinations of colors are possible. In the global mode a cursor position is shown by yellow characters and a block shadow outline. White characters are used for all other positions. This cursor position can be moved by use of the arrow, home and end keys. The cursor is used to select a specific machine to be shown on the zoom screen. Figure 7 shows the global display screen for the simulator.

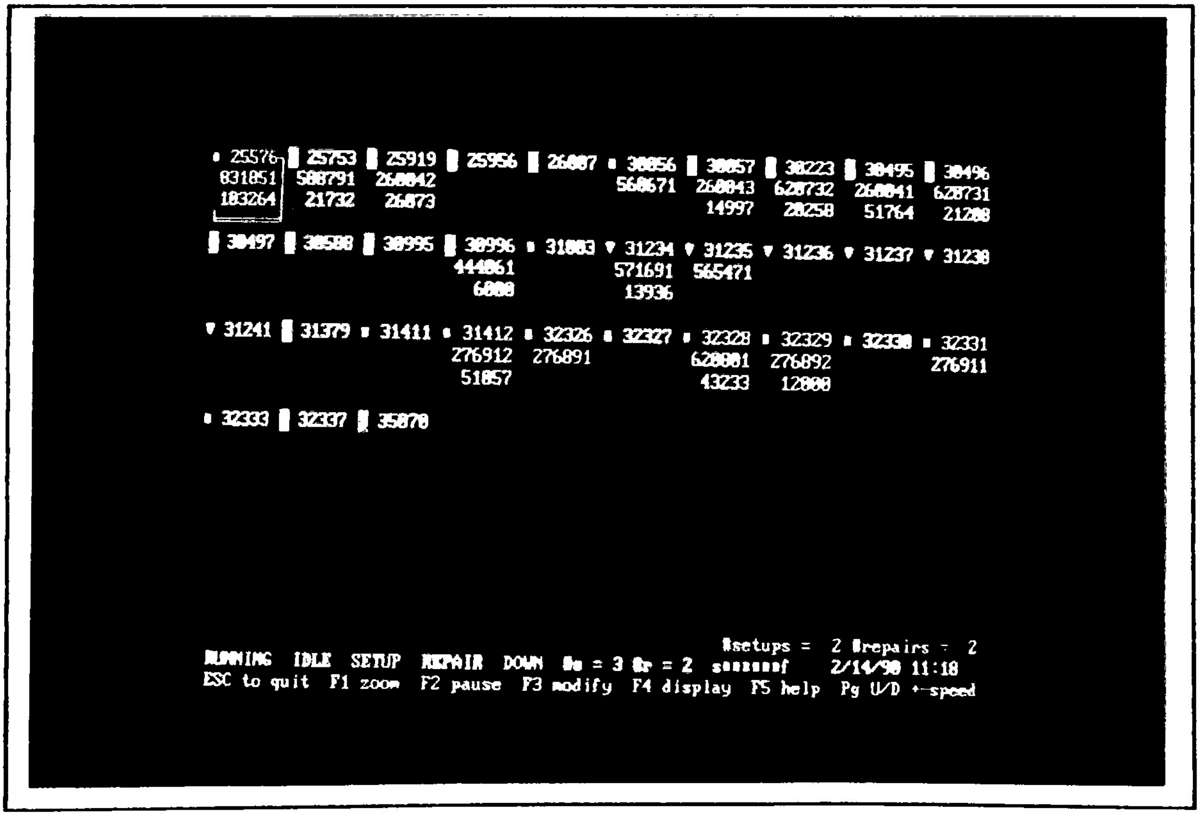


Figure 7: Simulator Global Mode View Screen

The status area at the bottom of the simulation screen gives the user the basic commands that are available during the simulation along with some basic simulator information. The third line from the bottom has two areas for information. The right hand side shows the number of setups and repairs that are occurring at the time while the left hand portion is used to display schedule shipment information. The second line of the status area includes a color key for the machine/mold status, the total number of setup and repair resources available, a schedule speedometer and the current simulator date and time. The last line shows the commands that the user can use.

The zoom mode basically takes the machine block at the cursor position and creates a full screen display for that one machine. The simulation continues to run in the zoom mode and the different machines can be scanned using the cursor movement keys. The zoom screen is shown in Figure 8:

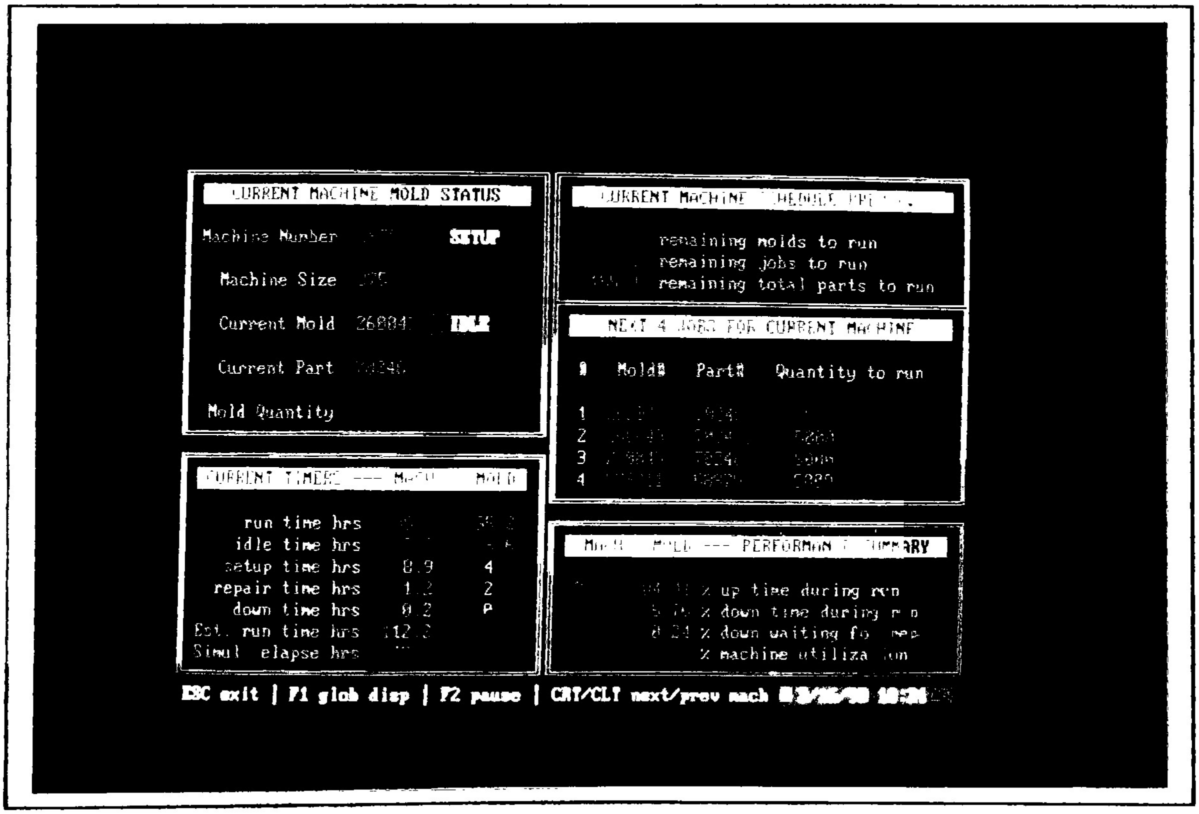


Figure 8: Simulator Zoom Mode View Screen

This screen contains all the information that is needed to watch the simulation progress. The display mode can be toggled back and forth while the simulation is running. As the simulation progresses, parts are shipped as they become available. When a specific model schedule requirement is filled a message is displayed on the global display only. If the printer is selected, the message is printed on the stdprn device. If the display is zoomed, the speaker is beeped to indicate that a requirement has been met.

The commands that are available during the simulation are as follows:

ESC Halts the simulation and exits to the Performance Data menu.

F1 Toggles the display mode. The default mode is the global display

F2 Toggles the pause mode. In the pause mode, the simulator stops executing. Cursor movement is still available in the pause mode.

F3 Allows the user to change the number of setup and repair people that are available. The defaults are seven setup and three repair people. Simulation resources can only be changed from the global display.

F4 Freezes the display but the simulation continues to run. This will allow the simulation to complete faster because less code is being executed. This command only operates in the global display mode. Cursor movement is not allowed in this mode.

F5 Displays a help screen.

PUP/PDN Increase/decrease simulation speed increment. Simulation increments of 1,2,3 and 6 minutes are available. The default increment is 3 minutes. The result of this command is shown on the speedometer.

Cursor keys move the cursor block

When the simulation is exited, the user can enter the performance viewer. The viewer allows the user to scan the performance reports for each machine and mold combination. The reporting structure of the simulator shows two boxes on the screen one for the mold and one for machine. If only the left hand box is shown, the data displayed is for the machine for the entire simulation. If both the machine and mold boxes are shown, the machine data displayed is the performance of the machine while running that mold only. If a minus one is displayed in the simulation row for the mtbf, mttr and setup values, it means that no new value could be calculated from the data gathered by the simulator. The report screen is shown in Figure 9:

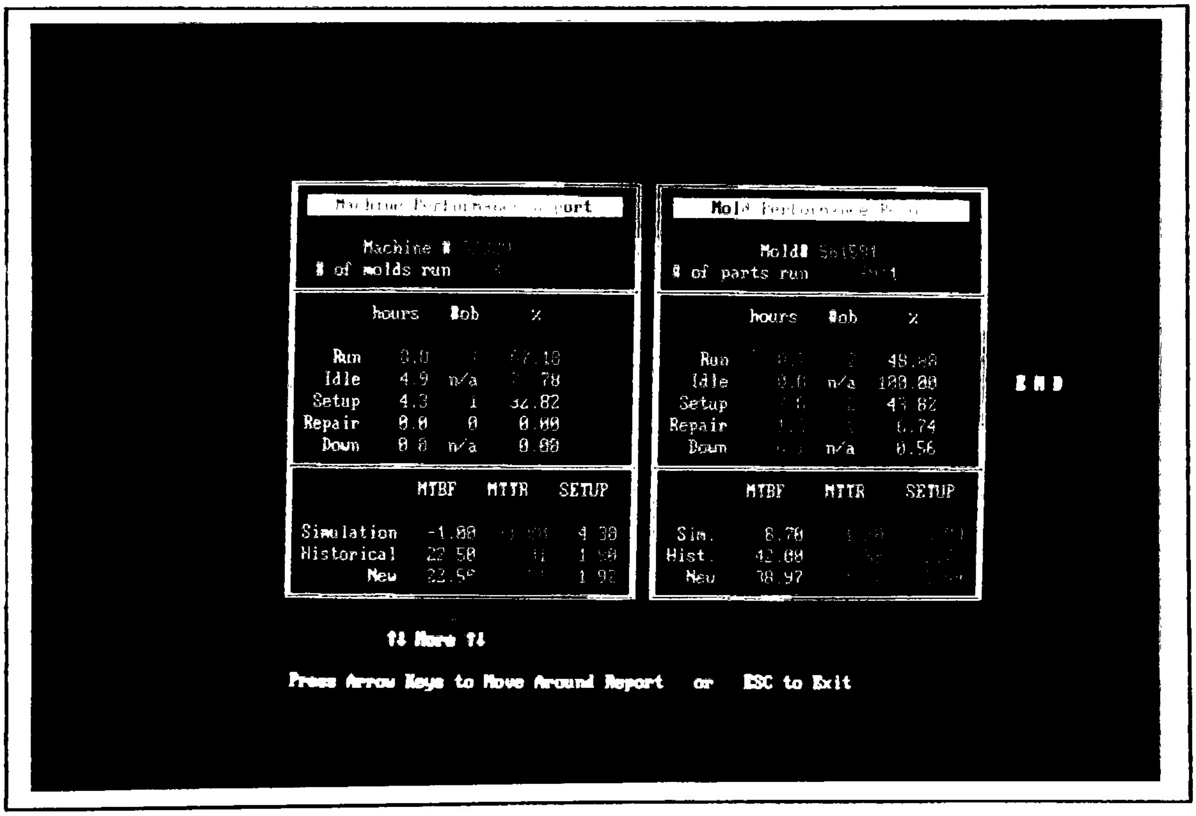


Figure 9: Post-Simulation Performance View Screen

The hours column indicates the number of hours that the machine or mold met a particular condition. The #ob column indicates the number of timing periods which are included in the hours column. A timing observation is incremented whenever the machine or mold itself makes a significant state

transition. A significant transition is, in general, when a random event occurs such as downtime. Setup observations is incremented whenever a major setup is done such as: a mold changeover or part changeover in a mold. A setup period does occur after every repair but the timing for that transition is brief. Idle time does not force an observation to increment. For instance, if a mold goes down, the machine is forced into an idle state. The machine's run timer stops incrementing but the number of run observations for the machine does not change when the machine and mold pair start running again.

