MODELLING, ANALYSIS AND DESIGN OF COMPUTER INTEGRATED MANUFACTURING SYSTEMS

Volume II of II

ABDULRAHMAN MUSHABAB ABDULLAH AL-AHMARI

October-1998

A thesis submitted for the
DEGREE OF DOCTOR OF PHILOSOPHY
MECHANICAL ENGINEERING DEPARTMENT, THE UNIVERSITY OF SHEFFIELD

CONTENTS

Volume II

APPENDIX-A

COMPUTER INTEGRATED MANUFACTURING COMPONENTS

A.1 Computer Aided Design

The study of integrated manufacturing systems involves all the major components of these systems, of which CAD is the first (Hatvany et al. 1977). Before defining the CAD system, the word 'design' should be defined. Hatvany et al. (1977) defined design as the set of activities leading from the establishment of a product requirement to the production of the information necessary for making the product. Main and Ward (1992) reported that some confusion seems to exist as to what CAD system actually means. A CAD system can be defined as any design activity that involves the effective use of the computer technology to create, modify or document an engineering design (Gunasekaran et al. 1994). Bedworth et al. (1991) specified CAD system as the reaction and optimisation of the design itself using the computer technology as a productivity tool.

The first interactive computer graphics appeared in 1963 as sketchpad by I.E Sutherland at MIT (Hitomi 1996). Before more discussion about CAD system, it would beneficial to understand the design process.

A.1.1 The Design Process

The design process has been defined by several contributors; hence, it is viewed differently by different people. Bedworth et al. (1991) and Groover and Zimmers (1984) defined the design process in six steps:

1. Recognition of needs;
2. Definition of problem;
3. Synthesis;
4. Analysis and optimisation;

5. Evaluation;
6. Presentation;

The hierarchy of the design process steps is illustrated in Figure A.4.

Hatvany et al. (1977) defined the design process in four other steps:

1. Conceptual design. This step starts with the specifications of a product from market or customer considerations including creative innovation and management decision process leading to the evaluation of product concepts.
2. Design analysis. In this step, analysis techniques are used to predict theoretically the expected performance of the product.
3. Detail design. This step in concerned with the design of all parts and sub-assemblies required for producing the product.
4. Design documentation. This step connects the results of the previous activities to other manufacturing functions such as scheduling and costs.

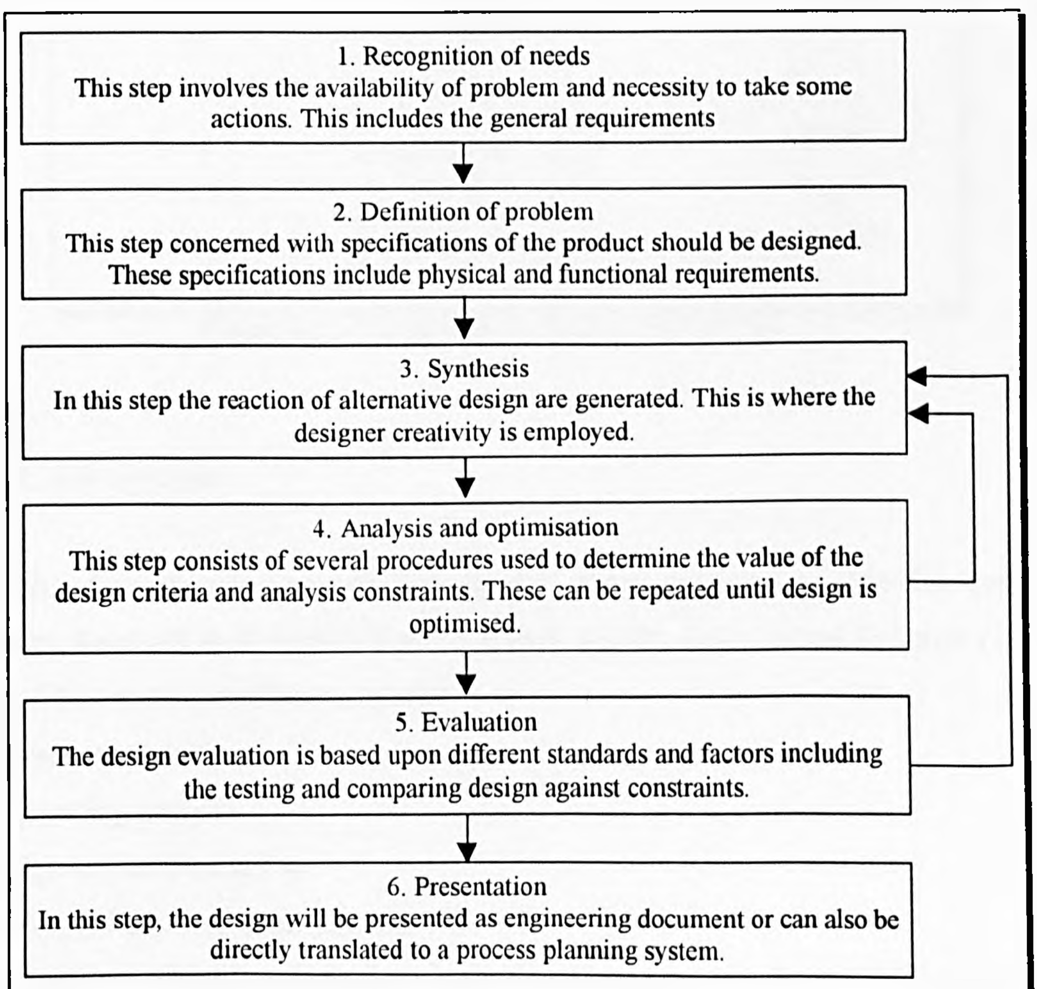


Figure A.4. The general steps of the design process

Singh (1996) also presented the design process in six steps:

1. Problem identification
2. Preliminarily ideas
3. Refinement process
4. Analysis process
5. Decision process
6. Implementation

The relationships between these steps are illustrated in Figure A.5.

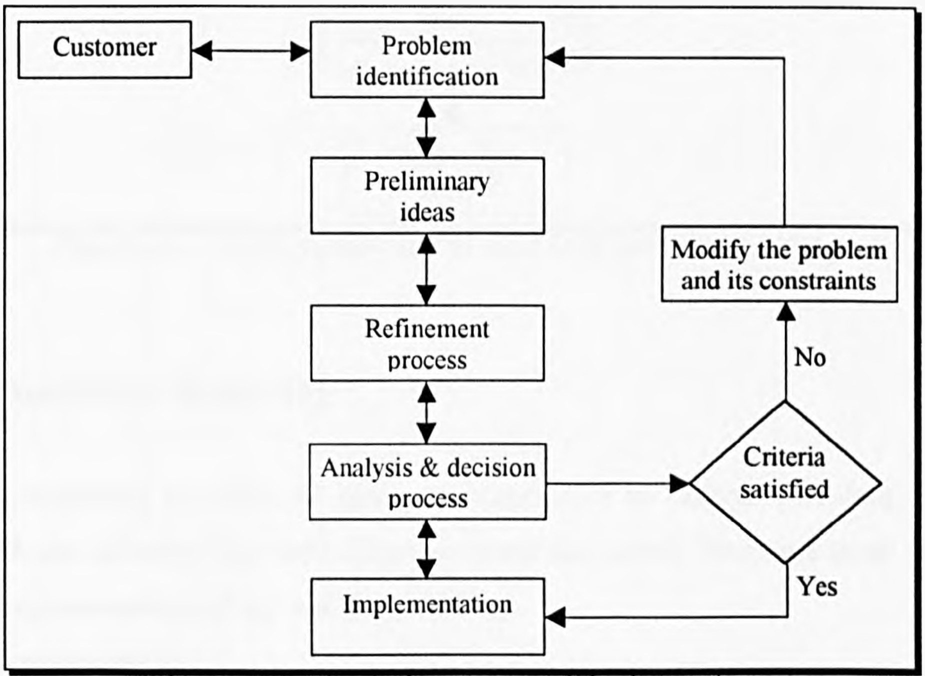


Figure A.5. The design Process (Singh 1996).

A.1.2 CAD System

As result of using the computer for design, many engineering activities can be automated. Rembold et al. (1993), Bedworth et al. (1991), Groover and Zimmers (1984) divided CAD system into four categories:

1. Geometric modelling
2. Engineering analysis
3. Design review/evaluation
4. Automated drafting/documentation

These categories of CAD are based upon the functional concept. Singh (1996) presented a classification scheme of CAD system based upon the hardware of the system, as shown in Figure A.6.

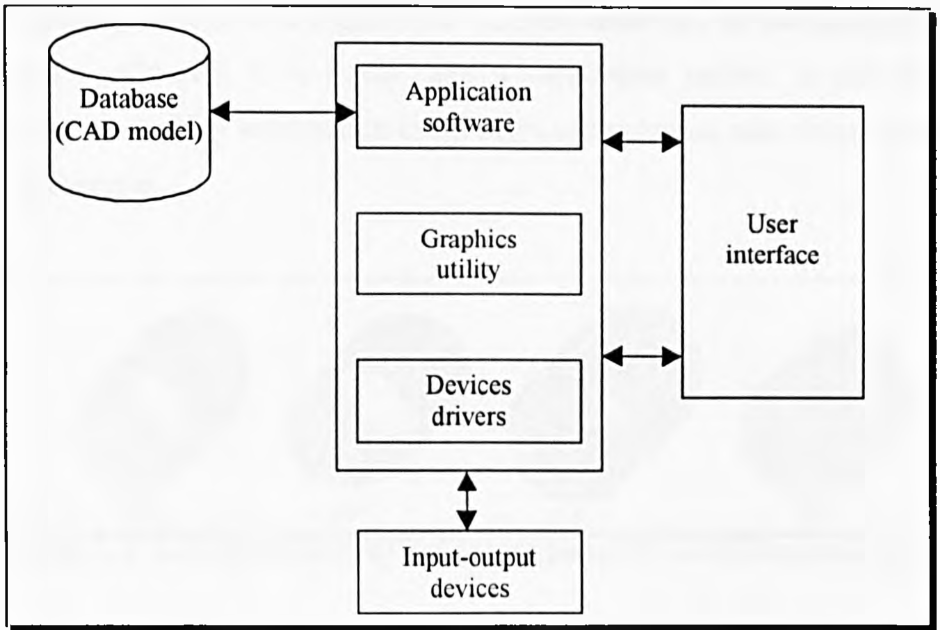


Figure A.6. Basic architecture of a CAD system (Singh 1996).

A.1.2.1 Geometric Modelling

Geometric modelling presents the geometric data about an object. This data includes type of surfaces and edges and their dimensions and tolerances. There are three different computer representations of geometric modelling:

1. Wire-frame model
2. Surface model
3. Volume (solid) model

A Wire-frame model describes the object shape in the form of interconnected lines lying along the edges of the object, as shown in Figure A.7.

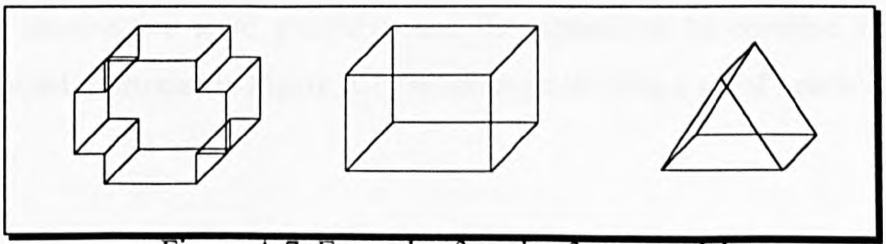


Figure A.7. Examples for wire-frame models.

The wire-frame model does not provide information on the properties of the object, such as mass, surfaces or volumes.

A Surface model uses the points and lines of a wire-frame and describes the objects by means of surfaces, as shown in Figure A.8. Surface areas can be calculated from this type of model. Although it is better than a wire-frame model, it still has some limitations such as lacking information about mass and volumes, and cannot be used for engineering analysis.

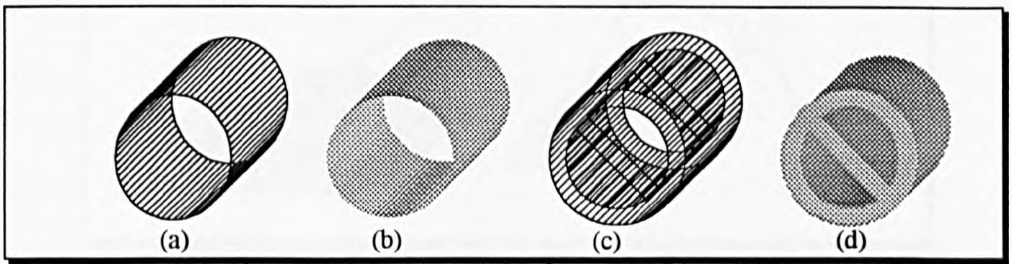


Figure A.8. Surface models (a) and (c) for parts (b) and(d) respectively.