The data included in the new row for mtbf/mttr/setup are the values written to the history files should the user request that to be done with the F5 command at the post simulation menu. Each history record also has the number of event observations included. These are also updated when F5 is used.

# Module Designs

## Implementation Tools

### The Machine

The system is implemented on an 80286 based machine running MS-DOS 3.3. The system has been tested on several other machines and has not encountered any problems on any of them. The other systems include an 80386 based machine, an IBM PS/2 and an XT class portable. Running the program from a harddisk improves the performance dramatically due to extensive file I/O. If a printer is connected to the machine, it will be used for the various reports that are available from the program. The program does not check if a printer is available. No errors are generated if the printer does not exist.

### Program Limitations

The modules have been designed to allow the user to enter as large a factory as required. Since the memory for the database is allocated dynamically, the only limitation is MS-DOS addressable memory. The modules were compiled as large memory models to allow up to 1 Megabyte of data storage. Neither expanded or extended memory is supported by this program. When the sample factory from Appendix One is running in a simulation, the approximate memory requirement is 308K.

The simulator has a display limitation for the number of machines displayed during a simulation. When the video display mode is externally set at 25 lines, the maximum number of machines that can be displayed on the global display is 50 machines. If the user can set a 43 line video mode, the simulator will display up to 100 machines. The program does not set it's own video mode. The default video mode is determined by the mode set on the computer prior to the execution of the program. The user can have more

machines in the factory than can be displayed on the global video display. All machines in the database are displayed on the zoom display of the simulator. In the global viewing mode, the machines that are displayed are the first 50 or 100 machines in the database.

## The Source Code

The program is written in C++, utilizing the Zortech C++ compiler version 1.07. Several Zortech utilities were used to automate the more mundane programming tasks.

The screen displays were designed utilizing the Zortech Professional Application Screen Designer-Generator Pro-Screen. This utility stores screen display formats in BURSTRECs. A BURSTREC contains a set of records that designates what to display, where to display it and with what video attribute to display it. An example of this is:

```
            Text to display                      screen position
 ___________________|_____________________________   _|_        ___ burstrec termination
|                                                 | |   |      |
"        The Simulation Main Menu                 ",0x23a,0x71,NULL
                                                         |__|
                                                    video attribute
```

By using a paint(BURSTREC) command the screen is updated almost instantaneously. Pro-Screen provides tools to gather input from the screen. These input tools were used for all data entry menus. The input tools provide basic range and syntax checking. More rigorous data validation can be done with user supplied error checking functions. This is a feature that I used extensively to ensure correct data input. The Pro-Screen facility also contains a message routine that uses the last line of the display. This routine is used extensively to communicate with the user.

The two main datatypes that are used in the program are doubly linked lists and arrays. Both of these are implemented as classes that provide dynamic memory allocation. Zortech C++ Tools provided the Class definition

of each of these. I modified the Dlist class to include a link before command and made the implementation into a circular list. Another of the Zortech C++ Tools that I used is the Time_Info class, This class is used to handle various clocks used by the simulator.

### The Data Handler Class

One of the cornerstones for this system is the Data Handler Class (DH). Each System input and output file has at least one DH associated with it. There are a total of twelve data handler classes included in this system. The DH is designed with one friend method for all instances, that being the file backup routine. The design of the backup routine is a simple character by character copy of the datafile to the backup file. The collection of the twelve data handlers is contained in a library. All the methods in these datahandlers are public. The purpose of these datahandlers is to isolate file and list handling outside the module designs. Simple calls are used to load all data and manipulate it within each module. This simplifies the module code and keeps the datatype interfaces robust between modules. The methods included in each data handler are described in Table 3. The interaction of the three executable modules with each of the database and output files is shown in the system block diagram in Figure 10.

| Method Name | Method Function |
| --- | --- |
| read_from_file() | Reads one record from the open input file. |
| write_to_file() | Writes one record to the open output file. |
| save_list_to_file() | Controls writing an entire list to a file. |
| load_list_from_file() | Controls reading all records from a file. |
| put_in_list() | Places a new item into the list in sorted order. |
| empty_list() | De-allocates memory used by the list. |
| printout() | Provides formatted output to the stdprn device. |
| get() | Moves the current list pointer to the desired position. |

TABLE 3: Datahandler Method Description

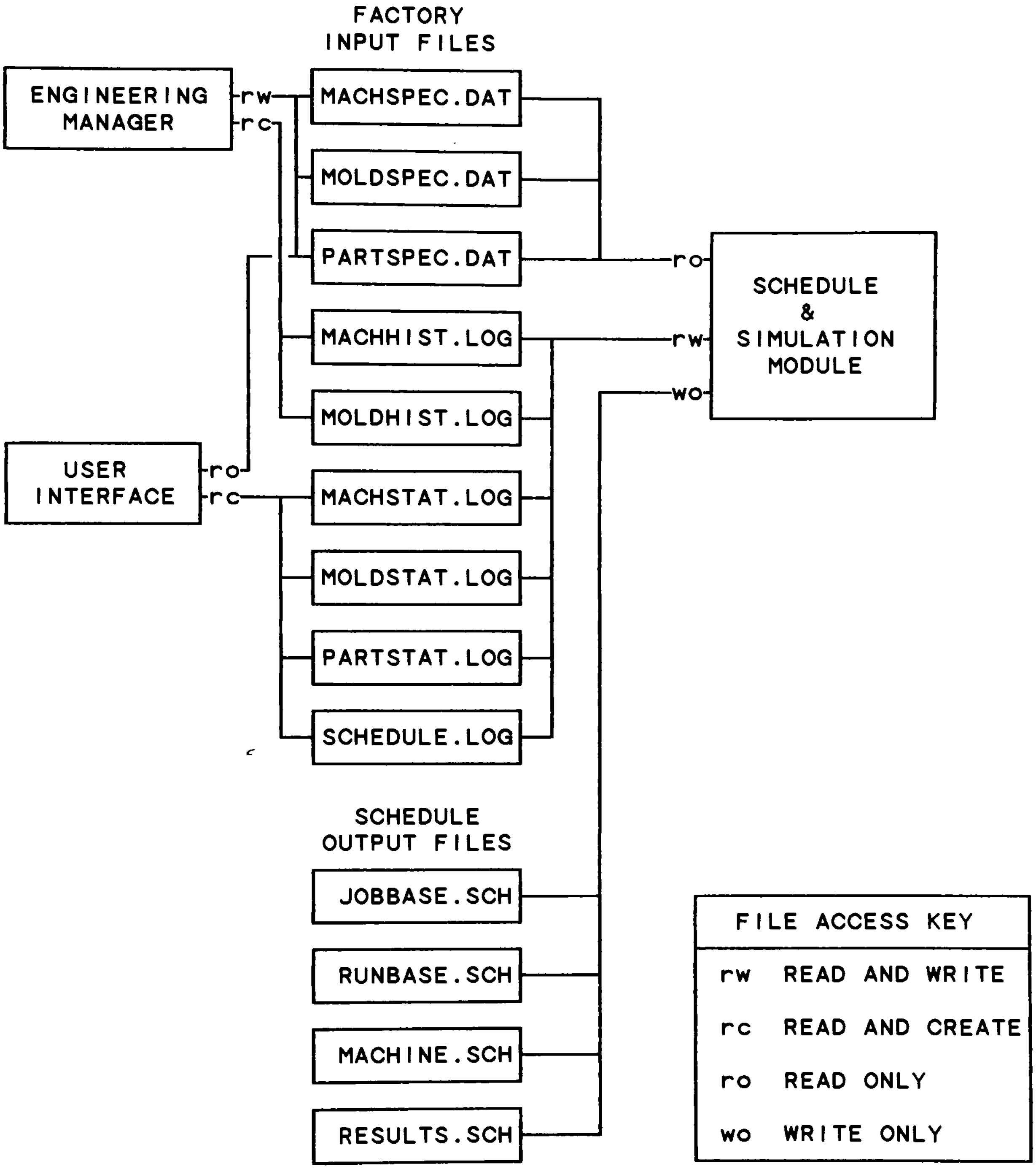


Figure 10: System Block Diagram

## User Interface Module

This module is designed with a main menu service routine and four data entry menu functions. Switch statements are used to execute the user commands. The main menu routine handles the list loading calls and the file save calls. Data entry menu functions return a value of true if the file should be updated. The basic function of this module is to serve as an interface between the user and the data handler classes for the status data and schedule entry files. The majority of the code in this module is dedicated to ensuring that the data placed into the database is acceptable. Less error checking is required in the SR module because of this. An additional function of this module is that it provides access to the other modules in the system. The modules are called with a system call to MS-DOS requesting execution of the particular module desired.

## Engineering Manager Module

This module is designed in the same manner as the UI module. The basic difference is the type of data that is accessible to the user. No system calls are done from this module but the same emphasis on error free data entry is incorporated. The Engineering Manager was conceived to be a separate module for data security reasons. The data included in the files accessed by the module are only entered when a new machine or mold is introduced into the factory. Otherwise, the user entered data is static from the data entry or editing standpoint. All access to day to day data is available through the UI or SR modules. In an implemented system, password protection or a separate computer should be used for access to the EM module. For the purpose of this thesis, the entire system is integrated.

## Schedule and Run Module

This component of the system has two major functions. The first is to calculate the schedule while the second is a simulation function that executes the calculated schedule. In the next section, I describe the design of the scheduler and then follow that with a description of the simulator.

### The Scheduler

The main goal of this scheduling algorithm is to minimize the resources required to complete the schedule on time. The algorithm minimizes inventory by using any existing inventory to offset the number of parts to build. All requirements for a specific part number are scheduled to be produced sequentially to lessen the number of setups needed. The molds are scheduled into fewest machines necessary. By having the fewest machines scheduled, the number of simultaneous repairs and setups are also minimized.

It is possible for the user to enter a schedule requirement that cannot be scheduled to be completed on time. This stems from the fact that a mold can only produce a limited amount of parts in a given time frame. This amount of parts is the capacity of the mold. If the schedule requirements are within the capacities of the molds needed, this algorithm provides good results. If requirements exceed the capacity of a mold, the scheduler will schedule as many of the parts to be produced in the given time frame as possible. Any remainder will not be scheduled. The user is notified of this event should it occur.

Attainment is a measure used to determine the performance of to meeting a particular requirement. When the schedule is run on the simulator, an attainment percentage is calculated at the end of the simulation. This attainment is calculated in the following manner:

$$\frac{\text{\#of ontime parts}}{\text{\#of parts required}} = \text{attainment percentage}$$

Parts are logged into inventory in batch quantities. A batch is sized by the quantities assigned by the user in the schedule requirements. If 5000 of model 34567 are required to ship by 10/1/89, the batch quantity will be 5000 for all parts required for model 34567. If all parts are available by midnight on 10/1/89, then the attainment would be 100 percent. If model 34567 requires four component parts and one of the part requirements cannot be completely satisfied, the attainment would be 75 percent. Attainment cannot exceed 100 percent since any additional parts are placed in inventory for use on future schedules.

The scheduling calculation is done in three basic steps that I listed in the introduction. They are listed here again with some additional terminology as an introduction to the detailed description of each.

1) Break the model based schedule into a component based schedule. will refer to this as the job based schedule. Each node is referred to as a job.

2) Assign the part based schedule into molds with adequate time windows. This is what I call the run based schedule. Each molds schedule is referred to as a run.

3) Schedule the molds into machines with adequate time windows. This is the machine based and final schedule.

<u>Pass One: The Job Based Schedule</u>

The first pass takes the user entered model schedule and breaks it into a component part schedule. Any inventory for a particular component is allocated to a given model as the model schedule is broken down. If there is sufficient inventory to cover a required quantity of a component, then that part is not listed in the job based schedule. Any left over inventory is allocated to the next request until no inventory is available to be allocated. This inventory allocation is done to keep inventory turns at a maximum. A bar chart is displayed while the job based scheduling is occurring to notify

the user of its progress. The output of this scheduling pass is an ordered linked list of component part builds. They are ordered by the part number and by the required ship date within each part number. The information that is included in each node of the job list is as follows:

Part number = as req'd
Ship time = as req'd
Shipped time = 0
Quantity to build = as req'd
Shipped quantity = 0
demand = 0
tagged = FALSE
order = 0
scheduled = FALSE

The shipped time and quantity are used by the simulator to indicate actual parts shipped. The quantity to build is decremented as the simulator runs. Demand, order and scheduled are used by the pass two of the scheduler for logic and array position control. Tagged is used by the simulator to track which job is running in a mold. Machine.sch in Appendix Three shows where each of these records is linked to the machine schedule. Each job based node is listed under the heading "[----- PARTS To RUN -----]". Capacity related errors are not detectable on this pass of the scheduler since it really is just a schedule conversion and list formatting procedure.

Pass Two: The Run Based Schedule

After the job based scheduling is completed, the scheduler moves directly into this second pass. A linked list entry is created for each mold tool in the data base with the following data:

Mold tool number = as req'd
Dynamic array of jobs = NULL
start time = 0 (always 0 since it is not used)
end time = 0
tagged = FALSE
order = 0
scheduled = FALSE
quantity = 0

The demand rates for each job are calculated on this pass. During the demand calculation if a part is encountered that is past due, the scheduler announces the discrepancy to the user. If a past due model is encountered, the user has the option of aborting the scheduling activity. If the user chooses to continue, the scheduler will attempt to schedule the past due parts to ship within three days of the schedule calculation time. Order and scheduled are used by the final pass of the scheduler. End time and quantity are incremented as each job is placed in the array of jobs. End Time is the length of time that the mold is expected to run in order to complete all the jobs in the array of jobs. Tagged is used to identify potentially usable molds for each job.

The basic function of the second pass is to take the highest priority job and place it in the array of jobs for the correct mold until all job based items are scheduled. The priority is simply determined by the earliest ship time that is not already scheduled. Once a job is selected for scheduling, all molds capable of producing that part are tagged. Each of these molds is checked to see if one currently has the component part setup. If one does and the part quantity can be built in time, the job is inserted into the array for that mold. The jobs order is set to the position in the array and it is labeled as scheduled. Two conditions exist that will cause a job to not be scheduled on the first scan of appropriate molds. First, if no mold has the part currently setup, then a changeover is required. The available molds are checked until one is found that has sufficient capacity to support a changeover. The second possibility is that all the molds lack sufficient capacity for the job, in this case the job is split. The schedulable portion of the split job is placed in the mold and the unscheduled remainder's ship time is set to 10 minutes later than the scheduled split. This new job is then subjected to the same scheduling algorithm. It is possible that a job cannot be scheduled

to ship on time. This is caused by trying to schedule beyond the capacity to produce a specific part. The scheduler will schedule as many of the parts to run prior to the ship time with the remainder not being scheduled at all. The user is notified of this occurrence at the end of this pass. When a job is scheduled in a mold, the quantity and end time for the mold are incremented the amount the job requires. The run portion of the schedule is included in machine.sch under the heading "[------ MOLD to RUN ------]".

Pass Three: The Machine Schedule

This pass is started in the same manner as the second pass with a record being created for each machine in the database. The information included here is:

Machine number = as req'd
dynamic array of runs = NULL
start time = current clock
end time = current clock
tagged = FALSE

Here the end time is incremented by the molds end time when a mold is added to the machine schedule. Tagged is used by the scheduler to identify each potentially usable machine, as each mold is scheduled.

Each run based schedule node is taken by priority and placed into an appropriate mold machine. The priority of the run is chosen by the earliest first job ship time in each mold. If a mold has been prioritized to be run in a particular machine, the mold is scheduled there. This is done due to special equipment or machine location concerns. Otherwise, all appropriate machines for the mold are tagged and each is taken in order of descending historical efficiency. The complete job array for the run is checked to see if the machine can produce the parts in time. If so, the run is placed into the machines run array with the end time being incremented, and the run scheduled Boolean is set to TRUE. All machines are untagged with the process

repeating itself until all molds are scheduled. The end result of this third pass is saved in machine.sch. A small example of this is included in Appendix Three.

It is possible to exceed the capacity of the machines in the factory. This can be done by requiring the scheduling of more molds for a particular machine size than can be accommodated. No run splitting is done in this pass of the scheduler. The molds that exceed the capacity of the bank of machines are simply not scheduled.

## The Simulator

The simulator is designed as a class with the majority of the methods and variables being private. This was done more for the learning experience of creating a class than the requirement that it had to be a class. One of the advantages of object oriented programming is the ability to reuse classes in different programs. The simulator class is so deeply tied to the datatypes of the entire system that its reusability is questionable.

There are three basic components that make up the simulator, the simulator display, the data update module and the report generator. When the simulator is started several data structures are initialized. A simulator linked list with a node for each machine is created with the following information:

| | |
|---|---|
| node | integer index |
| mach | pointer to machine status |
| mold | pointer to the current mold status |
| dcol | simulation display column for global display |
| drow | simulation display row for global display |
| mach_dn_prob | current probability of machine going down |
| mold_dn_prob | current probability of mold going down |
| run_accum | unused portion of time inc used to calc parts produced |
| mach_su | exponential time for machine setup |
| mach_rp | exponential time for machine repair |
| m_su | machine setup periods |
| m_rp | machine repair periods |
| m_rn | machine run periods |
| t_su | mold tool setup periods |
| t_rp | mold tool repair periods |
| t_rn | mold tool run periods |
| mold_su | exponential time for mold setup |
| mold_rp | exponential time for mold repair |
| mrn | run accumulator for machine in seconds |
| mid | idle accumulator for machine in seconds |
| msu | setup accumulator for machine in seconds |
| mrp | repair accumulator for machine in seconds |
| mdn | down accumulator for machine in seconds |
| trn | run accumulator for current mold in seconds |
| tid | idle accumulator for current mold in seconds |
| tsu | setup accumulator for current mold in seconds |
| trp | repair accumulator for current mold in seconds |
| tdn | down accumulator for current mold in seconds |

This structure is used by the simulator to keep track of all the necessary information about each machine being used by the simulator. One other list structure is created prior to the execution of the simulation. This is a reporting structure that is used whenever a mold run is complete. The report that is available upon exiting the simulator uses this structure to display the information. This structure contains the following data groups:

A report header

| | |
|---|---|
| mach | pointer to a machine status record |
| mold | pointer to a mold status record |
| mrep | pointer to a machine report |
| trep | pointer to a mold tool report |
| mrn | machine accumulated run time in seconds |
| mid | machine accumulated idle time in seconds |
| msu | machine accumulated setup time in seconds |
| mrp | machine accumulated repair time in seconds |
| mdn | machine accumulated down time in seconds |
| m_su | accumulated setup periods |
| m_rp | accumulated repair periods |
| m_rn | accumulated run periods |

A linked list of report headers

| | |
|---|---|
| machine | machine number used as key |
| tag | a tag used during reporting to track number of reports |
| arr | array of REPORTS for a machine |

A linked list of report headers is allocated prior to starting the simulator. As reports are made during the simulation, machine and mold reports are linked to a report header and added to the array of reports for the appropriate machine. The machine and mold reports are described on the following page. There is data duplication among the report records. This duplication was done to provide convenient units conversion during the report compilation process and rapid report displaying.

The machine report record

| | |
|---|---|
| mach | machine number |
| molds | number of molds machine ran |
| mrn | machine run time in hours |
| mid | machine idle time in hours |
| msu | machine setup time in hours |
| mrp | machine repair time in hours |
| mdn | machine down time in hours |
| r_ob | number of run periods |
| s_ob | number of setup periods |
| p_ob | number of repair periods |
| rnp | run time as percentage of non idle time |
| util | non idle time as percentage of total time |
| sup | setup time as percentage of non idle time |
| rpp | repair time as percentage of non idle time |
| dnp | down waiting for repair as percentage of non idle time |
| mtbf[3] | array to hold simulation/historical/new mtbf |
| mttr[3] | array to hold simulation/historical/new mttr |
| setup[3] | array to hold simulation/historical/new setup time |

The mold report record

| | |
|---|---|
| tool | mold number |
| numparts | number of parts mold ran |
| trn | mold run time in hours |
| tid | mold idle time in hours |
| tsu | mold setup time in hours |
| trp | mold repair time in hours |
| tdn | mold down time in hours |
| r_ob | number of run periods |
| s_ob | number of setup periods |
| p_ob | number of repair periods |
| rnp | run time as percentage of non idle time |
| util | non idle time as percentage of total time |
| sup | setup time as percentage of non idle time |
| rpp | repair time as percentage of non idle time |
| dnp | down waiting for repair as per. of non idle time |
| mtbf[3] | array to hold simulation/historical/new mtbf |
| mttr[3] | array to hold simulation/historical/new mttr |
| setup[3] | array to hold simulation/historical/new setup time |

Each time a report entry is made the current timers and values are assigned to the report type and added to the array. When the simulation is exited, a final report is made and assigned to location zero in the array. The report is compiled and made available to the user for viewing when the simulation is exited.

The simulation updating process is done by determining if machine or mold changes state. State changes are controlled by a set of transition rules. Each machine and mold can change state independently. For example, a mold can go down while the machine does not experience any downtime. In this example, the machine will be forced into an idle state while the mold is down or being repaired. Figure 10 shows all the possible states for a machine/mold combination.

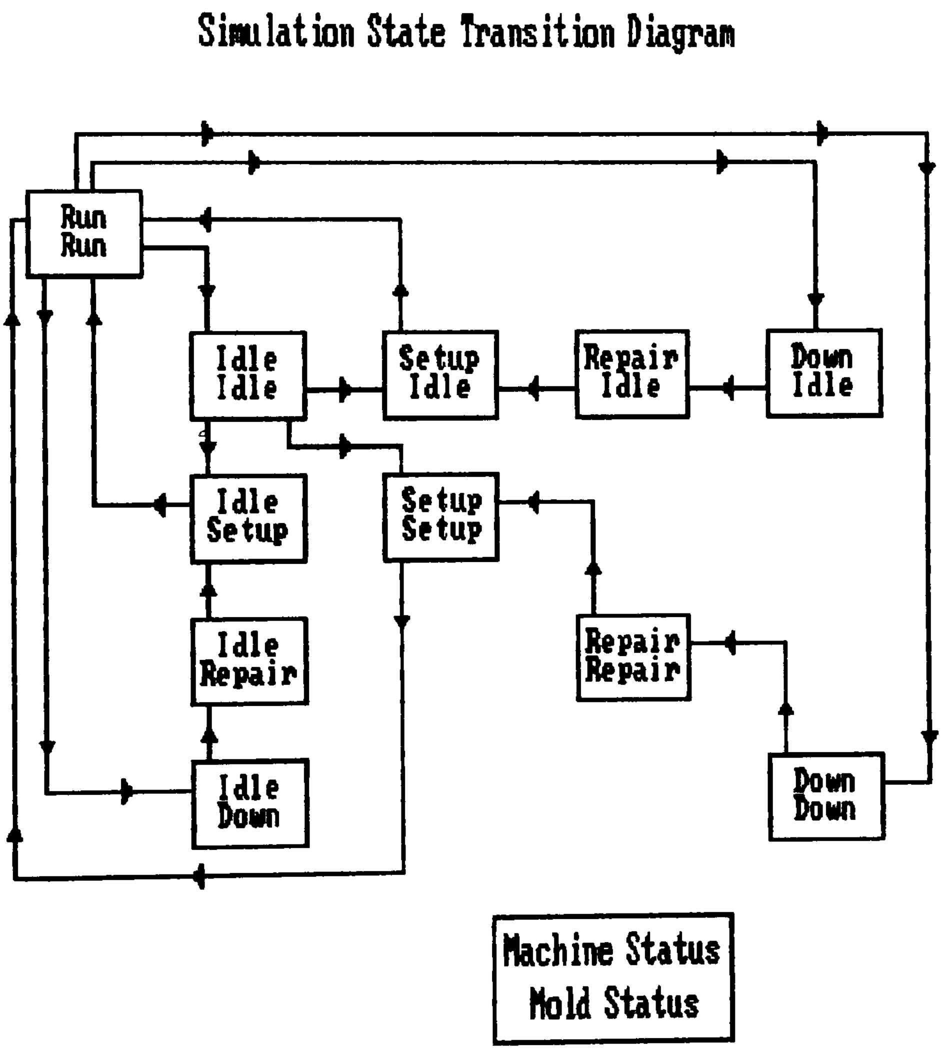


Figure 11: Simulator State Transition Diagram

The rules that control the state changes are outlined on the following page. The machine state is listed prior to the / and the mold is listed after.