A Volume model describes an object in its entirety and gives the designer a complete description of construct, shape, surface, volume and density. This type forms the object with the building blocks of elementary solid shapes called primitives (Milner and Vasiliou 1986). Figure A.9 illustrates a volume model example.

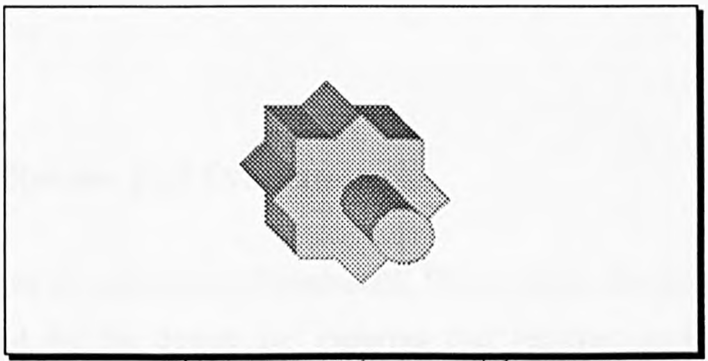


Figure A.9. Volume model

Cowan (1986) divided the volume model approaches into two main descriptive methods: constructive solid geometry and the operations to combine objects. The volume model illustrated in Figure A.10 is constructed using a set of operations.

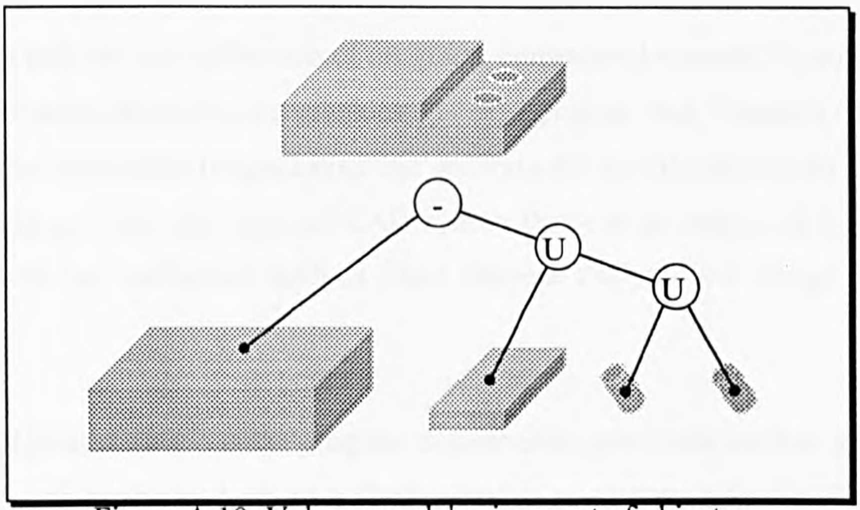


Figure A.10. Volume model using a set of objects.

A.1.2.2 Engineering Analysis

Product design usually requires types of analysis. This analysis may involve various engineering measures such as static, dynamic, stress-strain calculations and heat transfer computation. Finite element analysis techniques are the most frequently used in engineering analysis.

A.1.2.3 Design Review and Evaluation

In this stage, designs are reviewed and evaluated. This involves checking specifications that are established for the design and ensuring that required goals of design are achieved. Several constraints should be taken into account in this stage, such as design constraints, customer constraints and organisation constraints.

A.1.2.4 Automated Drafting/Documentation

In this stage, CAD system produces the detailed engineering drawing and information. These drawings and documentation will be used by other manufacturing functions such as process planning and manufacturing operations in order to complete the product production cycle.

A.1.3 CAD Software

The graphics software is a collection of programs constructed to make it convenient for the user to operate the computer graphics system (Groover and Zimmers 1984). The graphics software includes the packages that generate 2D and 3D models on the display devices. These are other packages of CAD system that can be integrated and used for design analysis and evaluation such as finite element analysis and design simulation packages.

The CAD graphics software is the program that receives primitives such as points, lines and surfaces, and produces them on a display device or an output device (Bedworth et al. 1991). Figure A.11 shows the general structure of CAD software.

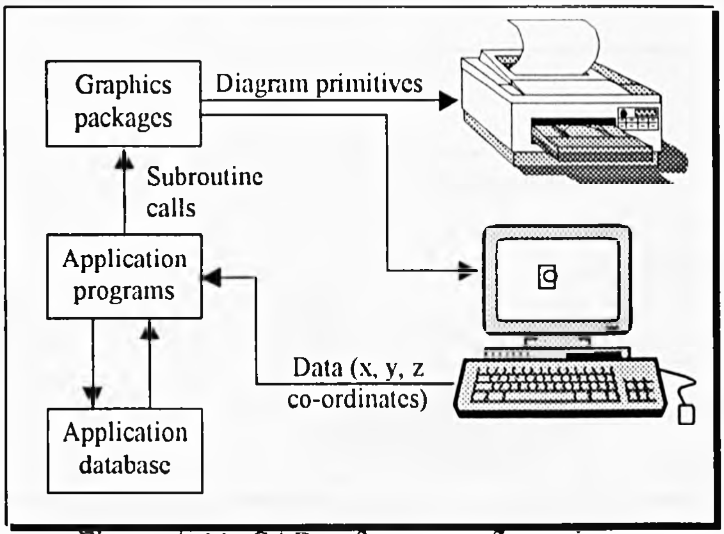


Figure A.11. CAD software configuration.

As illustrated in the figure above, the user provides design data using CAD input devices. Then data enters the application program in terms of numbers indicating geometry data values or other information. The application program links with the graphics packages using particular subroutine calls and refers to the CAD database if necessary. The graphics package is software that links between the application program and the graphics terminal or output devices. It generates output primitives in terms of drawing commands or any other forms of data which are understood by the display and output devices.

A.1.4 CAD Hardware

Hardware components for CAD are available in a variety of specifications; hence a CAD system can be configured in many ways. In general, the CAD system is based upon interactive computer graphics (ICG) which provides an immediate response to input carried out by the users. Figure A.12 illustrates the general configuration of CAD hardware.

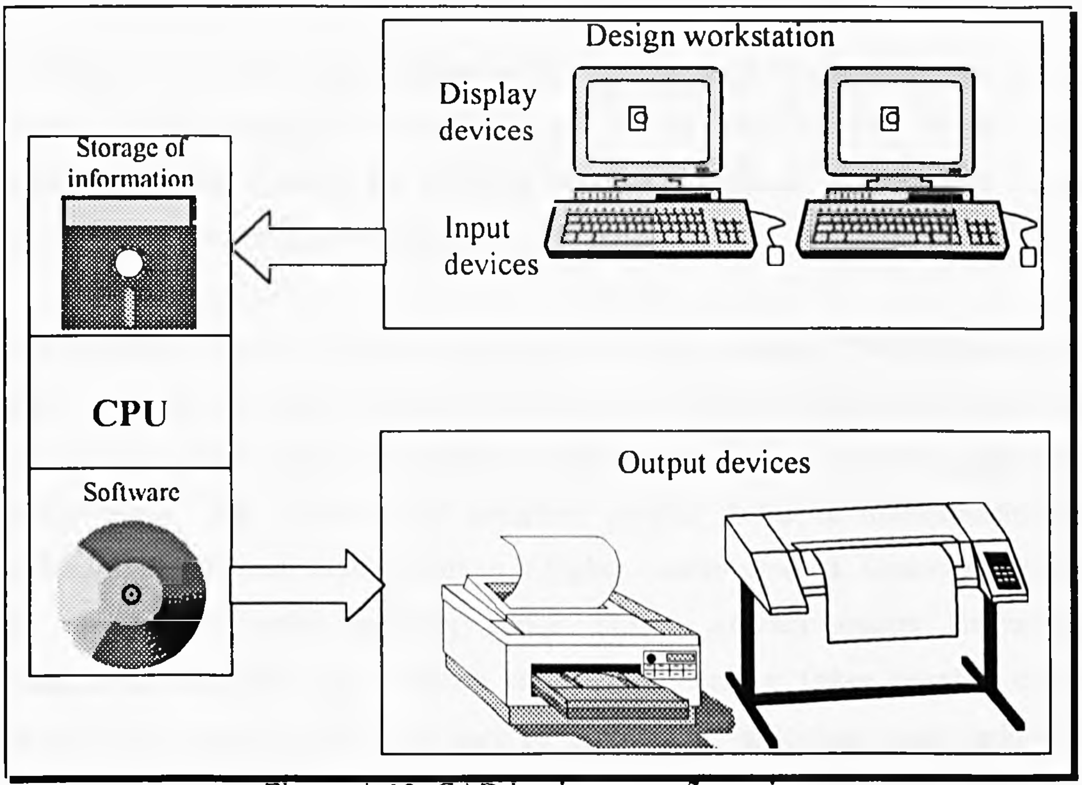


Figure A.12. CAD hardware configuration

In general, the CAD system can be divided into four main categories of hardware, as listed below:

1. A design workstation consists of:
    - Display devices such as graphics terminals.
    - Input devices such as joysticks and a mouse.
2. Output devices such a printer and a plotter.
3. CPU (Central Processing Unit) includes the central processing and CAD software.
4. Storage devices such as disk storage or magnetic tape storage.

A.1.5 The role of CAD in CIM

Two types of integration should be assigned. First the integration of CAD models and the integration of CAD with other components of CIM system. The design geometry developed by CAD users can be used by the analysis programme. This can be achieved when data is compatible for different system models. In the CIM environment, the design data can be used by the CAPP, CAM, CAQ, PP&C, etc. using databases and linking techniques that enable information to be transferred between system activities.

The design data can be used to prepare part programmes to run numerical control machines which actually produce the part (Kochan and Cowan 1986). These programmes can be prepared for different types of manufacturing processes such as machining centre, turning and milling, etc.

Marketing departments can also benefit from the CAD integration. They can use design data and drawings to prepare the advertisements, to create good views of product using CAD drawings, to develop part manuals and leaflets, and to presents proposal to customer faster. The 3D view of complete product helps in understanding the manufacturing processes which result in a higher quality product. Customers usually select the higher quality product; hence, this is another reason encouraging manufacturing enterprises to integrate their CAD systems. Other benefits can be achieved from integration of CAD such as a reduction in tooling costs, reduce the opportunities for misinterpretation, a shortening of lead times, better design provision, improved engineering productivity, improved accuracy of design, easier recognition of computer interfaces and designs that have more standardisation.

CAD can be considered as a child-strategy that is part of CIM strategy; hence, the interfaces between CAD and other components need to be carefully considered.

A.2 Computer Aided Process Planning

Process planning is an important activity that is used as a vital link between design and manufacturing functions, planning the strategy for the manufacture of a part (Pande and Walvekar 1990). Process planning can be defined as the activity which prepares the sequence instructions of manufacturing operations to be used for producing the part from initial stages (raw material form) to finished stages (ready product). The process plan instructions include information about the component route, geometrical features, manufacturing processes, machines, tools, fixtures and materials.

Zhang (1994) classified the approaches to process planning into two main categories: manual and computerised approaches. The manual approaches can be divided into two types: traditional and workbook; the computerised approaches can be divided into two main types: variant and generative approaches, as illustrated in Figure A.13.

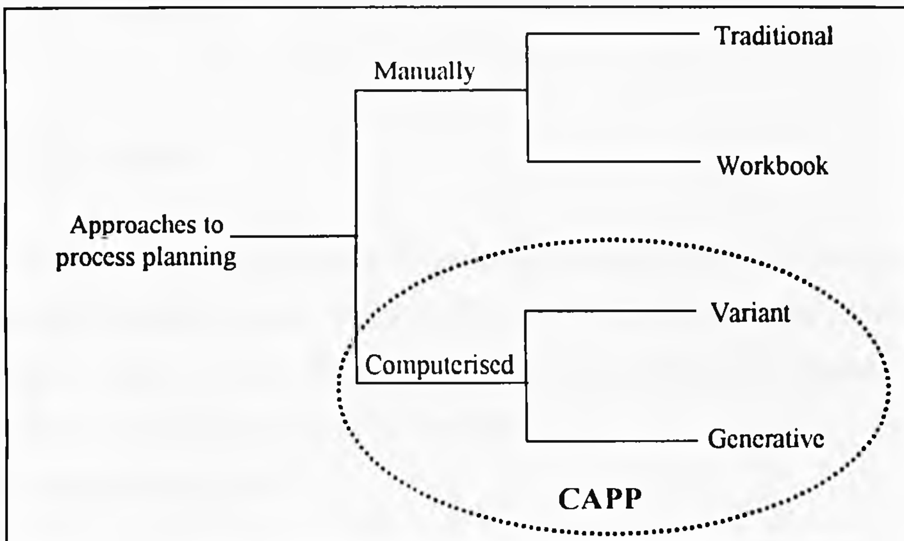


Figure A.13 Approaches to process planning.

In general, the development of process plans involves a number of activities including:

1. Analysis of part requirements
2. Selection of raw materials
3. Determine manufacturing operations and their routings
4. Selection of machine tools
5. Selection of tools, fixtures and inspection devices.
6. Determining machining conditions.

A.2.1 Manual Approach to Process Planning

Manual approach to process planning starts when the process planner receives the detailed design information and drawings. The planner uses the new design data to identify similarities with existing produced parts. If the similarities are recognised, the related process plan will be restored and used for the new part after carrying out the required modification if any. When the similarities are not recognised, the process planner will develop a new process plan for the new design.

The manual approach to process planning leads to inconsistency in the final part process plans. It is impossible that two process planners will produce identical process plans for the same part. Moreover, a process planner may develop a different process plan for the same product, because of the time period or increasing experiences. Another drawback of this approach is that it is time consuming; hence, it requires a process planner with sufficient knowledge and experience. CAPP approaches have been developed to overcome these limitations.

A.2.2 CAPP System

In 1965 Neible first suggested the idea of using the computer technology to assist in the determination of process plans (Alting and Zhang 1989). During the last 25 years, there has been much interest in the automation of process planning by means of CAPP systems (Groover 1987). In general CAPP should:

- interpret product design data
- select machining operations
- select machining tools
- select inspection devices
- select machining fixtures
- illustrate operation routings
- illustrate production tolerances
- describe cutting conditions
- calculate the overall times
- generate numerical control data
- specify raw material features
- provide list of costs
- estimate machining set-up times

A.2.2.1 The Variant CAPP Approach

The Variant approach to process planning was the first method used to computerise process planning steps. It is based upon Group Technology (GT) concept, in which the parts are grouped into families; hence, every part family has similar process plans. In the grouping methods, parts are classified based upon feature similarities among them. These features include the geometrical shape, size and manufacturing operation requirements. The computer is used as a tool to assist in identifying the similar process plans, retrieving the process plan from database and editing the selected plan in order to be suitable for a specific product design. The new process plans are manually developed and stored in CAPP database for every part family. Figure A.14 illustrates the CAPP variant approach.

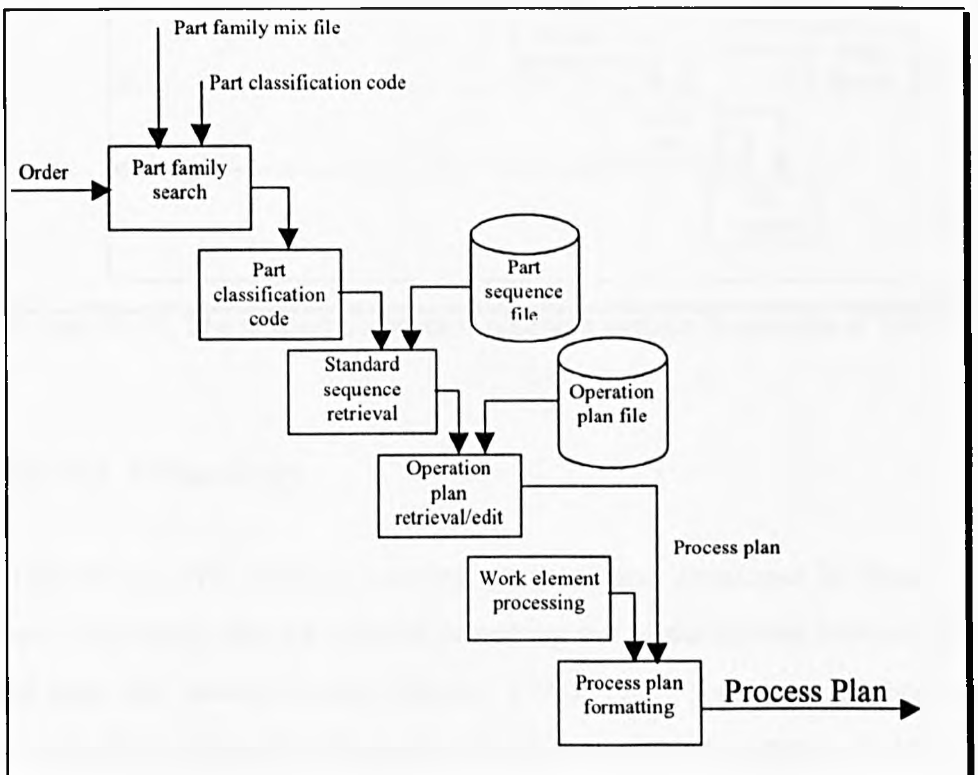


Figure A.14 CAPP system (the variant approach) (Rembold et al 1993).

A.2.2.2 The generative CAPP Approach

The generative approach to CAPP came into development in the late 1970s. In this approach, the process plans are not predefined and stored. Hence, the main function of generative approach is to generate a plan using information about a component design, machining, assembly, costs and other attributes and rules. The part data is used with specific databases, decision-making techniques and algorithms to produce plans. Figure A.15 shows the generative process planning system.

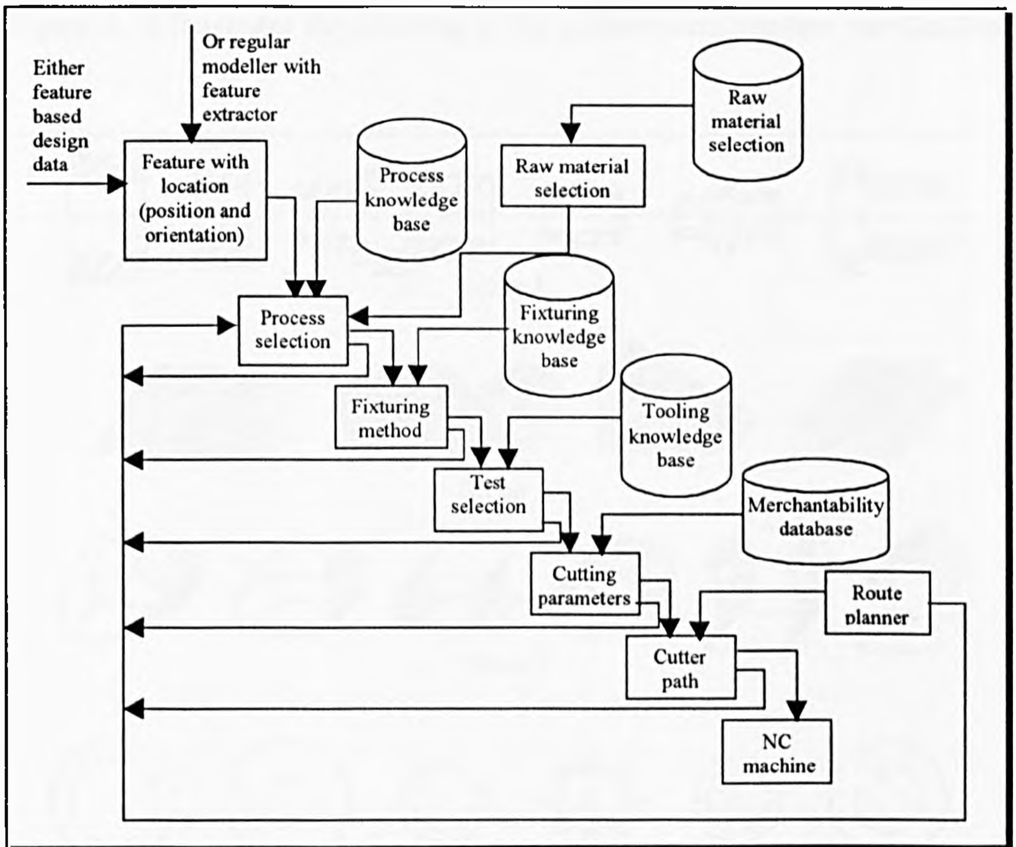


Figure A.15. The generative process planning system (Rembold et al. 1993).

A.2.3 Group Technology

Group Technology (GT) concept was first proposed and developed in Russia by S.P. Mitrofanov who recognised the benefits of making use of similarities between parts and classified them into several Groups (Hitomi 1996). The GT concept has been used in different manufacturing applications such as process planning, layout, inventories and cellular manufacturing systems.

GT can be defined as the philosophy of studying a large population of apparently different items and then classifying them into groups of items having the same or similar characteristics (Alting and Zhang 1989). Identifying the groups and their members is a large and difficult task. Hence, the person who carries out this job must collect the individual design and manufacturing attributes for every part and identify every attribute by a unique code number. The part design attributes may include material types, dimensions, basic external shape, basic internal shape, part function, surface finish and tolerances. The manufacturing attributes involve major processes, major operations, machine tools, batch size, cutting tools, annual production and fixture types. Figure A.16 illustrates the grouping of the various parts into few part families.

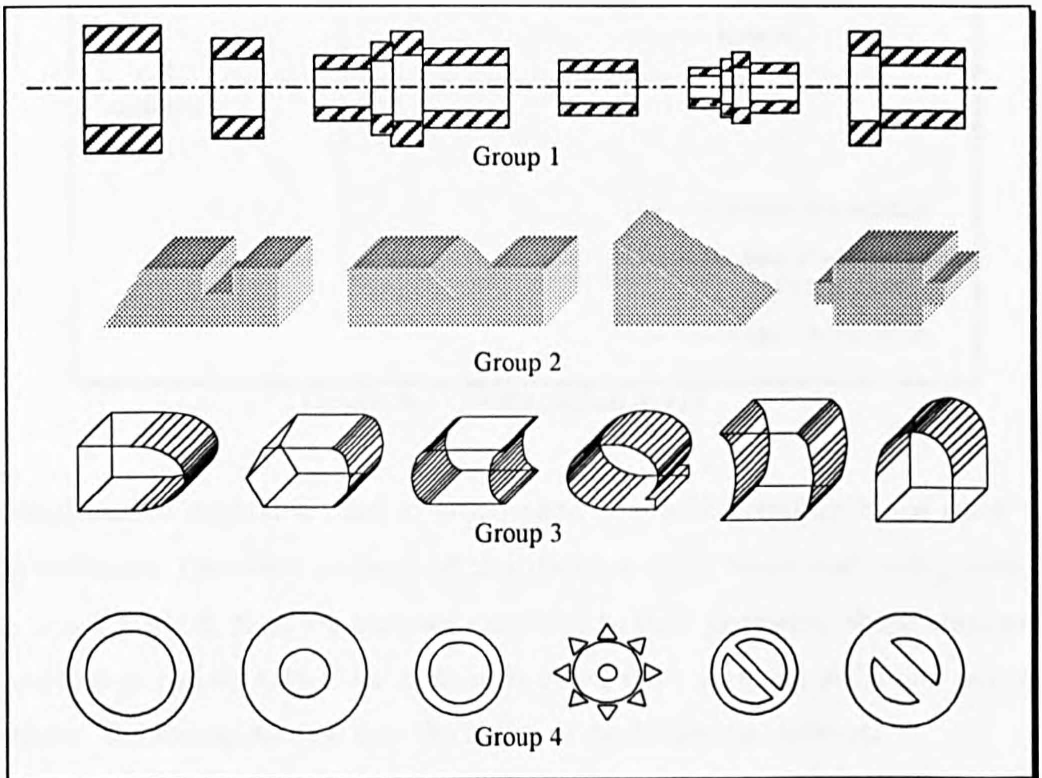


Figure A.16. Grouping parts into groups by similarity of shape.

A.2.3.1 GT Methods

In general, three methods are used for solving GT problem, as shown in Figure A.17:

1. Classification
2. Production flow analysis
3. Cluster analysis

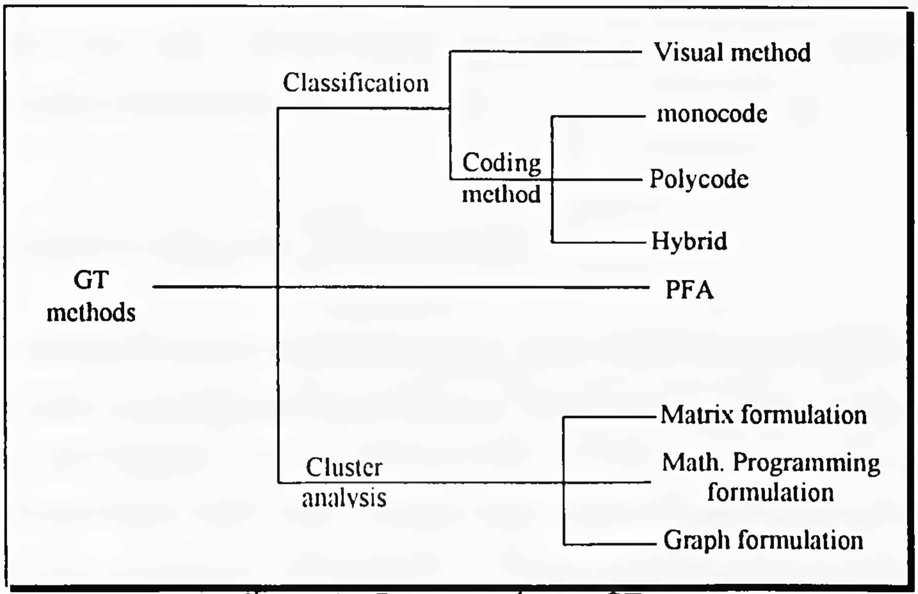


Figure A.17. Approaches to GT.