Run/Run to Idle/Idle

The current mold job must be completed. If the part number for the next job is the same, idle time will be minimal. If the job is not completed, the quantity for the current job is decremented according to the simulator time increment. When all jobs are completed in a run, the run is deemed complete.

Run/Run to Idle/Down

If the normalized random number exceeds the mold down probability, the mold goes down.

Run/Run to Down/Idle

If the normalized random number exceeds the machine down probability, the mold goes down.

Run/Run to Down/Down

Since both possibilities are calculated, it is possible that both the machine and mold will go down at the same time.

Idle/Idle to Setup/Idle

If the mold in the machine needs to be changed, the machine will be forced into a setup state. The length of time in setup is an exponentially distributed time based on the historical performance of the machine.

Idle/Idle to Idle/Setup

If the part number in the mold needs to be changed, the mold will forced into a setup state. The length of time in setup is an exponentially distributed time based on the historical performance of the mold.

Idle/Idle to Setup/Setup

This state is entered if both setup conditions are met or at the start of the simulation.

Idle/Down to Idle/Repair
Down/Down to Repair/Repair
Down/Idle to Repair/Idle

If a repair person is available, one will be allocated to the repair. If both the mold and machine are down, it will only require one repair person. Repair times are exponentially distributed based on the historical repair times for the equipment. If no repair people are available, the equipment stays down.

Idle/Repair to Idle/Setup
Repair/Repair to Setup/Setup
Repair/Idle to Setup/Idle

When the repair time has elapsed and there is an available setup person, the setup state is entered. Setup time is exponentially distributed. The allocated repair person is returned to the queue when this state transition occurs.

Idle/Setup to Run/Run
Setup/Setup to Run/Run
Setup/Idle to Run/Run

When the setup time has elapsed, the state is changed to run. If both the machine and mold are in setup, then both times have to elapse prior to status change. The setup person is returned to the queue when the this state transition occurs.

## Random Event Determination

When a machine or mold is running, it is always subject to a random occurrence of down time. This state change is determined by using a random number generator. The Zortech rand() function returns a number between 0 and 32767. I use a conversion routine to map this random number to a normal distribution curve. The return value from the function is the probability that the machine or mold will transfer from the run state to the down state. When the simulator is started, a normal curve distribution array is filled with 200 values. The values represent the summed area under the normal distribution curve shown in figure 11. Every six minutes during the simulation, a random number is generated and mapped to a value between 0 and 1. The array is indexed from 0 until the random value is exceeded by the value in the array. The search algorithm uses steps of 20 to get close and then single steps to arrive at the final value. The average number of searches is approximately 20 for each inquiry. The return value is the array index divided by 2. If this value is greater than the historical runtime percentage, the transition occurs.

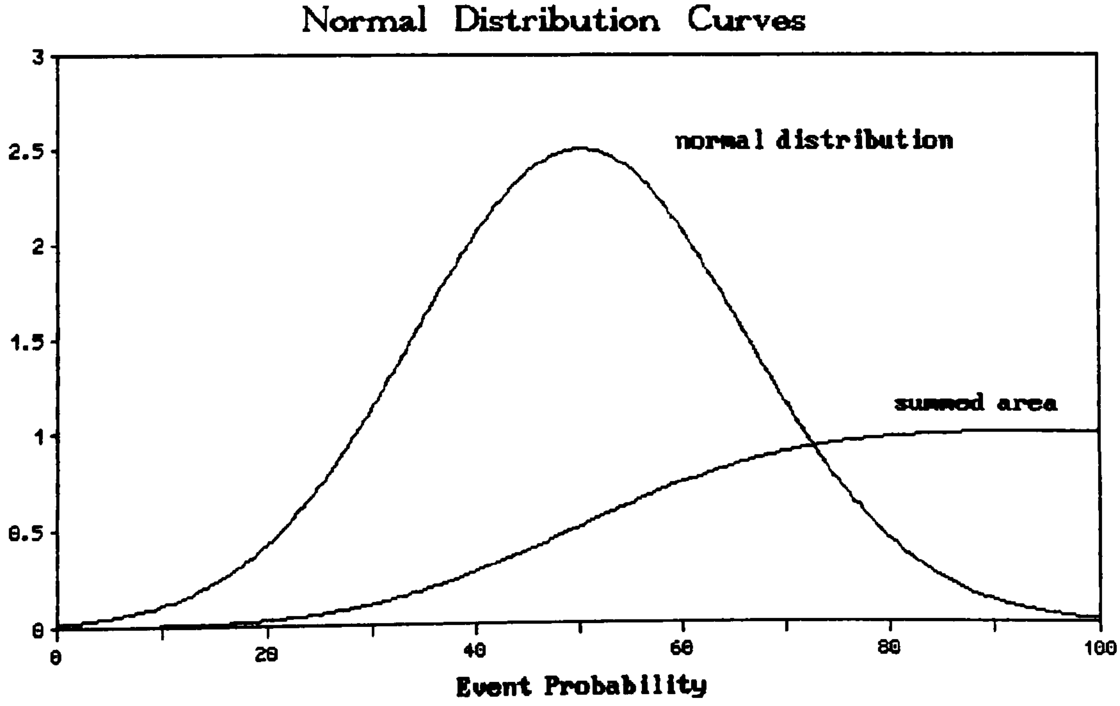


Figure 12: Normal Curve Used for Random Event Determination

The setup and repair timers are randomly calculated using an exponential distribution. The random number is divided by 32767.1 to ensure that the value used in the natural logarithm function will be less than one. Figure 12 shows the exponential curve that is used to determine the event time values. The calculation is a direct insertion of the random number into the logarithm equation. The function returns a value that is multiplied with the historical repair or setup times to determine the time value to use for the event timer.

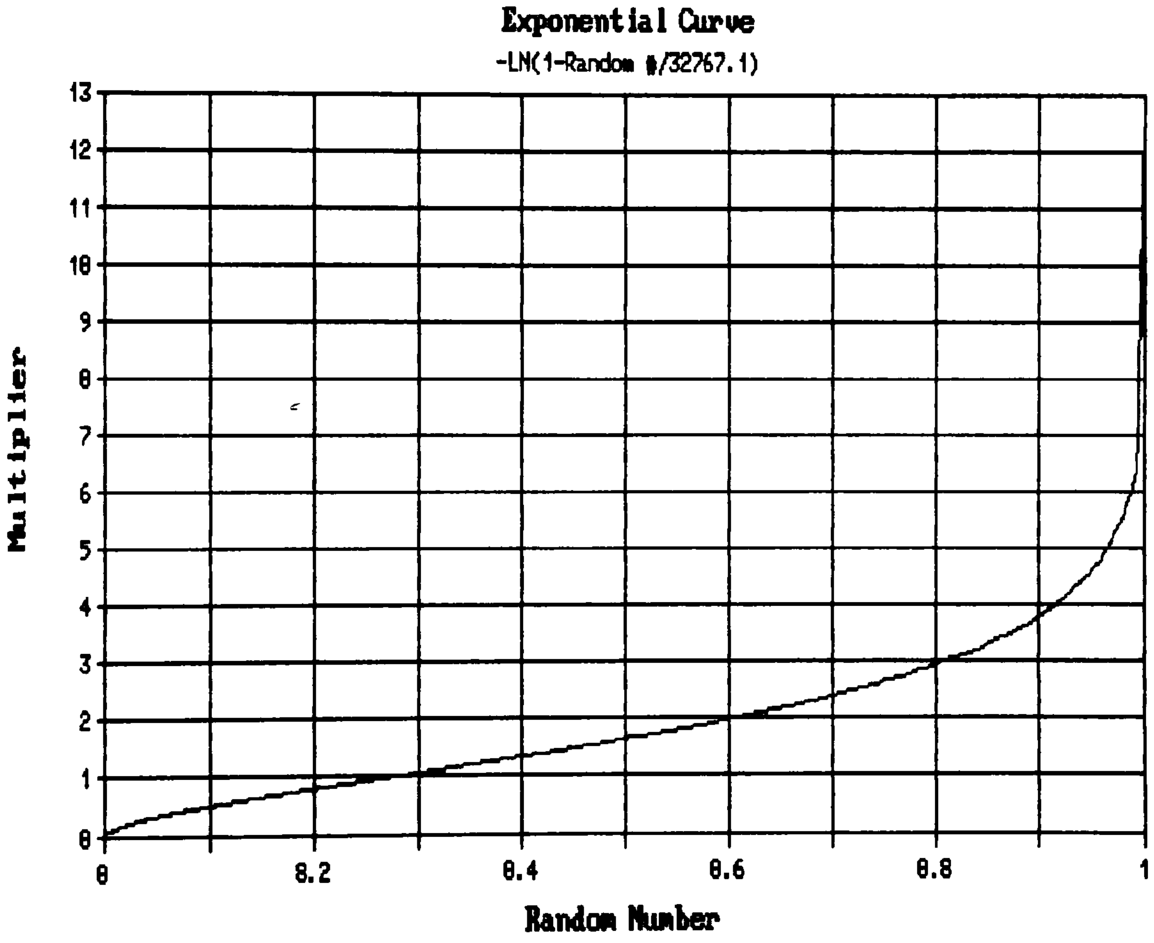


Figure 13: Exponential Curve Used By Simulator to Determine Duration of Setups and Repairs.

# System Validation

## System Concepts

The three executable modules make up a complete system. The system runs in a closed loop fashion making error detection for the programmer and user very easy. The schedule shipment printout can easily be compared to the user entered schedule to ensure that all parts where shipped. The machine.sch and results.sch can be compared to see what happened for each job scheduled in a machine. The simulator was designed as a test routine for the scheduler, but in the end it turned out to perform a complete system test. I contend that this system is self testing and that the output data contained in the appendices support this contention. No specific test plan was used to validate the system. Instead, simulator output is used to verify the operation of the system.

## Keyboard Command Acceptance

Robustness to errant commands was deemed a primary concern in the early stages of design. As each module was developed, it was tested at each level of revision for keyboard entry bugs. The program has proven itself to accept various keyboard entries and not exhibit any lock up problems.

## Accuracy

The only algorithms that may eventually introduce a shift in system performance are the random event determination functions. If the numerical correlation of these functions to a real world environment is not valid, then the data that is determined by using them will begin to shift in value. The main intent of this thesis was to construct a system that could be implemented in a real factory. Any correlation problem is likely to be minimal and will not affect the use of this system for short term scheduling.

# Conclusions

## Goals Achieved

The intent of this thesis was to create a dynamic scheduling algorithm for use in a manufacturing facility. It was to take advantage of real time data collection from the manufacturing process and create an effective schedule based on the current state of the factory. That intent has been achieved with the system that I have developed and presented in this document. I had intended to utilize existing algorithms to accomplish the scheduling, but in the end have developed my own. The published algorithms that I found while writing the proposal did not take into account the three pass scheduling that needed to implemented for my factory, an injection molding area. My algorithm is dedicated to my particular version of the factory. I doubt if it could be used in a system where several processes are required to create a saleable product. The difference here is that my factory only performs one process on each part. A several process system requires a routing scheme to be developed. The scheduling problem that I have solved does not have to deal with intermediate process queues or a backlog if a downstream process goes down. A true dynamic scheduler would modify the schedule as required based on system downtime. My system does not modify the schedule midstream. It bases the scheduling on historical data gathered from the running machinery to produce a realistic and attainable schedule. The entire schedule calculation process for the schedule that is located in Appendix One is less than one minute. This easily out performs the current manual method being used.

## The Learning Experience

This thesis has provided me with the opportunity to design and implement a complete system. It has turned out to be a much larger endeavor than what I had first imagined. The resulting system is much more than a scheduling algorithm. I believe that it can be used as an engineering tool for doing capacity studies in a molding department. It also provides a test bed for other scheduling algorithms.