The classification method is used to group parts in specific families based upon their design attributes. Two main methods of classification exist: visual and coding methods. In the visual method, parts are grouped according to their geometric shape similarities, as illustrated in Figure A.16. This method is based upon personal skills and has many limitations. The coding method uses the basics of the following attributes:

1. Part design attributes
2. Part manufacturing attributes
3. Both design and manufacturing attributes.

The coding method is carried out based upon a sequence of numerical digits derived to define the part design and manufacturing attributes. Three basic methods of coding have been used: monocode, polycode and hybrid code. In the monocode, the meaning of each digit depends on the meaning of the digit which precedes it in the code. In the polycode, each part attribute is assigned to a fixed position in a code. Hence, the meaning of each digit in the code does not depend on any other digit. The hybrid is a combination of the monocode and polycode, thus it combines the advantages of both types.

In Production Flow Analysis (PFA) method, all parts which are produced by a given group of machines are grouped into one family.

A cluster analysis method is used to group objects into homogenous clusters (groups) based upon the object attributes (Kusiak 1991). In GT, the cluster analysis method can be used in three basic forms: matrix formulation, mathematical programming formulation and graph formulation.

A.2.4 The Role of GT and CAPP in CIM

GT can be considered as a management strategy to help in eliminating task duplications. In fact, GT affects all manufacturing functions of organisations. Many benefits can be achieved by the integration of this important concept. GT can assist the design function to reduce several factors such as the number of drawings, new parts design and drafting effort in new shop drawings, and present easy retrieval techniques that can be used for similar design activities. It can assist in layout planning in order to reduce production floor space required and reduce material handling movements. GT can assist in reducing set-up time and production time, improve machine loading, production cycles, part routing, material flow, schedule changes, reduce Work-In-Progress (WIP) and scrap generation. In addition, GT can also assist in purchasing and customer services. Kusiak (1991) and Kochan and Cowan (1985) reported an estimation of GT benefits in manufacturing:

- Reduced production lead time (20-88%)
- Reduced WIP (up to 88%)
- Reduced labour (15-25%)
- Reduced tooling (20-30%)
- Reduced rework and scrap materials (15-75%)
- Reduced order time delivery (13-136%)
- Improved human relations
- Reduced paper work
- Increased worker satisfaction.

It is clear that GT is an important element of CAD, CAPP, CAM and PP&C. An essential aspect of the integration of CIM elements is the integration of the information used by engineering and design, manufacturing and the organisational activities in an organisation. GT provides a means and techniques to structure, save and manpulate information about parts e.g. design and manufacturing attributes, process routings, and resource capacities.

On the other hand, CAPP represents a link between engineering and manufacturing activities particularly in CAD/CAM area. Hence, CIM cannot be achieved without CAPP. The CAPP system provides a means and procedure to close the gap between manufacturing system activities.

It has been discussed in the previous sections that the CAPP system depends upon GT concept. GT is an important concept in the CIM system because GT is the key for automation of the CAPP system which has been recognised as playing a key role in CIM.

A.3 Computer Aided Manufacturing

Computer Aided Manufacturing (CAM) refers to the use of computer technologies to direct and control manufacturing equipment. CAM can be defined as the use of computers to assist in planning, managing and controlling manufacturing operations from inventory control to programming of machine tools through either direct or indirect computer interface with the production resources (Beasnt and Lui 1986 and Groover 1987). The computer system in CAM can be used for monitoring or control. The difference between monitoring and control is that the monitoring system receives data from the manufacturing processes using a computer interface. The control system observes manufacturing processes data and performs the required control using different interfaces.

One of the main accelerators of CAD/CAM was the research in Numerical Control (NC) at MIT in the 1950s. NC came into existence before CAD. This evolution led to Computer Numerical Control (CNC) and Direct Numerical Control (DNC).

A.3.1 Numerical Control

The term 'NC ' is used to describe the control of the various functions of a machine using numeric data (Ssemakula 1996). In the early age of NC, machines were fed with information by means of the punched tape. An Electro-mechanical tape reader was used to load a machine tape into the controller. In general there are three basic components of an operational NC: programme of instruction, a machine control unit and machine tool, as illustrated in Figure A.18.

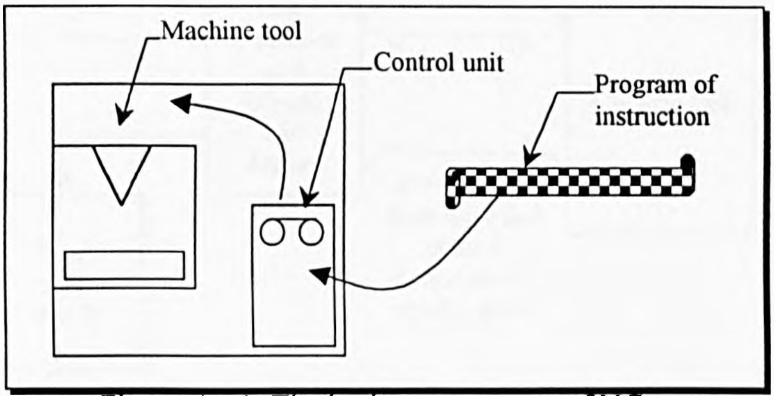


Figure A.18. The basic components of NC.

The program of instruction is a numerical or symbolic code that is detailed step-by-step to tell the machine tool what to do. The controller unit is the unit that reads the programme of instructions and converts it to real movement of a machine tool. Two basic types of control unit are used with NC machines: open-loop control and closed-loop control. The third basic component of NC system is the machine tool that performs the mechanical work and deals directly with the part being machined.

A.3.2 Computer Numerical Control (CNC)

CNC refers to a computer that is joined to the NC machine to make the machine versatile. Information can be stored in a memory bank (Rapello 1986). The programme is read from a storage medium such as the punched tape and retrieved to the memory of the CNC computer. Some CNC machines have a magnetic medium (tape or disk) for storing programs. This gives more flexibility for editing or saving CNC programs. Figure A.17 illustrates the general configuration of CNC.

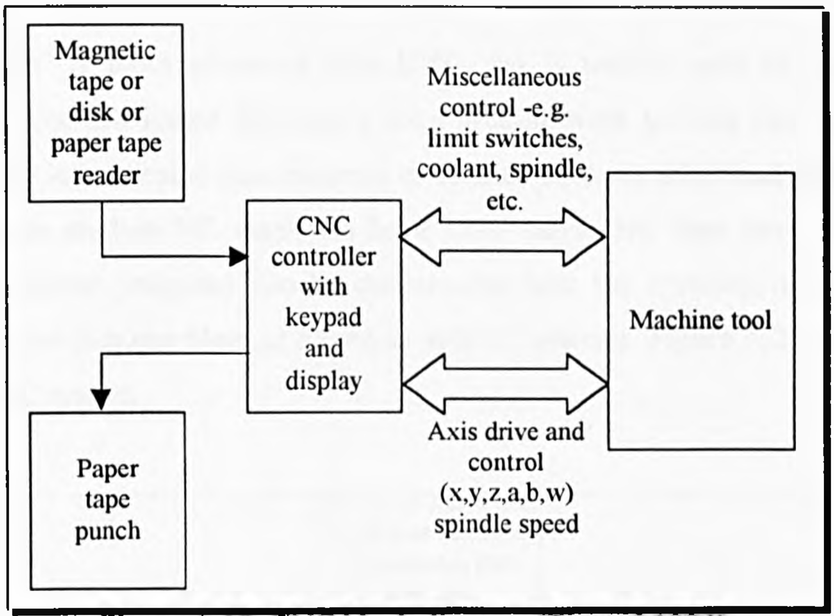


Figure A.19. CNC block diagram (Tizzard 1994).

A.3.3 Direct Numerical Control (DNC)

Direct Numerical control (DNC) can be defined as a set of NC machines that is connected to a main computer system to establish a direct interface between the DNC computer memory and the machine tools. The tape is not used in the DNC system; hence a central time-sharing computer is used. The basic structure of DNC system is illustrated in Figure A.20.

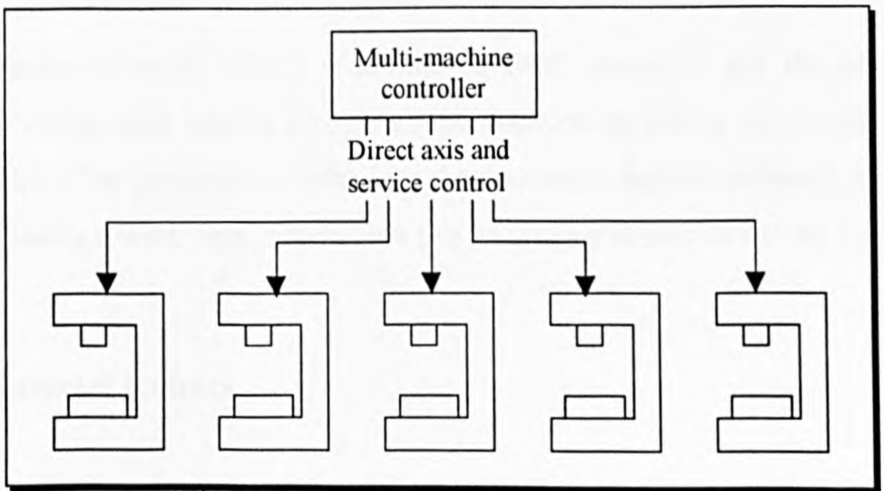


Figure A.20. DNC basic configuration

A.3.4 Distributed Numerical Control

Distributed NC is more advanced than DNC and is widely used in many current applications. The distributed NC uses a local area network but not like that in DNC. Ssemakula (1996) indicated that the main difference between DNC and distributed NC is that because modern NC machines have CNC capability, they have memory and therefore computer programs can be downloaded into the memory of a CNC local computer, rather than one block at a time as in DNC systems. Figure A.21 illustrates the distributed NC system.

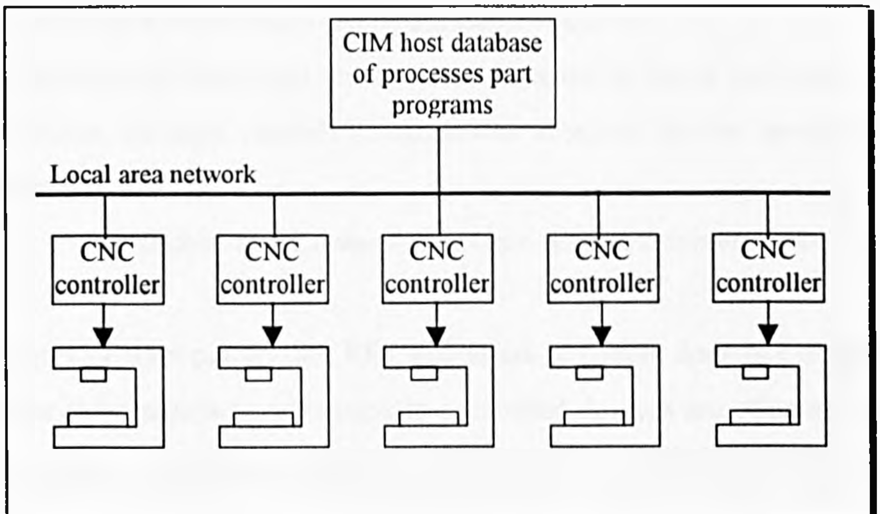


Figure A.21. Configuration of distributed NC system.

A.3.5 Voice Numerical Control (VNC)

Voice Numerical Control (VNC) is similar to DNC machines but the programmer conveys the information needed to operate the machine by means of computer system (Rapello 1986). The programmer talks into the computer, and the memory receives the information using a wire. This information can be taken and used to run the machines.

A.3.6 Industrial Robots

Many types of industrial robots exist, and the changes of applications that use them are also growing rapidly. Hence, the use of industrial robots has a strong impact on the efficiency and flexibility of manufacturing systems (Lee et al. 1992). Thus, an industrial robot should be defined. The international Standards Organisation (ISO) defined an industrial robot as (Nof 1992):

"A machine formed by a mechanism including several degrees of freedom, often having the appearance of one or more arms ending in a wrist capable of holding a tool or a work-piece or an inspection device. In particular, its control unit must use a memorising device and sometimes it can use sensing or a adaptation appliance taking into account environment and circumstances. These multipurpose machines are generally designed to carry out a repetitive function and can be adapted to other functions"

The Robotic Industries Association (RIA) defined a robot as:
"A programmable, multifunctional manipulator designed to move materials, parts, tools or special devices through variable programmed motions for the performance of a variety of tasks"

(Kochan and Cowan 1985, Groover and Zimmers 1984, Singh 1996)

Nof and Rajan (1994) reported that RIA definition of robots does not directly include pick-and-place arms as robots and remotely controlled devices are often also refer to as robots. They also concluded that both definitions mentioned above stress the multifunctional and programmable capabilities and, therefore exclude special-purpose "hard automation" tools and equipment typically found in high volume production.