# Bibliography


Lewis, William C., Moshe M. Barash, and James J. Solberg."Computer Integrated Manufacturing System Control: A Data Flow Approach." Journal of Manufacturing Systems, Vol. 6 No. 3 (1987) 177-91.

Lulu, Menberu, and J.T. Black. "Effect of Process Unreliability on Integrated Manufacturing/Production Systems." Journal of Manufacturing Systems, Vol. 6/No. 1 (1987) 15-22.

Dessouky, M.I., Y.M. Dessouky, and M.M. Dessouky. "A Case Study in Parallel Unrelated Machine Scheduling: A Heuristic Approach." Journal of Manufacturing Systems, Vol. 6/No. 1 (1987) 23-36.

Lee, Chung-Yee, and J. David Massey. "Multiprocessor Scheduling: An Extension of the MULTIFIT Algorithm." Journal of Manufacturing Systems, Vol. 7/No. 1 (1988) 25-32.

Ben-Arieh, David H., and Colin L. Moodie. "Knowledge Based Routing and Sequencing for Discrete Part Production." Journal of Manufacturing Systems, Vol. 6/No. 4 (1987) 287-97.

Jones Albert T., and Charles R. McLean. "A Proposed Hierarchical Control Model for Automated Manufacturing Systems." Journal of Manufacturing Systems, Vol. 5/No. 1 (1987) 15-25.

Kusaik, Andrew, and Agostino Villa. "Architectures of Expert Systems For Scheduling Flexible Manufacturing Systems." IEEE 1987 International Conference On Robotics and Automation, Vol. 1 (1987) 113-17.

Graves, Dr. Robert J. "Hierarchical Scheduling Approach in Flexible Assembly Systems. "IEEE 1987 International Conference On Robotics and Automation, Vol. 1 (1987) 118-23.

Conterno, R., and Y.C. Ho. "Order Scheduling Problem in Manufacturing Systems." IEEE 1987 International Conference On Robotics and Automation, Vol. 1 (1987) 124-29.

Maimon, Oded Z. and Stanley B. Gershwin. "Dynamic Scheduling and Routing for Flexible Manufacturing Systems." IEEE 1987 International Conference On Robotics and Automation, Vol. 1 (1987) 281-88.

Villa, Agostino. "An Hybrid Knowledge-Based/Analytical Control System For Failure-Prone Manufacturing Systems." IEEE 1987 International Conference On Robotics and Automation, Vol. 1 (1987) 289-94.

Akella, Ram and Bruce Krogh. "Hierarchical Control Structures for Multi-Cell Flexible Assembly System Co-ordination." IEEE 1987 International Conference On Robotics and Automation, Vol. 1 (1987) 295-301.

# APPENDIX ONE

This appendix includes all factory data files listed in TABLE 1. They are listed in the same format that is printed to the stdprn device when hardcopy is requested. The data included in this first appendix represents the factory prior to execution of the simulation.

Files Included:
partspec.dat
machspec.dat
machhist.dat
moldspec.dat
moldhist.dat
schedule.log
partstat.log
machstat.log
moldstat.log

PARTS LIST FILE LISTING

| MODEL | PART NUMBERS | | | | |
|---|---|---|---|---|---|
| 56292 | | | | | |
| | 56471 | 56067 | 70246 | 27689 | 27691 |
| 56466 | | | | | |
| | 60724 | 56067 | 56547 | 57169 | 83185 |
| | 70246 | 27689 | 27691 | | |
| 56539 | | | | | |
| | 60341 | 70246 | 27689 | 62121 | |
| 56542 | | | | | |
| | 60342 | 70246 | 27689 | 27691 | |
| 56582 | | | | | |
| | 61170 | 56067 | 56547 | 70246 | 27689 |
| | 27691 | | | | |
| 56978 | | | | | |
| | 57989 | 57169 | 83185 | 70246 | 27689 |
| | 27691 | | | | |
| 56979 | | | | | |
| | 58602 | 56067 | 57169 | 83185 | 70246 |
| | 27689 | 27691 | | | |
| 57971 | | | | | |
| | 58879 | 56067 | 56547 | 70246 | 27689 |
| | 56159 | 88104 | | | |
| 60868 | | | | | |
| | 60871 | 56067 | 56547 | 57169 | 83185 |
| | 70246 | 27689 | 27691 | | |
| 61000 | | | | | |
| | 62653 | 56067 | 56547 | 57169 | 83185 |
| | 70246 | 64821 | 64822 | 62121 | |
| 61002 | | | | | |
| | 63889 | 56067 | 56547 | 57169 | 83185 |
| | 70246 | 27689 | 62121 | | |
| 61003 | | | | | |
| | 61740 | 56067 | 56547 | 70246 | 27689 |
| | 62121 | | | | |
| 61004 | | | | | |
| | 61607 | 56067 | 56547 | 57169 | 83185 |
| | 70246 | 27689 | 62121 | | |

PARTS LIST FILE LISTING

| MODEL | PART NUMBERS | | | | |
|---|---|---|---|---|---|
| 61005 | | | | | |
| | 62073 | 56067 | 56547 | 57169 | 83185 |
| | 62489 | 62074 | 88104 | 62079 | 62080 |
| 61006 | | | | | |
| | 64511 | 56067 | 56547 | 57169 | 83185 |
| | 62489 | 62074 | 62079 | 62080 | |
| 61007 | | | | | |
| | 61609 | 56067 | 56547 | 57169 | 70246 |
| | 27689 | 62121 | | | |
| 61008 | | | | | |
| | 61548 | 56067 | 56547 | 57169 | 83185 |
| | 70246 | 27689 | 62121 | | |
| 61009 | | | | | |
| | 63717 | 56067 | 56547 | 70246 | 27689 |
| | 62121 | | | | |
| 63010 | | | | | |
| | 63720 | 56067 | 57169 | 83185 | 70246 |
| | 27689 | 27691 | | | |

MOLD MACHINE FILE LISTING

| MACHINE NUMBER | MACHINE SIZE | SERVICE DATE | PRIORITIZED MOLDS 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| 25576 | 220 | 06/27/89 | | | | | |
| 25753 | 375 | 06/27/89 | | | | | |
| 25919 | 375 | 06/27/89 | | | | | |
| 25956 | 375 | 06/27/89 | | | | | |
| 26007 | 375 | 06/27/89 | | | | | |
| 30056 | 220 | 06/27/89 | | | | | |
| 30057 | 375 | 06/27/89 | | | | | |
| 30223 | 375 | 06/27/89 | | | | | |
| 30495 | 375 | 06/27/89 | | | | | |
| 30496 | 375 | 06/27/89 | | | | | |
| 30497 | 375 | 06/27/89 | | | | | |
| 30588 | 375 | 06/27/89 | | | | | |
| 30995 | 375 | 06/27/89 | | | | | |
| 30996 | 375 | 06/27/89 | | | | | |
| 31003 | 220 | 06/27/89 | | | | | |
| 31234 | 75 | 07/16/89 | | | | | |
| 31235 | 75 | 07/16/89 | | | | | |
| 31236 | 75 | 07/16/89 | | | | | |
| 31237 | 75 | 07/16/89 | | | | | |
| 31238 | 75 | 07/16/89 | | | | | |
| 31241 | 75 | 07/16/89 | | | | | |
| 31379 | 375 | 06/27/89 | | | | | |
| 31411 | 220 | 06/27/89 | | | | | |
| 31412 | 220 | 06/27/89 | | | | | |
| 32326 | 220 | 06/27/89 | | | | | |
| 32327 | 220 | 06/27/89 | | | | | |
| 32328 | 220 | 06/27/89 | | | | | |
| 32329 | 220 | 06/27/89 | | | | | |
| 32330 | 220 | 06/27/89 | | | | | |
| 32331 | 220 | 06/27/89 | | | | | |
| 32333 | 220 | 06/27/89 | | | | | |
| 32337 | 375 | 06/27/89 | | | | | |
| 35070 | 375 | 07/16/89 | | | | | |

MOLD MACHINE HISTORY REPORT

| Machine number | down time % | run time % | repair (hrs) | setup (hrs) | MTBF (hrs) | # of observations RUN | SETUP | REPAIR |
|---|---|---|---|---|---|---|---|---|
| 25576 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 25753 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 25919 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 25956 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 26007 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30056 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30057 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30223 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30495 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30496 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30497 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30588 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30995 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30996 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31003 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31234 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31235 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31236 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31237 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31238 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31241 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31379 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31411 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31412 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32326 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32327 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32328 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32329 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32330 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32331 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32333 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32337 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 35070 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |

MOLD TOOLING FILE LISTING

| TOOL NUM8ER | SERVICE DATE | MACH SIZE | NUM8ER OF CAV | 8ASE CYCLETIME |
|---|---|---|---|---|
| 260041 | 07/17/89 | 375 | 2 | 22 |
| | PART#'S AVAILA8LE<br>70246 | | | |
| 260042 | 07/17/89 | 375 | 2 | 22 |
| | PART#'S AVAILA8LE<br>70246 | | | |
| 260043 | 07/17/89 | 375 | 2 | 22 |
| | PART#'S AVAILA8LE<br>70246 | | | |
| 260044 | 07/17/89 | 375 | 2 | 22 |
| | PART#'S AVAILA8LE<br>70246 | | | |
| 260045 | 07/17/89 | 375 | 2 | 22 |
| | PART#'S AVAILA8LE<br>70246 | | | |
| 276891 | 07/17/89 | 220 | 4 | 17 |
| | PART#'S AVAILA8LE<br>27689 | | | |
| 276892 | 07/17/89 | 220 | 4 | 17 |
| | PART#'S AVAILA8LE<br>27689 | | | |
| 276893 | 07/17/89 | 220 | 4 | 17 |
| | PART#'S AVAILA8LE<br>27689 | | | |
| 276911 | 07/17/89 | 220 | 4 | 17 |
| | PART#'S AVAILA8LE<br>27691 | | | |
| 276912 | 07/17/89 | 220 | 4 | 17 |
| | PART#'S AVAILA8LE<br>62121 | | | |
| 276913 | 07/17/89 | 220 | 4 | 17 |
| | PART#'S AVAILA8LE<br>62121 | | | |
| 276914 | 07/17/89 | 220 | 4 | 17 |
| | PART#'S AVAILA8LE<br>62121 | | | |
| 444061 | 07/17/89 | 375 | 4 | 18 |
| | PART#'S AVAILA8LE<br>60341 60342 | | | |

MOLD TOOLING FILE LISTING

| TOOL NUMBER | SERVICE DATE | MACH SIZE | NUMBER OF CAV | BASE CYCLETIME | |
|---|---|---|---|---|---|
| 560671 | 07/19/89 | 220 | 8 | 18 | |
| | PART#'S AVAILABLE | | | | |
| | 56067 | | | | |
| 560672 | 07/17/89 | 220 | 8 | 18 | |
| | PART#'S AVAILABLE | | | | |
| | 56067 | | | | |
| 561591 | 07/17/89 | 220 | 4 | 25 | |
| | PART#'S AVAILABLE | | | | |
| | 56159 | | | | |
| 565471 | 07/17/89 | 75 | 8 | 15 | |
| | PART#'S AVAILABLE | | | | |
| | 56547 | | | | |
| 571691 | 07/17/89 | 75 | 24 | 10 | |
| | PART#'S AVAILABLE | | | | |
| | 57169 | | | | |
| 571692 | 07/17/89 | 75 | 24 | 18 | |
| | PART#'S AVAILABLE | | | | |
| | 57169 | | | | |
| 579711 | 07/17/89 | 375 | 2 | 32 | |
| | PART#'S AVAILABLE | | | | |
| | 58879 | | | | |
| 579B91 | 07/17/89 | 375 | 2 | 20 | |
| | PART#'S AVAILABLE | | | | |
| | 57989 | | | | |
| 586021 | 07/17/89 | 375 | 2 | 20 | |
| | PART#'S AVAILABLE | | | | |
| | 60724 | 58602 | 60871 | 63889 | 61607 61609 |
| | 61548 | 63720 | | | |
| 586022 | 10/10/89 | 375 | 2 | 20 | |
| | PART#'S AVAILABLE | | | | |
| | 63720 | 63889 | | | |
| 588791 | 07/17/89 | 375 | 2 | 20 | |
| | PART#'S AVAILABLE | | | | |
| | 56471 | 61170 | 61740 | 63717 | |
| 620731 | 07/17/89 | 375 | 1 | 30 | |
| | PART#'S AVAILABLE | | | | |
| | 62073 | 64511 | | | |
| 620732 | 07/17/89 | 375 | 1 | 30 | |
| | PART#'S AVAILABLE | | | | |
| | 62073 | 64511 | | | |

MOLD TOOLING FILE LISTING

| TOOL NUMBER | SERVICE DATE | MACH SIZE | NUMBER OF CAV | BASE CYCLETIME |
|---|---|---|---|---|
| 620733 | 07/17/89<br>PART#'S AVAILABLE<br>62073 64511 | 375 | 1 | 30 |
| 620734 | 07/17/89<br>PART#'S AVAILABLE<br>62073 64511 | 375 | 1 | 30 |
| 620735 | 07/17/89<br>PART#'S AVAILABLE<br>62073 64511 | 375 | 1 | 30 |
| 620736 | 07/17/89<br>PART#'S AVAILABLE<br>87136 | 375 | 1 | 30 |
| 620737 | 07/17/89<br>PART#'S AVAILABLE<br>62073 64511 | 375 | 1 | 30 |
| 620741 | 07/17/89<br>PART#'S AVAILABLE<br>62074 | 220 | 8 | 25 |
| 620791 | 07/17/89<br>PART#'S AVAILABLE<br>62079 83262 | 220 | 2 | 22 |
| 620792 | 07/17/89<br>PART#'S AVAILABLE<br>62079 | 375 | 4 | 22 |
| 620801 | 07/17/89<br>PART#'S AVAILABLE<br>62080 | 220 | 2 | 22 |
| 620802 | 07/17/89<br>PART#'S AVAILABLE<br>62080 | 220 | 2 | 22 |
| 620803 | 07/17/89<br>PART#'S AVAILABLE<br>62080 | 375 | 4 | 36 |
| 624B91 | 07/17/89<br>PART#'S AVAILABLE<br>62489 | 220 | 16 | 20 |
| 648211 | 07/17/89<br>PART#'S AVAILABLE<br>64821 | 220 | 4 | 17 |

MOLD TOOLING FILE LISTING

| TOOL NUMBER | SERVICE DATE | MACH SIZE | NUMBER OF CAV | BASE CYCLETIME |
|---|---|---|---|---|
| 648212 | 07/17/89<br>PART#'S AVAILABLE<br>64821 | 220 | 4 | 17 |
| 648221 | 07/17/89<br>PART#'S AVAILABLE<br>64822 | 220 | 8 | 24 |
| 702461 | 07/19/89<br>PART#'S AVAILABLE<br>70246 | 375 | 2 | 20 |
| 702462 | 07/17/89<br>PART#'S AVAILABLE<br>70246 | 375 | 2 | 20 |
| 702463 | 07/17/89<br>PART#'S AVAILABLE<br>70246 | 375 | 2 | 20 |
| 702491 | 07/17/89<br>PART#'S AVAILABLE<br>62653 | 375 | 2 | 20 |
| 831851 | 07/17/89<br>PART#'S AVAILABLE<br>83185 | 220 | 8 | 20 |
| 831852 | 07/17/89<br>PART#'S AVAILABLE<br>83185 | 220 | 8 | 20 |
| 881041 | 07/17/89<br>PART#'S AVAILABLE<br>88104 | 75 | 8 | 24 |

MOLD TOOLING HISTORY REPORT

| Tool number | down time % | run time % | accept % | setup (hrs) | repair (hrs) | MTBF (hrs) | # of observations RUN | SETUP | REPAIR |
|---|---|---|---|---|---|---|---|---|---|
| 260041 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 260042 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 260043 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 260044 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 260045 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276891 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276892 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276893 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276911 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276912 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276913 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276914 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 444061 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 560671 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 560672 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 561591 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 565471 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 571691 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 571692 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 579711 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 579891 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 586021 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 586022 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 588791 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620731 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620732 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620733 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620734 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620735 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620736 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620737 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620741 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620791 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620792 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620801 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620802 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620803 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 624891 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 648211 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 648212 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 648221 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 702461 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 702462 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 702463 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 702491 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 831851 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 831852 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 881041 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |

SCHEDULE FILE LISTING

| MODEL | QUANT | SHIP DATE | SCHEDULED? |
|---|---|---|---|
| 56292 | 5000 | 09/26/89 | 0 |
| 56292 | 6000 | 10/06/89 | 0 |
| 56466 | 5000 | 09/25/89 | 0 |
| 56539 | 5000 | 09/26/89 | 0 |
| 56542 | 5000 | 09/23/89 | 0 |
| 56582 | 5000 | 09/24/89 | 0 |
| 56978 | 5000 | 09/30/89 | 0 |
| 56979 | 5000 | 09/26/89 | 0 |
| 57971 | 5000 | 09/28/89 | 0 |
| 60868 | 5000 | 09/28/89 | 0 |
| 61000 | 5000 | 09/23/89 | 0 |
| 61002 | 5000 | 09/25/89 | 0 |
| 61003 | 5000 | 09/25/89 | 0 |
| 61004 | 5000 | 09/25/89 | 0 |
| 61005 | 5000 | 09/28/89 | 0 |
| 61005 | 4000 | 10/06/89 | 0 |
| 61006 | 5000 | 09/28/89 | 0 |
| 61006 | 7800 | 10/06/89 | 0 |
| 61007 | 5000 | 09/29/89 | 0 |
| 61008 | 5000 | 09/28/89 | 0 |
| 61008 | 3000 | 10/05/89 | 0 |
| 61009 | 5000 | 09/26/89 | 0 |
| 63010 | 5000 | 09/25/89 | 0 |

PART STATUS LISTING

| PART# | INVENTORY | PRODUCTION | DEMAND |
|---|---|---|---|
| 27689 | 0 | 0.00 | 0.00 |
| 27691 | 0 | 0.00 | 0.00 |
| 56067 | 0 | 0.00 | 0.00 |
| 56159 | 0 | 0.00 | 0.00 |
| 56471 | 0 | 0.00 | 0.00 |
| 56547 | 0 | 0.00 | 0.00 |
| 57169 | 0 | 0.00 | 0.00 |
| 57989 | 0 | 0.00 | 0.00 |
| 58602 | 0 | 0.00 | 0.00 |
| 58879 | 0 | 0.00 | 0.00 |
| 60341 | 0 | 0.00 | 0.00 |
| 60342 | 0 | 0.00 | 0.00 |
| 60724 | 0 | 0.00 | 0.00 |
| 60871 | 0 | 0.00 | 0.00 |
| 61170 | 0 | 0.00 | 0.00 |
| 61548 | 0 | 0.00 | 0.00 |
| 61607 | 0 | 0.00 | 0.00 |
| 61609 | 0 | 0.00 | 0.00 |
| 61740 | 0 | 0.00 | 0.00 |
| 62073 | 0 | 0.00 | 0.00 |
| 62074 | 0 | 0.00 | 0.00 |
| 62079 | 0 | 0.00 | 0.00 |
| 62080 | 0 | 0.00 | 0.00 |
| 62121 | 0 | 0.00 | 0.00 |
| 62489 | 0 | 0.00 | 0.00 |
| 62653 | 0 | 0.00 | 0.00 |
| 63717 | 0 | 0.00 | 0.00 |
| 63720 | 0 | 0.00 | 0.00 |
| 63889 | 0 | 0.00 | 0.00 |
| 64511 | 0 | 0.00 | 0.00 |
| 64821 | 0 | 0.00 | 0.00 |
| 64822 | 0 | 0.00 | 0.00 |
| 70246 | 0 | 0.00 | 0.00 |
| 83185 | 0 | 0.00 | 0.00 |
| 88104 | 0 | 0.00 | 0.00 |

MOLD MACHINE STATUS REPORT

| Machine | Mold | Status |
|---|---|---|
| 25576 | 0 | 1 |
| 25753 | 0 | 1 |
| 25919 | 0 | 1 |
| 25956 | 0 | 1 |
| 26007 | 0 | 1 |
| 30056 | 0 | 1 |
| 30057 | 0 | 1 |
| 30223 | 0 | 1 |
| 30495 | 0 | 1 |
| 30496 | 0 | 1 |
| 30497 | 0 | 1 |
| 30588 | 0 | 1 |
| 30995 | 0 | 1 |
| 30996 | 0 | 1 |
| 31003 | 0 | 1 |
| 31234 | 0 | 1 |
| 31235 | 0 | 1 |
| 31236 | 0 | 1 |
| 31237 | 0 | 1 |
| 31238 | 0 | 1 |
| 31241 | 0 | 1 |
| 31379 | 0 | 1 |
| 31411 | 0 | 1 |
| 31412 | 0 | 1 |
| 32326 | 0 | 1 |
| 32327 | 0 | 1 |
| 32328 | 0 | 1 |
| 32329 | 0 | 1 |
| 32330 | 0 | 1 |
| 32331 | 0 | 1 |
| 32333 | 0 | 1 |
| 32337 | 0 | 1 |
| 35070 | 0 | 1 |

MOLD TOOLING STATUS REPORT

| Tool number | Curr part | loc. | status |
|---|---|---|---|
| 260041 | 70246 | 1 | 1 |
| 260042 | 70246 | 1 | 1 |
| 260043 | 70246 | 1 | 1 |
| 260044 | 70246 | 1 | 1 |
| 260045 | 70246 | 1 | 1 |
| 276891 | 27689 | 1 | 1 |
| 276892 | 27689 | 1 | 1 |
| 276893 | 27689 | 1 | 1 |
| 276911 | 27691 | 1 | 1 |
| 276912 | 62121 | 1 | 1 |
| 276913 | 62121 | 1 | 1 |
| 276914 | 62121 | 1 | 1 |
| 444061 | 60341 | 1 | 1 |
| 560671 | 56067 | 1 | 1 |
| 560672 | 56067 | 1 | 1 |
| 561591 | 56159 | 1 | 1 |
| 565471 | 56547 | 1 | 1 |
| 571691 | 57169 | 1 | 1 |
| 571692 | 57169 | 1 | 1 |
| 579711 | 58879 | 1 | 1 |
| 579891 | 57989 | 1 | 1 |
| 586021 | 61548 | 1 | 1 |
| 586022 | 63720 | 1 | 1 |
| 588791 | 56471 | 1 | 1 |
| 620731 | 64511 | 1 | 1 |
| 620732 | 62073 | 1 | 1 |
| 620733 | 62073 | 1 | 1 |
| 620734 | 62073 | 1 | 1 |
| 620735 | 62073 | 1 | 1 |
| 620736 | 87136 | 1 | 1 |
| 620737 | 62073 | 1 | 1 |
| 620741 | 62074 | 1 | 1 |
| 620791 | 62079 | 1 | 1 |
| 620792 | 62079 | 1 | 1 |
| 620801 | 62080 | 1 | 1 |
| 620802 | 62080 | 1 | 1 |
| 620803 | 62080 | 1 | 1 |
| 624891 | 62489 | 1 | 1 |
| 648211 | 64821 | 1 | 1 |
| 648212 | 64821 | 1 | 1 |
| 648221 | 64822 | 1 | 1 |
| 702461 | 70246 | 1 | 1 |
| 702462 | 70246 | 1 | 1 |
| 702463 | 70246 | 1 | 1 |
| 702491 | 62653 | 1 | 1 |
| 831851 | 83185 | 1 | 1 |
| 831852 | 83185 | 1 | 1 |
| 881041 | 88104 | 1 | 1 |

# APPENDIX TWO

This appendix includes all factory data files listed in Appendix One, with the exception of the specification files as they are not changed by the simulation. The data included in this second appendix represents the factory after the execution of the simulation. Schedule.log is not included since the file is empty after the execution of the simulator.