A.3.6.1 Industrial Robot Classification

Industrial robots can be classified based upon several factors such as method of teaching, physical configurations, their industrial applications or control systems. In Japan, the JIRA (Japanese Industrial Robot Association) classifies robots by the method of input information and method of teaching (Nof 1992, Nof and Rajan 1994):

1. Manual manipulator. This type is directly activated by the operator.
2. Fixed-sequence robot. This type is once programmed for a given sequence of operations and is not easily changed.
3. Variable-sequence robot. This robot can be programmed for a given sequence of operations and can easily be changed or reprogrammed.

4. Playback robot. This robot memorises work sequences taught by a human being who physically leads the device through the intended work pattern. The robot can then create this sequence repetitively from memory.
5. Numerically controlled robot. This type operates and is controlled by digital data, as in the form of punched tape or card. It operates like a NC machine.
6. Intelligent robot. This robot uses sensory perception to evaluate its environment and make decisions and proceeds to operate accordingly.

Singh (1996) classified robots based upon their physical configurations and control systems. Using the physical configurations, robots can be classified into four types:

1. Cartesian configuration. The body moves horizontally and vertically and the arm horizontally, as illustrated in Figure A.22a.
2. Cylindrical configuration. The body is a column rotates on a vertical axis and moves the arm up ward and down ward, as shown in Figure A.22b. The space in which this robot works is cylindrical in shape.
3. Polar configuration. The body with vertical motion pivots either horizontally (rotary) or vertically, or a combination of the two, as illustrated in Figure A.22c. The space in which this robot works is spherical in shape.
4. Jointed-arm configuration. The body, the column and the arm flexibly rotate about their respective joints; hence this type is considered as a combination of cylindrical and articulated configurations, as shown in Figure A.22d.

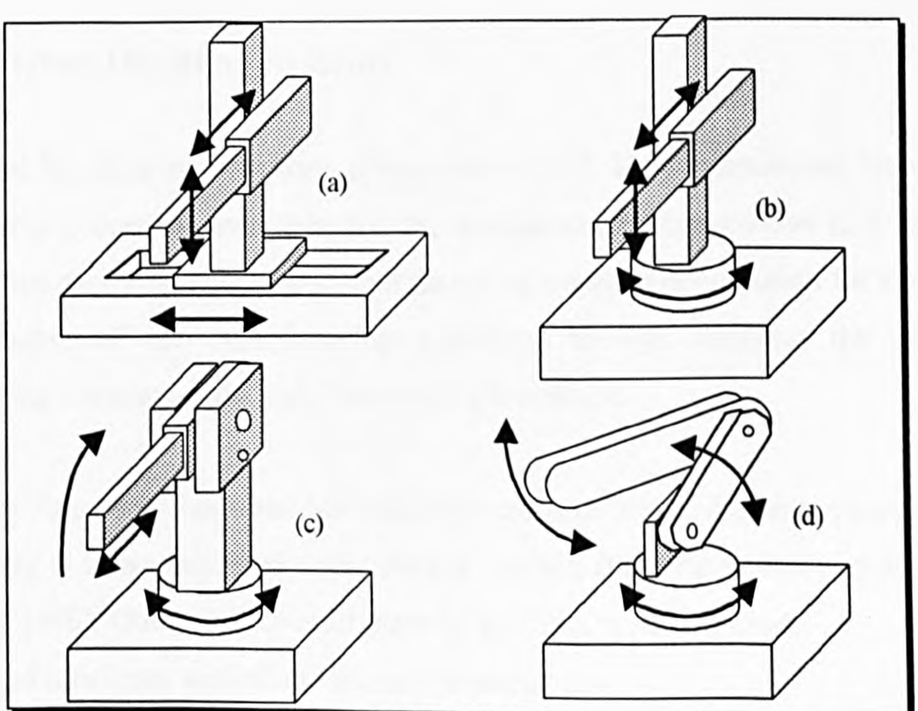


Figure A.22. Robot types.

Using the control systems of industrial robots, robots have been classified into three categories (Singh 1996):

1. Point-to-point control robot. This type is controlled to move from one point to another point. It does not control the path to get from one location to another.
2. Continuous-patch control robot. This type is capable of moving along a controlled path. All points along the controlled path are stored in the memory.
3. Controlled-path robot. This type moves at a high degree of accuracy. The path can be generated at different geometry such as straight lines, circles and curves using the control equipment.

A.3.6.2 Industrial Robot in CIM

The application of robots in CIM has been increasing steadily over the years; hence, robots are essential elements in CIM systems. The manufacturing applications of robots includes many operations such as pick-and-place, machine-tending, palletising operations, painting, welding, finishing, inspection, assembly operations and other applications. Moir (1989) reported that when fully engineered industrial assembly robots for the fourth generation - as exemplified by the Cambridge prototype- will become available, probably by the late 1990s, and are likely to find a wide range applications in manufacturing.

A.3.7 Material Handling Systems

The material handling systems have a key role in CIM. They interconnect the various manufacturing processes, materials, robots, storage and other activities to a function unit. The manufacturing handling system makes an essential contribution for achieving high utilisation of the manufacturing resources, through handling the different manufacturing elements at the right time in a right position.

The primary function of material handling systems is to ensure that the material in the right quantity is safely delivered to the desired position at the right time with minimum cost (Singh 1996). Other objectives of material handling systems include:

1. Increased efficiency and effectiveness of material flow
2. Increased productivity in manufacturing

3. Improved safety and work conditions
4. Reduced costs
5. Reduced space and equipment use
6. Improved integration
7. Improved flexibility
8. Improved utilisation

A.3.7.1 Principles of Material Handling Systems

Gelders and Pintelon (1994) reported the principle equation of material handling system design as:

"Materials + Moves + Methods = Best System"

To understand the material handling attitudes several quotations should be answered such as: why, what, where, when, how, who and which? Aiming to specify the material handling to be moved, quantities, units, source, destination, speed, frequencies, etc. The material Handling Institute Inc., consolidated the experience of many contributors into a list of twenty material handling principals (Gelders and Pintelon 1994, Groover 1987), as shown in table A.1.

1.	Planning. Study the problem thoroughly to identify potential solutions and constraints and to establish clear objectives.
2.	Flow. Integrate data flow which physical material flow in handling and storage.
3.	Simplification. Try to simplify material handling by eliminating, reducing, or combining unnecessary movement and equipment.
4.	Gravity. Use gravity to move material wherever possible, while respecting limitations concerning safety and damage.
5.	Standardisation. Standardise handling methods and equipment wherever possible.
6.	Flexibility. Use methods and equipment that can perform a variety of tasks.
7.	Unit load. Handle product in as large a unit load as possible.
8.	Maintenance. Plan maintenance carefully to ensure high system reliability and availability.
9.	Obsolescence. Make a long-range plan, taking into account equipment life cycle costs and equipment replacement.
10.	Performance. Determine the efficiency, effectiveness, and cost of the material handling alternatives.
11.	Safety. Provide safe material handling equipment and methods.
12.	Ecology. Use equipment and procedures that have no negative impact on the environment.
13.	Ergonomics. Take human capabilities and limitations into account while designing a material handling system.
14.	Computerisation. Consider computerisation wherever viable for improved material and information control.
15.	Utilisation. Try to obtain a good use of the installed capacity.
16.	Automation. Consider automation of the handling process to increase efficiency and economy.
17.	Operation. Include operation costs (energy) in the comparison of material handling alternatives.
18.	Integration. Integrate as much as handling and storage activities into one co-ordinated system, covering receiving, inspection, storage, transportation, production, packaging, warehousing, and shipping.
19.	Layout. Keep in mind that layout and material handling are closely linked and that an interactive procedure is often needed to obtain their best co-ordination.
20.	Space use. Choose the material handling equipment to that effective use is made of all (cubic) space.

Table A.1. Basic material handling principles (Slightly modified by Gelders and Pintelon 1994).

A.3.7.2 Material Handling Equipment

Material handling equipment can be classified into five categorise (Singh 1996):

1. Industrial trucks
2. Conveyors
3. Monorails, hoists and cranes
4. Automated guided vehicle system
5. Automated storage/retrieval system

A.3.7.2.1 Industrial Trucks

Industrial trucks are used to move loads intermittently over path with suitable surfaces. They can be used to carry out a number of functions such as transferring, lifting, loading and unloading. This type of material handling equipment is characterised by its high flexibility and is not restricted to a fixed path. Its main drawback is that it does not allow integration with other manufacturing functions.

A.3.7.2.2 Conveyors

Conveyors are also used to move materials (bulk or units) continuously over a fixed path. This type of handling system has high capacity, adjustable speed and versatility (floor or overhead types), can be integrated with the other manufacturing functions. Its main drawback is that it has a fixed path and its breakdowns may impact on the whole related manufacturing operations. There are different types of conveyors such as belt conveyors, slat conveyors, roller conveyors, wheel conveyors, gravity conveyors, trolley conveyors and two line conveyors.

A.3.7.2.3 Monorails, Hoists and Cranes

Monorails, hoists and cranes are over head material handling equipment used to move loads between points within an area. They can be supported and guided by rails. The main advantage of this type is that it does not consume valuable floor space. They can be controlled by electronic devices and can be integrated into other manufacturing functions.

A.3.7.2.4 Automated Guided Vehicle

Automated Guided Vehicle (AGV) is a material-handling vehicle with automatic guidance used to move large and palletised parts. It moves along a fixed path prescribed by a guide wire embedded in the floor. This type of material handling can be integrated and controlled using computer systems in the CIM environment. In the integrated system, AGV can be controlled to transport material or parts from Storage/Retrieval systems and convey the material or parts to various production workstations. A computer control system usually directs the AGV from one point in the production system to another. The links that compose the AGV path are usually divided into zones that can be occupied by one vehicle at a time. There are several types of AGVs available such as AGVs towing vehicle, AGVs unit load, AGVs pallet trucks, AGVs light-load and AGVs assembly-line vehicles to meet different manufacturing applications

A.3.7.2.5 Automated Storage/Retrieval System

Automated Storage/Retrieval System (AS/RS) is a rack storage system with rail running vehicle serving the rack structure for amount loading and unloading (Ingersoll Engineers 1985). The AS/RS consists of a series of storage aisle. These aisles have storage racks that are served by storage/retrieval machines. The main operations of AS/RS are:

1. Automatic retrieval of a part from the storage rack
2. Transportation of the part to a specific location (manufacturing process, packing, assembly, etc.).
3. Automatic storage of a part in a specific location.

Figure A.23 illustrates a brief explanations for AS/RS.

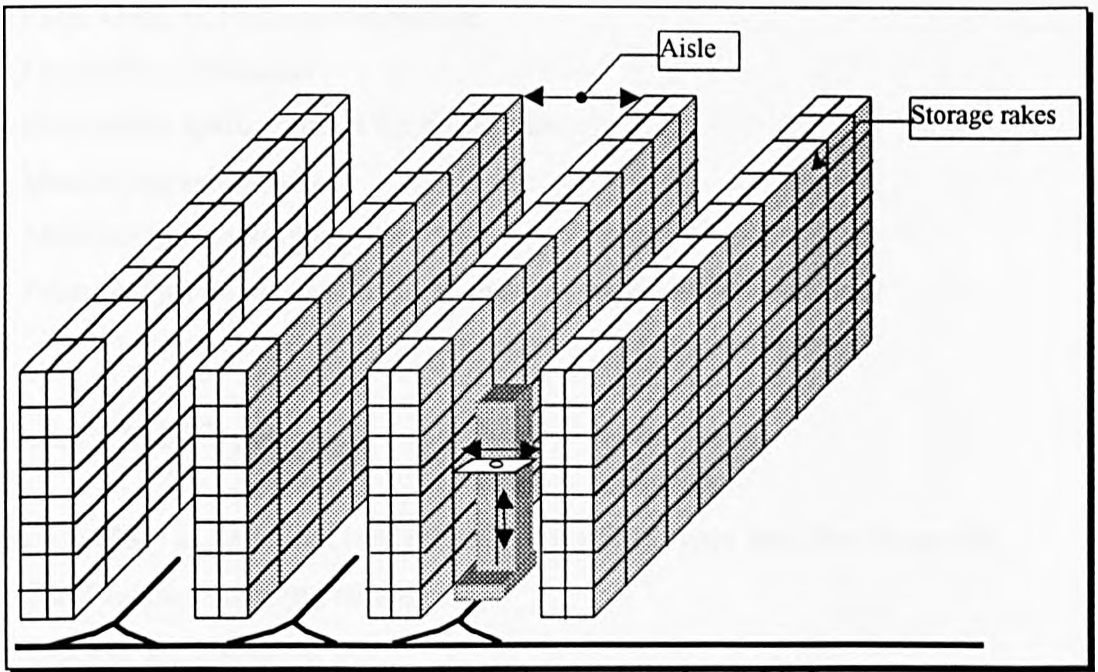


Figure A.23. General structure of an AS/RS.