MOLD MACHINE HISTORY REPORT

| Machine number | down time % | run time % | repair (hrs) | setup (hrs) | MTBF (hrs) | # of observations RUN | SETUP | REPAIR |
|---|---|---|---|---|---|---|---|---|
| 25576 | 6.396 | 93.604 | 1.51 | 2.46 | 36.0 | 15 | 15 | 12 |
| 25753 | 8.940 | 91.060 | 1.74 | 2.69 | 27.4 | 18 | 18 | 17 |
| 25919 | 6.928 | 93.072 | 1.73 | 2.33 | 31.3 | 17 | 17 | 15 |
| 25956 | 7.692 | 92.308 | 2.30 | 2.50 | 30.0 | 20 | 15 | 10 |
| 26007 | 6.014 | 93.986 | 1.46 | 2.15 | 33.6 | 15 | 15 | 13 |
| 30056 | 5.966 | 94.034 | 1.31 | 2.43 | 38.3 | 13 | 13 | 11 |
| 30057 | 8.080 | 91.920 | 2.23 | 2.98 | 33.9 | 20 | 13 | 12 |
| 30223 | 7.692 | 92.308 | 2.00 | 2.50 | 30.0 | 11 | 11 | 10 |
| 30495 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30496 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30497 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30588 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30995 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 30996 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31003 | 6.192 | 93.808 | 1.58 | 2.35 | 35.6 | 15 | 15 | 12 |
| 31234 | 10.758 | 89.242 | 1.96 | 3.05 | 25.3 | 16 | 16 | 14 |
| 31235 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 11 | 11 | 10 |
| 31236 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31237 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31238 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31241 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31379 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 31411 | 8.322 | 91.678 | 2.17 | 2.36 | 26.0 | 15 | 15 | 14 |
| 31412 | 9.504 | 90.496 | 2.21 | 2.51 | 23.9 | 18 | 18 | 15 |
| 32326 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32327 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32328 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32329 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32330 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32331 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32333 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 32337 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |
| 35070 | 7.692 | 92.308 | 1.50 | 2.50 | 30.0 | 10 | 10 | 10 |

MOLD TOOLING HISTORY REPORT

| Tool number | down time % | run time % | accept % | setup (hrs) | repair (hrs) | MT8F (hrs) | # of observations RUN | SETUP | REPAIR |
|---|---|---|---|---|---|---|---|---|---|
| 260041 | 4.819 | 95.181 | 95.000 | 1.07 | 1.62 | 32.0 | 14 | 14 | 13 |
| 260042 | 5.660 | 94.340 | 95.000 | 1.32 | 1.50 | 25.0 | 11 | 11 | 10 |
| 260043 | 6.287 | 93.713 | 95.000 | 1.00 | 1.61 | 24.0 | 14 | 14 | 13 |
| 260044 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 260045 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276891 | 5.241 | 94.759 | 95.000 | 1.10 | 1.51 | 27.3 | 14 | 14 | 13 |
| 276892 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276893 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276911 | 5.364 | 94.636 | 95.000 | 1.05 | 1.57 | 27.7 | 12 | 12 | 11 |
| 276912 | 5.582 | 94.418 | 95.000 | 1.00 | 1.62 | 27.4 | 12 | 12 | 11 |
| 276913 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 276914 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 444061 | 7.615 | 92.385 | 95.000 | 1.13 | 1.97 | 23.9 | 13 | 13 | 11 |
| 560671 | 4.818 | 95.182 | 95.000 | 1.30 | 1.22 | 24.1 | 14 | 14 | 13 |
| 560672 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 561591 | 5.660 | 94.340 | 95.000 | 1.17 | 1.50 | 25.0 | 11 | 11 | 10 |
| 565471 | 5.484 | 94.516 | 95.000 | 1.68 | 1.41 | 24.3 | 13 | 13 | 12 |
| 571691 | 5.660 | 94.340 | 95.000 | 1.04 | 1.50 | 25.0 | 11 | 11 | 10 |
| 571692 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 579711 | 5.660 | 94.340 | 95.000 | 1.08 | 1.50 | 25.0 | 11 | 11 | 10 |
| 579891 | 5.660 | 94.340 | 95.000 | 1.55 | 1.50 | 25.0 | 11 | 11 | 10 |
| 586021 | 5.517 | 94.483 | 95.000 | 0.95 | 1.60 | 27.4 | 21 | 21 | 13 |
| 586022 | 5.660 | 94.340 | 95.000 | 1.54 | 1.50 | 25.0 | 11 | 11 | 10 |
| 588791 | 4.653 | 95.347 | 95.000 | 0.94 | 1.43 | 29.3 | 16 | 16 | 11 |
| 620731 | 7.171 | 92.829 | 95.000 | 1.15 | 1.97 | 25.5 | 15 | 15 | 14 |
| 620732 | 5.660 | 94.340 | 95.000 | 1.10 | 1.50 | 25.0 | 11 | 11 | 10 |
| 620733 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620734 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620735 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620736 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620737 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620741 | 5.660 | 94.340 | 95.000 | 1.02 | 1.50 | 25.0 | 11 | 11 | 10 |
| 620791 | 5.660 | 94.340 | 95.000 | 0.94 | 1.50 | 25.0 | 11 | 11 | 10 |
| 620792 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620801 | 5.132 | 94.868 | 95.000 | 0.96 | 1.32 | 24.4 | 14 | 14 | 13 |
| 620802 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 620803 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 624891 | 5.660 | 94.340 | 95.000 | 1.02 | 1.50 | 25.0 | 11 | 11 | 10 |
| 648211 | 5.660 | 94.340 | 95.000 | 1.01 | 1.50 | 25.0 | 11 | 11 | 10 |
| 648212 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 648221 | 5.660 | 94.340 | 95.000 | 0.96 | 1.50 | 25.0 | 11 | 11 | 10 |
| 702461 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 702462 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 702463 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 702491 | 6.396 | 93.604 | 95.000 | 1.04 | 1.64 | 24.0 | 12 | 12 | 11 |
| 831851 | 5.660 | 94.340 | 95.000 | 1.10 | 1.50 | 25.0 | 11 | 11 | 10 |
| 831852 | 5.660 | 94.340 | 95.000 | 1.00 | 1.50 | 25.0 | 10 | 10 | 10 |
| 881041 | 5.743 | 94.257 | 95.000 | 1.17 | 1.45 | 23.8 | 12 | 12 | 11 |

PART STATUS LISTING

| PART# | INVENTORY | PRODUCTION | DEMAND |
|---|---|---|---|
| 27689 | 25 | 0.00 | 0.00 |
| 27691 | 37 | 0.00 | 0.00 |
| 56067 | 0 | 0.00 | 0.00 |
| 56159 | 11 | 0.00 | 0.00 |
| 56471 | 16 | 0.00 | 0.00 |
| 56547 | 72 | 0.00 | 0.00 |
| 57169 | 800 | 0.00 | 0.00 |
| 57989 | 4 | 0.00 | 0.00 |
| 58602 | 4 | 0.00 | 0.00 |
| 58879 | 17 | 0.00 | 0.00 |
| 60341 | 40 | 0.00 | 0.00 |
| 60342 | 0 | 0.00 | 0.00 |
| 60724 | 4 | 0.00 | 0.00 |
| 60871 | 4 | 0.00 | 0.00 |
| 61170 | 4 | 0.00 | 0.00 |
| 61548 | 28 | 0.00 | 0.00 |
| 61607 | 4 | 0.00 | 0.00 |
| 61609 | 4 | 0.00 | 0.00 |
| 61740 | 4 | 0.00 | 0.00 |
| 62073 | 0 | 0.00 | 0.00 |
| 62074 | 89 | 0.00 | 0.00 |
| 62079 | 29 | 0.00 | 0.00 |
| 62080 | 29 | 0.00 | 0.00 |
| 62121 | 72 | 0.00 | 0.00 |
| 62489 | 88 | 0.00 | 0.00 |
| 62653 | 4 | 0.00 | 0.00 |
| 63717 | 4 | 0.00 | 0.00 |
| 63720 | 4 | 0.00 | 0.00 |
| 63889 | 4 | 0.00 | 0.00 |
| 64511 | 4 | 0.00 | 0.00 |
| 64821 | 82 | 0.00 | 0.00 |
| 64822 | 40 | 0.00 | 0.00 |
| 70246 | 58 | 0.00 | 0.00 |
| 83185 | 40 | 0.00 | 0.00 |
| 88104 | 40 | 0.00 | 0.00 |

MOLD MACHINE STATUS REPORT

| Machine | Mold | Status |
| --- | --- | --- |
| 25576 | 624891 | 1 |
| 25753 | 260041 | 1 |
| 25919 | 620731 | 1 |
| 25956 | 579891 | 1 |
| 26007 | 620732 | 1 |
| 30056 | 620791 | 1 |
| 30057 | 586021 | 1 |
| 30223 | 586022 | 1 |
| 30495 | 0 | 1 |
| 30496 | 0 | 1 |
| 30497 | 0 | 1 |
| 30588 | 0 | 1 |
| 30995 | 0 | 1 |
| 30996 | 0 | 1 |
| 31003 | 561591 | 1 |
| 31234 | 881041 | 1 |
| 31235 | 571691 | 1 |
| 31236 | 0 | 1 |
| 31237 | 0 | 1 |
| 31238 | 0 | 1 |
| 31241 | 0 | 1 |
| 31379 | 0 | 1 |
| 31411 | 560671 | 1 |
| 31412 | 831851 | 1 |
| 32326 | 0 | 1 |
| 32327 | 0 | 1 |
| 32328 | 0 | 1 |
| 32329 | 0 | 1 |
| 32330 | 0 | 1 |
| 32331 | 0 | 1 |
| 32333 | 0 | 1 |
| 32337 | 0 | 1 |
| 35070 | 0 | 1 |

MOLD TOOLING STATUS REPORT

| Tool number | Curr part | loc. | status |
|---|---|---|---|
| 260041 | 70246 | 0 | 1 |
| 260042 | 70246 | 1 | 1 |
| 260043 | 70246 | 1 | 1 |
| 260044 | 70246 | 1 | 1 |
| 260045 | 70246 | 1 | 1 |
| 276891 | 27689 | 1 | 1 |
| 276892 | 27689 | 1 | 1 |
| 276893 | 27689 | 1 | 1 |
| 276911 | 27691 | 1 | 1 |
| 276912 | 62121 | 1 | 1 |
| 276913 | 62121 | 1 | 1 |
| 276914 | 62121 | 1 | 1 |
| 444061 | 60341 | 1 | 1 |
| 560671 | 56067 | 0 | 1 |
| 560672 | 56067 | 1 | 1 |
| 561591 | 56159 | 0 | 1 |
| 565471 | 56547 | 1 | 1 |
| 571691 | 57169 | 0 | 1 |
| 571692 | 57169 | 1 | 1 |
| 579711 | 58879 | 1 | 1 |
| 579891 | 57989 | 0 | 1 |
| 586021 | 61548 | 0 | 1 |
| 586022 | 63720 | 0 | 1 |
| 588791 | 56471 | 1 | 1 |
| 620731 | 64511 | 0 | 1 |
| 620732 | 62073 | 0 | 1 |
| 620733 | 62073 | 1 | 1 |
| 620734 | 62073 | 1 | 1 |
| 620735 | 62073 | 1 | 1 |
| 620736 | 87136 | 1 | 1 |
| 620737 | 62073 | 1 | 1 |
| 620741 | 62074 | 1 | 1 |
| 620791 | 62079 | 0 | 1 |
| 620792 | 62079 | 1 | 1 |
| 620801 | 62080 | 1 | 1 |
| 620802 | 62080 | 1 | 1 |
| 620803 | 62080 | 1 | 1 |
| 624891 | 62489 | 0 | 1 |
| 648211 | 64821 | 1 | 1 |
| 648212 | 64821 | 1 | 1 |
| 648221 | 64822 | 1 | 1 |
| 702461 | 70246 | 1 | 1 |
| 702462 | 70246 | 1 | 1 |
| 702463 | 70246 | 1 | 1 |
| 702491 | 62653 | 1 | 1 |
| 831851 | 83185 | 0 | 1 |
| 831852 | 83185 | 1 | 1 |
| 881041 | 88104 | 0 | 1 |

# APPENDIX THREE

This appendix includes two versions of the schedule. The first file listed is machine.sch, this is the schedule that is determined in the scheduler. The second file is results.sch, this file shows how the schedule appears after the simulation. Both of these files are a combination of jobbase.sch, moldbase.sch and machbase.sch. The files are shown with only two machines for the sake of keeping the appendix smaller.