A.3.8 Flexible Manufacturing Systems

Flexible Manufacturing System (FMS) is the most advanced manufacturing equipment of CAM system and plays an important role in the construction of the CIM system. The flexibility is the cornerstone and one of the key concepts used in the design of integrated manufacturing systems. But there is still some uncertainty relating to the concept of flexibility and which manufacturing system is termed "flexible". In fact, there are several types of flexibility that have been reported such as those considered in chapter-2. Talavage and Hannam (1994) reported that any definition of FMS should include its capability to produce different parts simultaneously and for the changeover between parts to be automatic or without significant human intervention. The word "flexible" in FMS can be defined as easily and quick adaptable or changeable. Hence, the FMS is capable of manufacturing different parts without significant delay in changing between them. In general, it is a manufacturing system for medium volume, medium variety parts production.

In FMS, the material handling equipment (robots, AS/RS or AGV) is used to complement the DNC machines and to move materials/parts from one workstation to another. All FMS elements may be connected to a common computer system to provide the instructions for job sequencing, monitoring and controlling the different elements of the FMS. Goals of FMS system include:

1. Maximising utilisation
2. Minimising space required for production
3. Minimising set-up time
4. Minimising inventory levels
5. Providing proper integration of all areas in the manufacturing floor

A.3.8.1 FMS Types

Dupont (1981) and Kusiak (1985) classified the FMS types into five categories:

1. Flexible manufacturing module.
2. Flexible manufacturing cell.
3. Flexible manufacturing group.
4. Flexible production system.
5. Flexible manufacturing line.

This classification was designed based upon the most applicable range in terms of the number of different part type per system and the variety of parts, as shown in Figure A.24.

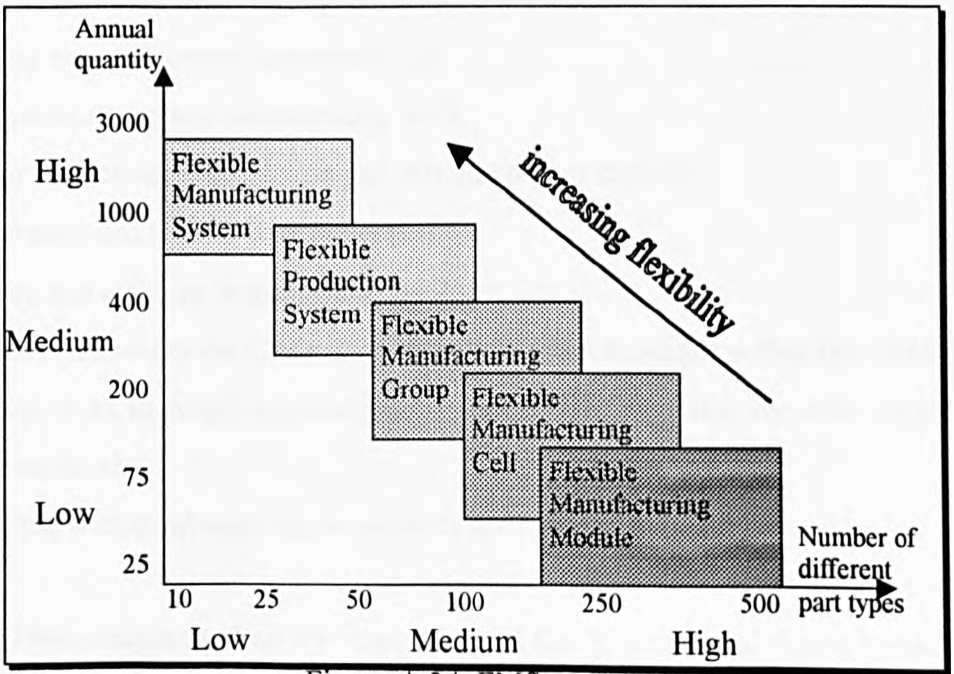


Figure A.24. FMS types.

A.4 Computer Aided Quality Control

Computer Aided Quality Control (CAQ) is one of the basic components of the CIM system. CAQ, Computer Aided Inspection (CAI) and Computer Aided Testing (CAT), all have similar meaning and objectives. The CAQ is an application computer system that tests parts and product design specifications through interaction with other manufacturing functions such as CAD/CAM functions. Blasing (1993) classified the CAQ objectives into two main categories: strategic objectives and operational objectives.

The strategic objectives of CAQ are:

- Maintaining and promoting the capability for quality.
- Reducing the risk of poor quality.
- Preventive action to avert product liability.
- Transparency and topicality of quality activity.
- Faster reaction to quality problems.
- Extending market share by improving information.

The operational objectives of CAQ are:

- Rationalisation of inspection and testing procedures.
- Reducing idle time and throughput time at the inspection points.
- Cutting expenditure on documentation.
- Less re-work, rejects and warranty work.
- Butter utilisation of inspection and test equipment capacity.
- Systematic and regular data collection.
- Timely and efficient inspection technology.
- Directly links from the CAQ systems to individual locations within the company.
- On-line links between machines and in-house CAQ systems for error location and error removal.
- Working with databases whose contents are relevant to quality.

Hitomi (1996) emphasised on the importance of CAQ, because of recent trends in high production speed, quick detection of defective products, high labour cost, on-line process control and so on. Groover and Zimmers (1984) also indicated the importance of CAQ in carrying out quality inspection at 100%. By using CAQ, inspection and testing will typically be performed on a 100% basis rather than by the sampling methods that are used in the traditional quality control systems. In addition to these, CAQ can be integrated into other manufacturing functions such as production processes in CAM system. This will help to reduce manufacturing time.

The CAQ system can be considered as the cornerstone of the CIM system, that the CIM will be capable of realising its strength for the future manufacturing strategy. Hence it is important to add the CAQ system to the CIM framework. This enables the CAQ system to use other manufacturing databases to perform its function properly. Another way in which a common integration is helpful to CAQ is when the CAD system changes the product specifications; this will make changes for CAQ procedures. Integrated CAQ also leads to integration of inspection planning into job planning, optimisation of machine usage and warehouse in accordance with quality aspects.

A.5 Production Planning and Control

Every one is becoming aware about the manufacturing problems. The serious search for solutions of these organisation's problems causing great concern to many researchers and manufacturing engineers. Production Planning and Control (PP&C) represents the organisational function of CIM system. This field can be considered as a classic field of computer applications. The PP&C integration with the manufacturing technical aspects represents the framework of CIM strategy. Hence PP&C is expected to be the most critical component of the CIM.

The PP&C task involves managing and organising the manufacturing systems. The objective PP&C is to ensure that the desired products are produced at the right time, right quantity, right quality and at minimum cost (Singh 1996). The term 'PP&C' is used now to describe the use of computer systems for the planning, control and monitoring of manufacturing systems, from the processing of proposals to the dispatch of products, taking into account quality, delivery data and capacities (Maisch 1993).

There are different views concerning, the nature of PP&C tasks; some perspectives have restricted the PP&C to inventory control, Bill of Material (BOM), forecasting, order processing or shop floor control. The development of these aspects and the introduction of the intelligent systems would be helpful, but the most beneficial factor would be achieved when those PP&C different perspectives are integrated and supported by powerful computer systems.

The current trend concerning PP&C is to build large scale computer-based systems which are typically based upon well-known approaches such as Material Requirement Planning (MRP), Just In Time (JIT), Kanban and Optimised Production Technology (OPT) (Sarin and Das 1994).

The integrated PP&C can achieve many benefits such as:

1. Reduced inventory.
2. Reduced costs.
3. Increased outputs.
4. Reduced overtime.
5. Reduced lead times.
6. Improved customer relations.
7. Improved communications.

Figure A.25 illustrates a general definition of the PP&C system.

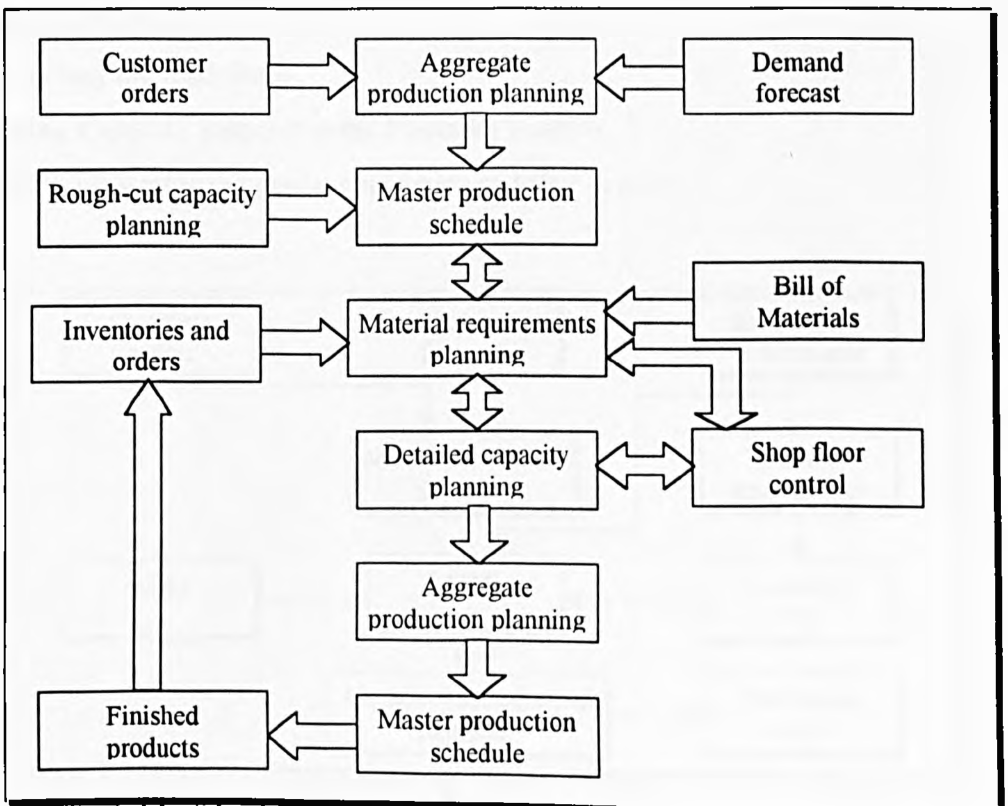


Figure A.25. A basic framework for manufacturing planning and control

A.5.1 Material Requirements Planning

Bedworth and Bailey (1987) defined Material Requirements Planning (MRP) as "a computer-based system for managing inventory and production schedules. This approach to material management applies to large job-shop situations in which many products are manufactured in periodic lots in several process steps. It does not apply to continuous-flow-type manufacturing systems".

The MRP system is designed to make purchased and in house manufacturing components available for dispatch before they are needed by the next stage of production (Sarin and Das 1994). Procedures of MRP system start by establishing a master production schedule that involves the required quantities of product demands and their related planning periods. Then, by using BOM of a specific product and inventory file, the production information for dispatching multiple jobs is generated on the hierarchical mutable-stage manufacturing system by taking into account commonness and possible substitution of many parts needed to assemble those mutable products. Thus, MRP schedules and controls the general flow of material and components from the first stage of manufacturing until the finished product stage according to time scheduling. These procedures of MRP system can be summarised as:

1. Calculating the requirements for products/components.
2. Making up the lot.
3. Subtracting the lead times.
4. Running Capacity Requirements Planning System.

Figure A.26 illustrates the general structure of MRP system.

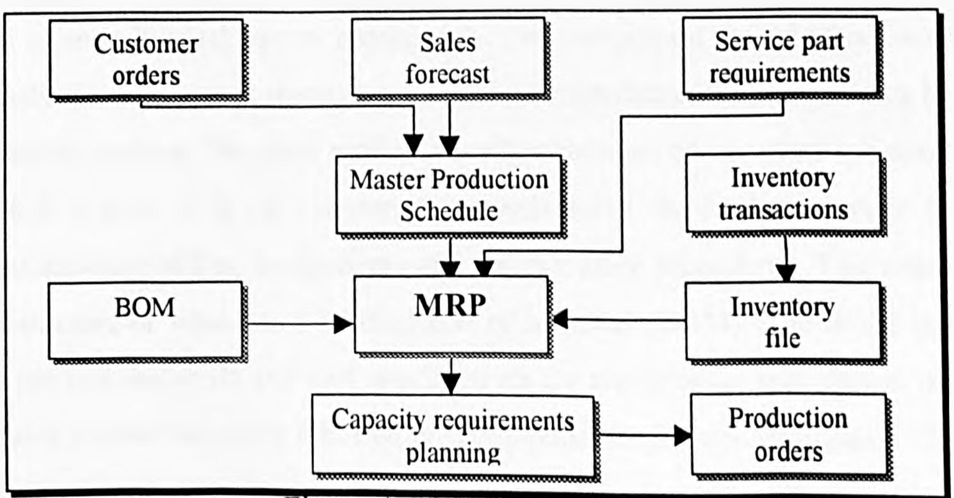


Figure A.26. MRP structure.