File: MACHINE.SCH

```
>>>>>>>>>- MOLD MACHINE -<<<<<<<<<
----------------------------------
25576     622389600 622978272 0
          [------ MOLD to RUN -----]
          276891    0        422038    0        1        1        89000
                    [----- PARTS to RUN -----]
                     27689 622537200 0    5000        0 0.0339 0   1 1
                     27689 622623600 0    5000        0 0.0214 0   2 1
                     27689 622710000 0    5000        0 0.0156 0   3 1
                     27689 622710000 0    5000        0 0.0156 0   4 1
                     27689 622710000 0    5000        0 0.0156 0   5 1
                     27689 622710000 0    5000        0 0.0156 0   6 1
                     27689 622710000 0    5000        0 0.0156 0   7 1
                     27689 622796400 0    5000        0 0.0123 0   8 1
                     27689 622796400 0    5000        0 0.0123 0   9 1
                     27689 622796400 0    5000        0 0.0123 0  10 1
                     27689 622796400 0    5000        0 0.0123 0  11 1
                     27689 622969200 0    5000        0 0.0086 0  12 1
                     27689 622969200 0    5000        0 0.0086 0  13 1
                     27689 622969200 0    5000        0 0.0086 0  14 1
                     27689 623055600 0    5000        0 0.0075 0  15 1
                     27689 623142000 0    5000        0 0.0066 0  16 1
                     27689 623574000 0    3000        0 0.0025 0  17 1
                     27689 623660400 0    6000        0 0.0047 0  18 1
          [------ MOLD to RUN -----]
          620741    0        76012     0        2        1        21800
                    [----- PARTS to RUN -----]
                     62074 622969200 0    5000        0 0.0086 0   1 1
                     62074 622969200 0    5000        0 0.0086 0   2 1
                     62074 623660400 0    7800        0 0.0061 0   3 1
                     62074 623660400 0    4000        0 0.0031 0   4 1
          [------ MOLD to RUN -----]
          624891    0        30402     0        3        1        21800
                    [----- PARTS to RUN -----]
                     62489 622969200 0    5000        0 0.0086 0   1 1
                     62489 622969200 0    5000        0 0.0086 0   2 1
                     62489 623660400 0    7800        0 0.0061 0   3 1
                62489 623660400 0    4000        0 0.0031 0   4 1

>>>>>>>>>- MOLD MACHINE -<<<<<<<<<
----------------------------------
25753     622389600 623113005 0
          [------ MOLD to RUN -----]
          260041    0        662775    0        1        1        54000
                    [----- PARTS to RUN -----]
                     70246 622537200 0    5000        0 0.0339 0   1 1
                     70246 622623600 0    5000        0 0.0214 0   2 1
                     70246 622710000 0    5000        0 0.0156 0   3 1
                     70246 622796400 0    5000        0 0.0123 0   4 1
                     70246 622969200 0    5000        0 0.0086 0   5 1
                     70246 622969200 0    5000        0 0.0086 0   6 1
                     70246 622969200 0    5000        0 0.0086 0   7 1
                     70246 623055600 0    5000        0 0.0075 0   8 1
                     70246 623142000 0    5000        0 0.0066 0   9 1
                     70246 623574000 0    3000        0 0.0025 0  10 1
                     70246 623660400 0    6000        0 0.0047 0  11 1
```

File: RESULT.SCH

```
>>>>>>>>>- MOLD MACHINE -<<<<<<<<<
---------------------------------
25576     622389600 622978272 3
          [------ MOLD to RUN -----]
          276891   0      422038   18      1       1       0
                   [----- PARTS to RUN -----]
                    27689 622537200 622420560    0   5000 0.0339 0  1 1
                    27689 622623600 622441800    0   5000 0.0214 0  2 1
                    27689 622710000 622463040    0   5000 0.0156 0  3 1
                    27689 622710000 622484280    0   5000 0.0156 0  4 1
                    27689 622710000 622505520    0   5000 0.0156 0  5 1
                    27689 622710000 622536120    0   5000 0.0156 0  6 1
                    27689 622710000 622580040    0   5000 0.0156 0  7 1
                    27689 622796400 622601280    0   5000 0.0123 0  8 1
                    27689 622796400 622622520    0   5000 0.0123 0  9 1
                    27689 622796400 622643760    0   5000 0.0123 0 10 1
                    27689 622796400 622665000    0   5000 0.0123 0 11 1
                    27689 622969200 622693440    0   5000 0.0086 0 12 1
                    27689 622969200 622736280    0   5000 0.0086 0 13 1
                    27689 622969200 622757520    0   5000 0.0086 0 14 1
                    27689 623055600 622778760    0   5000 0.0075 0 15 1
                    27689 623142000 622800000    0   5000 0.0066 0 16 1
                    27689 623574000 622812600    0   3000 0.0025 0 17 1
                    27689 623660400 622838160  -25   6025 0.0047 0 18 1
          [------ MOLD to RUN -----]
          620741   0      76012    4       2       1       0
                   [----- PARTS to RUN -----]
                    62074 622969200 622858680    0   5000 0.0086 0  1 1
                    62074 622969200 622874160    0   5000 0.0086 0  2 1
                    62074 623660400 622898640    0   7800 0.0061 0  3 1
                    62074 623660400 622911240  -89   4089 0.0031 0  4 1
          [------ MOLD to RUN -----]
          624891   0      30402    4       3       1       0
                   [----- PARTS to RUN -----]
                    62489 622969200 622922400    0   5000 0.0086 0  1 1
                    62489 622969200 622928520    0   5000 0.0086 0  2 1
                    62489 623660400 622938240    0   7800 0.0061 0  3 1
                    62489 623660400 622943280  -88   4088 0.0031 0  4 1

>>>>>>>>>- MOLD MACHINE -<<<<<<<<<
---------------------------------
25753     622389600 623113005 1
          [------ MOLD to RUN -----]
          260041   0      662775   11      1       1       0
                   [----- PARTS to RUN -----]
                    70246 622537200 622487880    0   5000 0.0339 0  1 1
                    70246 622623600 622560600    0   5000 0.0214 0  2 1
                    70246 622710000 622630800    0   5000 0.0156 0  3 1
                    70246 622796400 622685880    0   5000 0.0123 0  4 1
                    70246 622969200 622750680    0   5000 0.0086 0  5 1
                    70246 622969200 622827360    0   5000 0.0086 0  6 1
                    70246 622969200 622897920    0   5000 0.0086 0  7 1
                    70246 623055600 622979640    0   5000 0.0075 0  8 1
                    70246 623142000 623059200    0   5000 0.0066 0  9 1
                    70246 623574000 623092320    0   3000 0.0025 0 10 1
                    70246 623660400 623158200    0   6000 0.0047 0 11 1
```

# APPENDIX FOUR

This appendix includes the simulation performance printout that is available after the simulation has been exited. The machines shown here are the same machines that a schedule is shown for in Appendix Three. Also included in this appendix are a sample simulation final report and a schedule shipment report.

## MACHINE / MOLD PERFORMANCE REPORT

MACH # 25576 # of Molds 3

| | RUN | IDLE | SETUP | REPAIR | DOWN |
|---|---|---|---|---|---|
| hours | 126.4 | 30.8 | 17.1 | 11.3 | 0.4 |
| #obs | 7 | | 6 | 4 | |
| % | 81.47 | 83.46 | 10.99 | 7.28 | 0.26 |

| | MTBF | MTTR | SETUP |
|---|---|---|---|
| simul | 31.60 | 2.83 | 2.84 |
| hist. | 30.00 | 2.50 | 1.50 |
| new | 30.46 | 2.59 | 2.00 |

MOLD # 276891 # of Parts 84028

| | RUN | IDLE | SETUP | REPAIR | DOWN |
|---|---|---|---|---|---|
| hours | 99.7 | 21.3 | 3.0 | 2.4 | 0.2 |
| #obs | 3 | | 3 | 2 | |
| % | 94.64 | 83.21 | 2.90 | 2.28 | 0.19 |

| | MTBF | MTTR | SETUP |
|---|---|---|---|
| simul | 49.85 | 1.20 | 1.02 |
| hist. | 25.00 | 1.50 | 1.00 |
| new | 29.14 | 1.45 | 1.00 |

MOLD # 620741 # of Parts 21888

| | RUN | IDLE | SETUP | REPAIR | DOWN |
|---|---|---|---|---|---|
| hours | 18.9 | 0.0 | 0.9 | 0.0 | 0.0 |
| #obs | 1 | | 1 | 0 | |
| % | 95.45 | 100.00 | 4.55 | 0.00 | 0.00 |

| | MTBF | MTTR | SETUP |
|---|---|---|---|
| simul | -1.00 | -1.00 | 0.90 |
| hist. | 25.00 | 1.50 | 1.00 |
| new | 25.00 | 1.50 | 0.99 |

MOLD # 624891 # of Parts 21888

| | RUN | IDLE | SETUP | REPAIR | DOWN |
|---|---|---|---|---|---|
| hours | 7.5 | 0.0 | 5.9 | 0.0 | 0.0 |
| #obs | 1 | | 1 | 0 | |
| % | 55.97 | 100.00 | 44.03 | 0.00 | 0.00 |

| | MTBF | MTTR | SETUP |
|---|---|---|---|
| simul | -1.00 | -1.00 | 5.90 |
| hist. | 25.00 | 1.50 | 1.00 |
| new | 25.00 | 1.50 | 1.45 |

MACH # 25753 # of Molds 1

| | RUN | IDLE | SETUP | REPAIR | DOWN |
|---|---|---|---|---|---|
| hours | 164.6 | 4.7 | 4.2 | 12.1 | 0.4 |
| #obs | 5 | | 4 | 4 | |
| % | 90.79 | 97.50 | 2.32 | 6.68 | 0.22 |

| | MTBF | MTTR | SETUP |
|---|---|---|---|
| simul | 41.14 | 3.02 | 1.05 |
| hist. | 30.00 | 2.50 | 1.50 |
| new | 33.18 | 2.65 | 1.37 |

MOLD # 260041 # of Parts 53656

| | RUN | IDLE | SETUP | REPAIR | DOWN |
|---|---|---|---|---|---|
| hours | 164.4 | 16.6 | 0.9 | 1.8 | 0.2 |
| #obs | 3 | | 3 | 2 | |
| % | 98.27 | 91.00 | 0.54 | 1.08 | 0.12 |

| | MTBF | MTTR | SETUP |
|---|---|---|---|
| simul | 82.23 | 0.90 | 0.30 |
| hist. | 25.00 | 1.50 | 1.00 |
| new | 34.54 | 1.40 | 0.84 |

Final Simulation Screen Report

## SIMULATION FINAL REPORT

| | |
|---|---|
| Real Start Time | 17:32:50 |
| Real Finish Time | 17:38:29 |
| Simulated Start Time | September 21,1989 07:00 |
| Simulated Finish Time | September 29 1989 00:48 |
| Required Number of Parts | 609721 |
| On Time Parts | 609721 |
| Late Parts | 0 |
| Non-scheduled Parts | 0 |
| Excess Parts | 1100 |
| Attainment | 100.00 percent |

Example schedule shipment printout

```
Model 56542 shipped  9/21/89 23:54 req'd 09/23/89
Model 56466 shipped  9/22/89 02:54 req'd 09/25/89
Model 63010 shipped  9/22/89 06:24 req'd 09/25/89
Model 56539 shipped  9/22/89 12:12 req'd 09/26/89
Model 61004 shipped  9/22/89 18:30 req'd 09/25/89
Model 61000 shipped  9/23/89 03:00 req'd 09/23/89
Model 61002 shipped  9/23/89 06:18 req'd 09/25/89
Model 56582 shipped  9/23/89 20:00 req'd 09/24/89
Model 56979 shipped  9/24/89 02:18 req'd 09/26/89
Model 61003 shipped  9/24/89 10:12 req'd 09/25/89
Model 56292 shipped  9/25/89 02:12 req'd 09/26/89
Model 60868 shipped  9/25/89 05:42 req'd 09/28/89
Model 61009 shipped  9/25/89 16:30 req'd 09/26/89
Model 61008 shipped  9/26/89 08:30 req'd 09/28/89
Model 61008 shipped  9/27/89 02:06 req'd 10/05/89
Model 61005 shipped  9/27/89 11:06 req'd 09/28/89
Model 61006 shipped  9/27/89 12:48 req'd 09/28/89
Model 57971 shipped  9/27/89 13:18 req'd 09/28/89
Model 61005 shipped  9/27/89 19:12 req'd 10/06/89
Model 56978 shipped  9/28/89 17:12 req'd 09/30/89
Model 61007 shipped  9/29/89 10:06 req'd 09/29/89
Model 61006 shipped  9/30/89 03:24 req'd 10/06/89
Model 56292 shipped  9/30/89 04:30 req'd 10/06/89
```