A.5.2 Master Production Schedule

A Master Production Schedule (MPS) is a list of what end items are to be produced, how many of each item is to be produced and when the items are to be ready for dispatching, as illustrated in Figure A.27

Time	1	2	3	4	5	6	7	8
Item 1		20			50			21
Item 2		30				23		40
Item 3				40				
Item 4			20				15	
Item 5						15		60
Item 6			32		50			80

Figure A.27. MPS representation.

The MPS drives the whole operations in terms of what is to be manufactured, what is to be assembled and what is to be bought. During the creation of MPS, several elements should be taken into account such as: known orders, forecast demand, plant demand, R&D demand, safety stock requirements, sparse demand, inventory levels and key capacity constraints. The MPS can be expressed in terms of make-to-stock, assemble-to-stock or make-to-order production systems.

A.5.3 Bill of Material

Any product may be made for one or more sub-assembles and components or may be produced as an individual part or component. This component is made from some form of raw material which goes through a number of manufacturing processes to be at the end a saleable product. The final product usually combines one or more sub-assemblies. In the MRP system, it is very important to understand the final product in terms of number of sub-assemblies, components and the assembly procedures. This refers to the product structure or what can be called Bill of Material (BOM). The BOM is used to compute the raw materials and part requirements for end product specified in the MSP. It provides a product structure that involves components and sub-assemblies. The BOM should be generated for each end item and it can be presented in different forms (tree, incident matrix, graph and identified form), as depicted in Figure A.28.

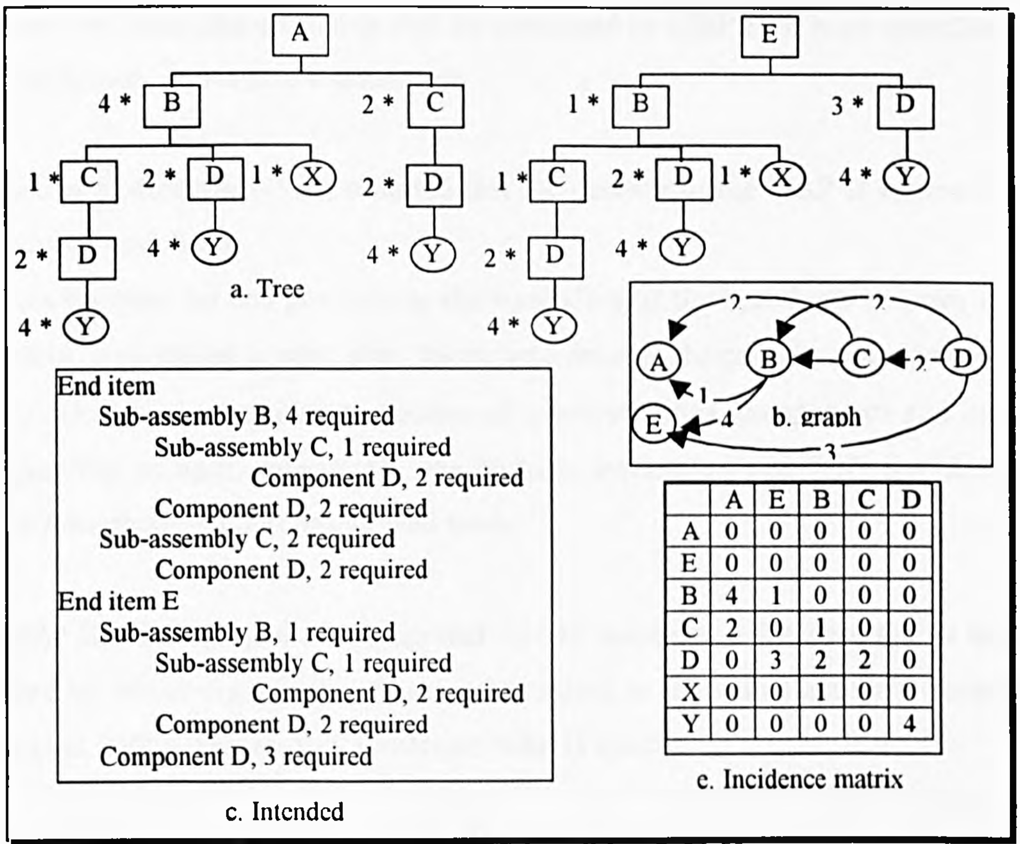


Figure A.28. Different methods for describing BOM (Grunbbstrom 1995).

A.5.4 Capacity Requirements Planning

Capacity Requirements Planning (CRP) is used to examine the possibility of assembling products and machining parts in relation to the capacity of production facilities. The CRP depends on the calculated required due dates for each batch order without taking into account other constraints of manufacturing resources. The main problem of CRP is that it lacks methods that can be used to optimise production throughout the factory. It is too difficult to calculate exactly when every job will be finished and what the load will be on each production stage. Therefore, the integration of system activities is very important.

A.5.5 Material Resources Planning

Material Resources Planning (MRP II) includes planning and control of production and inventory. The concept of MRP was extended to include other manufacturing areas. This extended concept becomes MRP II. In general, the MRP II is represented by the link-up between a closed-loop MRP system and financial systems of the organisation. There are two basic characteristics that are possessed by MRP II: it is an operational and financial system, and is also a simulation.

Bertrand and Muntslag (1993) reported that the essence of the MRP II approach could be described as follows:

- To determine the end products or the equivalent of final products in terms of what should be produced at what time, taking into account the capacity consequences.
- To calculate the required production of sub-assemblies, components and materials depending on up-to-date BOMs, the on-hand inventories and WIP, the batch sizes and manufacturing purchasing lead times.

The MRP II is based upon one integrated system involving a database that is accessed and used by whole organisation function according to individual activity requirements (Slack et al. 1995). Figure A.29 illustrates MRP II system.

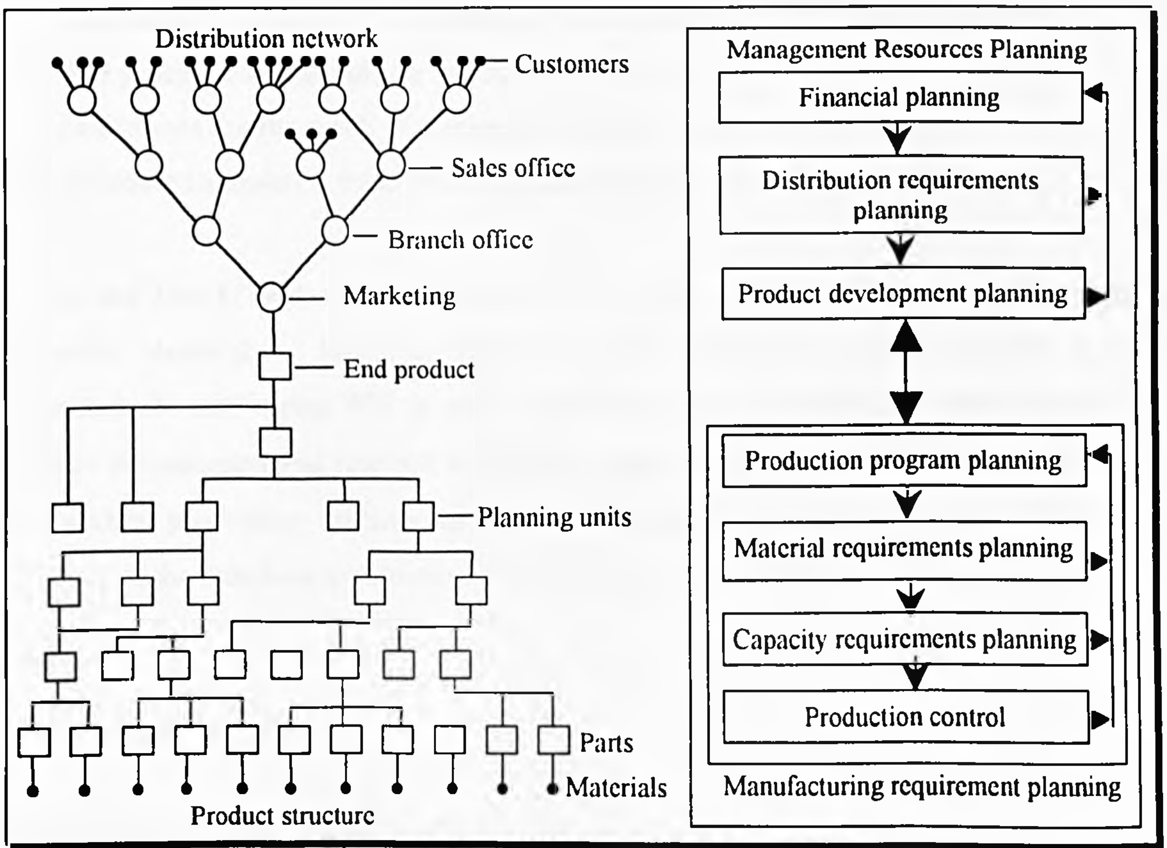


Figure A.29. MRP II model (Scheer 1994)

A.5.6 Optimised Production Technology

The Optimised Production Technology (OPT) system is computer-based techniques and tools that help to schedule production systems to the pace dictated by the most heavily loaded resources; i.e. bottlenecks (Slack et al. 1995). The OPT thus, efficiently focuses on the identification of bottlenecks and generates an optimal reschedule to reduce lead time (Nicholson 1991). The foundation of OPT system can be summarised in ten rules that are adopted by OPT users (Slack et al. 1995):

1. Balance flow, not capacity.
2. The level of utilisation of a non-bottleneck is specified by other constraints in the system, not by its own capacity.
3. Utilisation and activation of a resource are not the same.
4. An hour lost at a bottleneck is an hour lost forever out of the entire system.
5. An hour saved at a bottleneck is a mirage.
6. Bottlenecks govern both production rate and inventory in the manufacturing systems.
7. The transfer batch may not, and many times should not, equal the process batch.
8. The process batch should be variable, not fixed.
9. Lead times are the result of a schedule and cannot be predetermined.
10. Scheduled should be established by looking at all constraints simultaneously.

Sarin and Das (1994) suggested that it is not clear whether OPT covers long-range capacity planning or inventory control; hence, they have considered OPT as an approach for minimising WIP at shop level. However, the general philosophy of OPT is widely documented and relevant to CIM development. OPT takes a proactive approach to tackling production bottlenecks; it uses its algorithms to identify these bottlenecks and adjust the schedule to minimise their impact on production.

A.5.7 Just In Time

Multiple production process can be divided into two types: push and pull systems. The difference between these two systems is that in the pull system, the succeeding production stage demands and withdraws WIP units from the preceding production stage only. On the other hand, the push system is controlled through inventory level sets at every stage in the production system based upon the rate of demand. Just In Time (JIT) system is referred to as a pull system approach.

In 1987, Voss (addressed by Slack et al. 1995) reported a fuller definition to JIT: "Just-in-time (JIT) is disciplined approach to improving overall productivity and eliminating waste. It provides for cost-effective production and delivery of only the necessary quantity of parts at the right quality, at the right time and place, while using a minimum amount of facilities, equipment, materials and human resources. JIT is dependent on the balance between the supplier's flexibility and the user's flexibility. It accomplished through the application of elements which require total employee involvement and team-work. A key philosophy of JIT is simplification."

In general, JIT is simply used to operate a manufacturing system efficiency to deliver only the required quantities of products at the right time and to the right place, using the minimum manufacturing resources. This system has been applied in Japan with flexible automation. It represents a highly integration production, sales and distribution system, production continuous and efficient flow of material, components and products through the overall system (Nicholson 1991). In addition to these, JIT can lead to reduced inventories, improved quality, reduced lead times, increased utilisation, increased productivity, minimised changes of WIP and synchronised production processes. Figure A.30 illustrates the traditional and JIT flow between manufacturing stages.

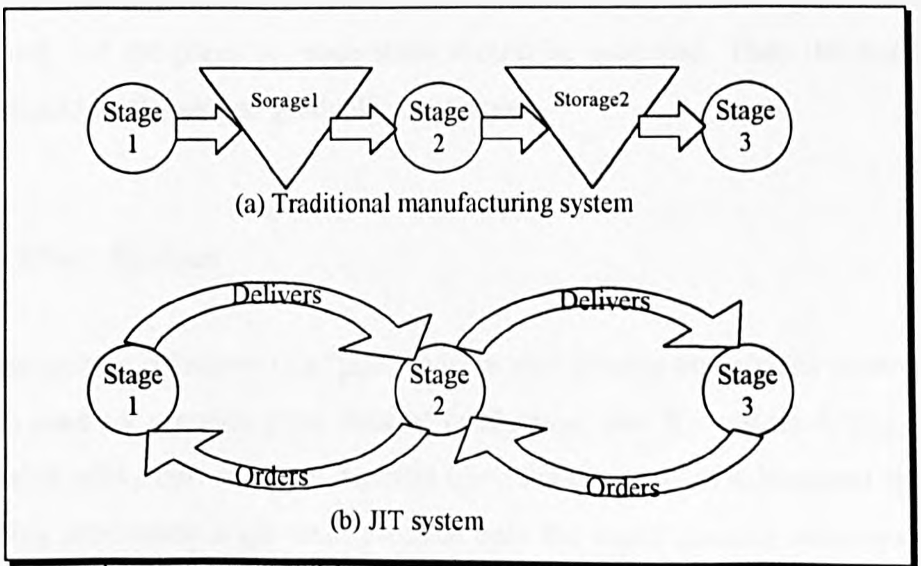


Figure A.30. Traditional manufacturing system and JIT.

A.5.7.1 JIT Factors

Bowman (1992) reported eight factors that JIT is based on:

1. Design product for economical production. The product manufacturablity should be considered in the design stage taking into account the existing manufacturing facilities and process constraints.
2. Change plant layout to facilitate "flow" production. A long portion of manufacturing lead times is consumed in non-production. Hence, the plant layout should be changed in order to shorten material moves.
3. Institute worker inventory programs (e.g., quality circles). The waste should be eliminated. This can be done by running programs related to manufacturing knowledge to provide the employees with the most important fundamentals of manufacturing systems using suitable methods.
4. Improve data accuracy. To take a right decision you should have the right data. Hence, the accuracy of data should be considered and measured using specific tools.
5. Reduce paper work. Data needs to be updated from time to time. To achieve this, hard copy reports are out of date and invalid for most decision-making.
6. Reduce scrap. The cost of scrap affects on several aspects of manufacturing systems. It is not only cost of scarp only, but also costs of rescheduling, reordering, handling, lead time loss, etc.
7. Reduce inventories. Levels of inventories should be reduced to the minimum levels and the balance between orders manufacturing and inventory should be achieved.
8. Strive for continuous improvement in all areas. Organisation goals should be identified, and the plans to reach them should be proposed. Then the organisation goals should be developed gradually over time.

A.5.8 Kanban System

The Kanban system is known as a "pull" system that focuses on capacity control. It is a mechanism used for regulating the flow of production, like JIT system. Using Kanban, the production of the current stage depends upon the demands of subsequent stages, i.e. the preceding production stage must produce only the exact quantity withdrawn by the subsequent stage (Huang and Kusiak 1995). Scheer (1994) reported that Kanban is organised generally as a pick-up system: order prescriptions from the prior production level determine the further absorption of output product quantities into the production process. Figure A.31 gives a graphical representation of a general Kanban system.

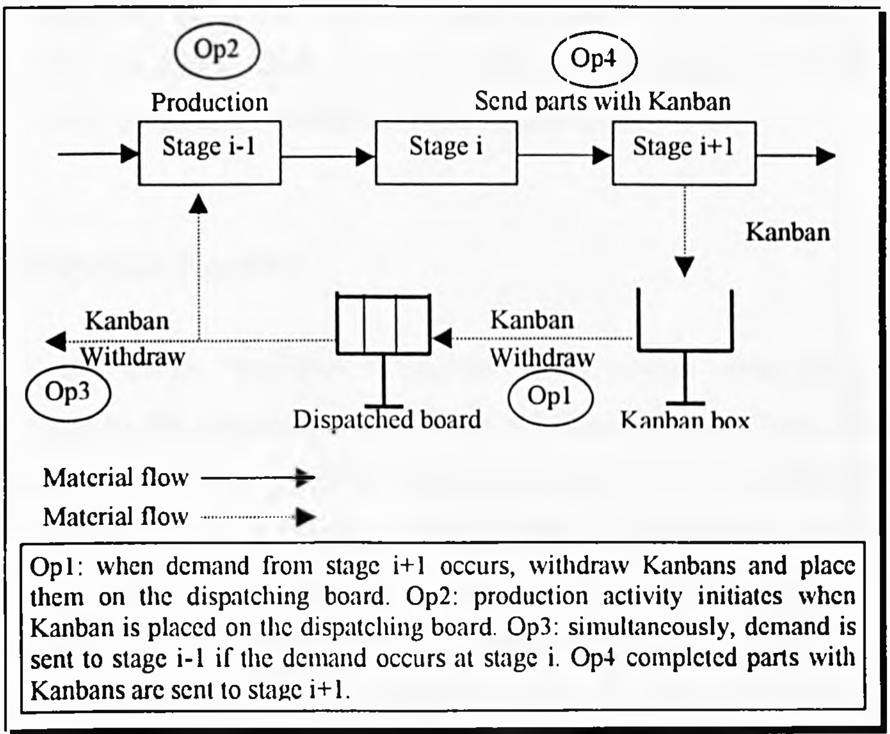


Figure A.31. The general Kanban system (Huang and Kusiak 1995).

The Kanban system was created to specify what is needed at the manufacturing stage and communicates all the production stages to each other. It is used to move materials by the usage of parts and to control WIP, production and inventory flow, based upon the organisation's production plan that is given to the final production line.

A.5.8.1 Origins of Kanban

Kanban (in Japanese) means visible record (Singh and Falkenburg 1994, Singh 1996, Slack et al. 1995). Hitomi (1996) reported that the JIT principle was mentioned by L.P. Alford in 1928, and that the pull system procedure has been employed in supermarkets for many years. He also mentioned that instruction cards like Kanbans had been proposed by F.W. Taylor J.M. Turan, who contributed to introducing quality control to Japan, developed a method, now called the Kanban system, during world war II, and brought the technique to Japan after the war; hence, it is difficult to identify the original source of JIT and Kanban ideas.

A.5.8.2 Types of Kanban

Kanban systems have been classified based upon their functions and control methods. Using the functional aspect of Kanban systems, they can be classified into three categories: the conveyance Kanban, the production Kanban and the vendor Kanban (Slack et al. 1995, Singh and Falkenburg 1994, Signh 1996).

a. The conveyance Kanban

This type is also called "withdrawal Kanban". It is carried when going from one production stage to the preceding stage. Once it brings the parts from the preceding stage and moves the parts to the following stage, it remains with the parts until the last part has been finished by the subsequent stage. Then the Kanban travels back to the preceding stage. This type of Kanban would normally have information about part number and name, the source and destination production stages, lot size, container type and capacity, routing process and the number of container released, as shown in Figure A.32.

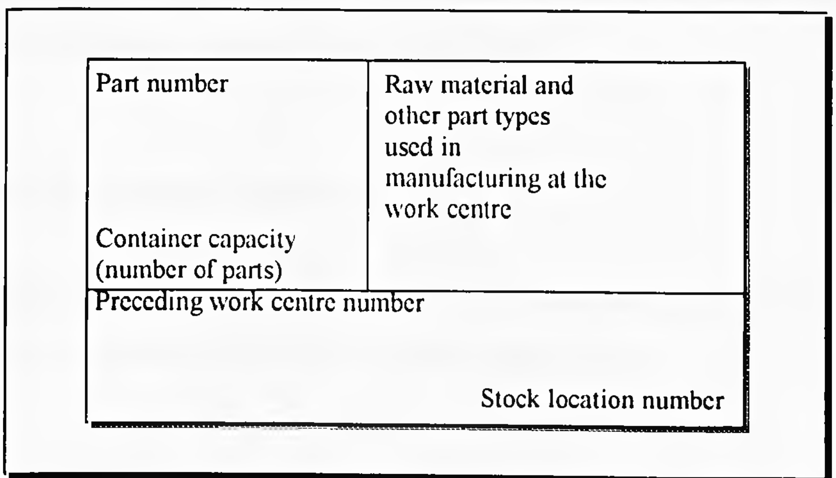


Figure A.32. Sample layout of a conveyance Kanban (Singh 1996).

b. The Production Kanban

The production Kanban is used to release an order to the preceding production stage to produce parts equal to the size specified on the card. The production Kanban card normally has details of part number and name, description of the process, the material required for the production of the part and the subsequent production stage. Figure A.33 illustrates the production Kanaden layout.

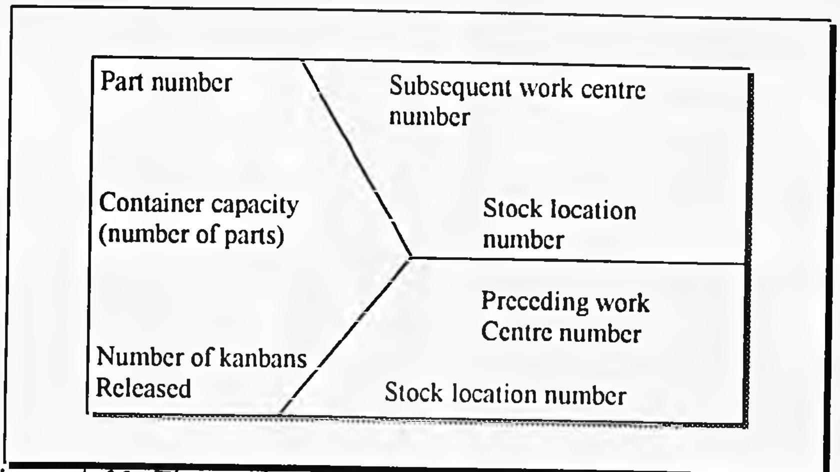


Figure A.33. The production Kanban layout (Singh and Falkenburg 1994)

c. The vendor Kanban

In general, the vendor Kanban system travels from a supplier to production stages. It is used to order parts or raw materials from suppliers and to send those materials or parts to the manufacturing stages. Hence, it is similar to the conveyance Kanban but is usually used with external suppliers (Slack et al. 1995).

A.5.8.3 Kanban Control Methods

There are three methods have been used to control Kanban systems: the single-card system, the dual-card system and the semi-dual Kanban system.

1) The single-card system

In the single-card system (single Kanban), each production station has only outbound inventory and each Kanban container is assigned a single type Kanban (Yavuz and Satir 1995). When a Kanban container is withdrawn by the subsequent production stage, the single Kanban is detached and becomes a production order for the preceding manufacturing stage. Figure A.34 illustrated the general structure of the single-card system.

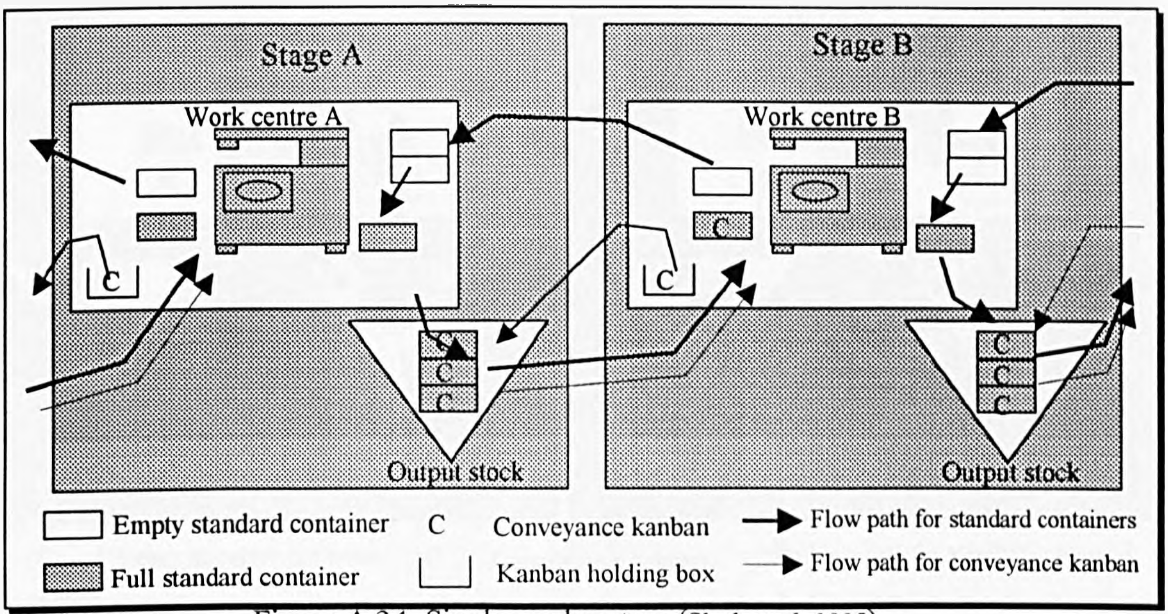


Figure A.34. Single-card system (Sleck et al. 1995).

2) The dual-card System

The dual-card system (dual Kanban system) uses two types of Kanban: the conveyance Kanban and the production Kanban to implement both station and material handling blocking using part type. The role of conveyance Kanban is to define the quantity that the subsequent stage needs to withdraw from the preceding stage. The role production Kanban here is to depict the quantity of specific part that the manufacturing stage should produce to replace those removed (Yavz and Satir 1995b). Figure A.35 illustrates the dual-card system.

3) Semi-dual Kanban System

The semi-dual Kanban system also uses the conveyance and production Kanbans. The general idea of this type is to change between the production Kanbans and conveyance Kanbans at intermediate production stages. The semi-dual Kanban system is characterised by the large WIP between the subsequent production stages, slow turnover of Kanbans and slow turnover of WIP. Figure A.36 depicts the semi-dual Kanban system.

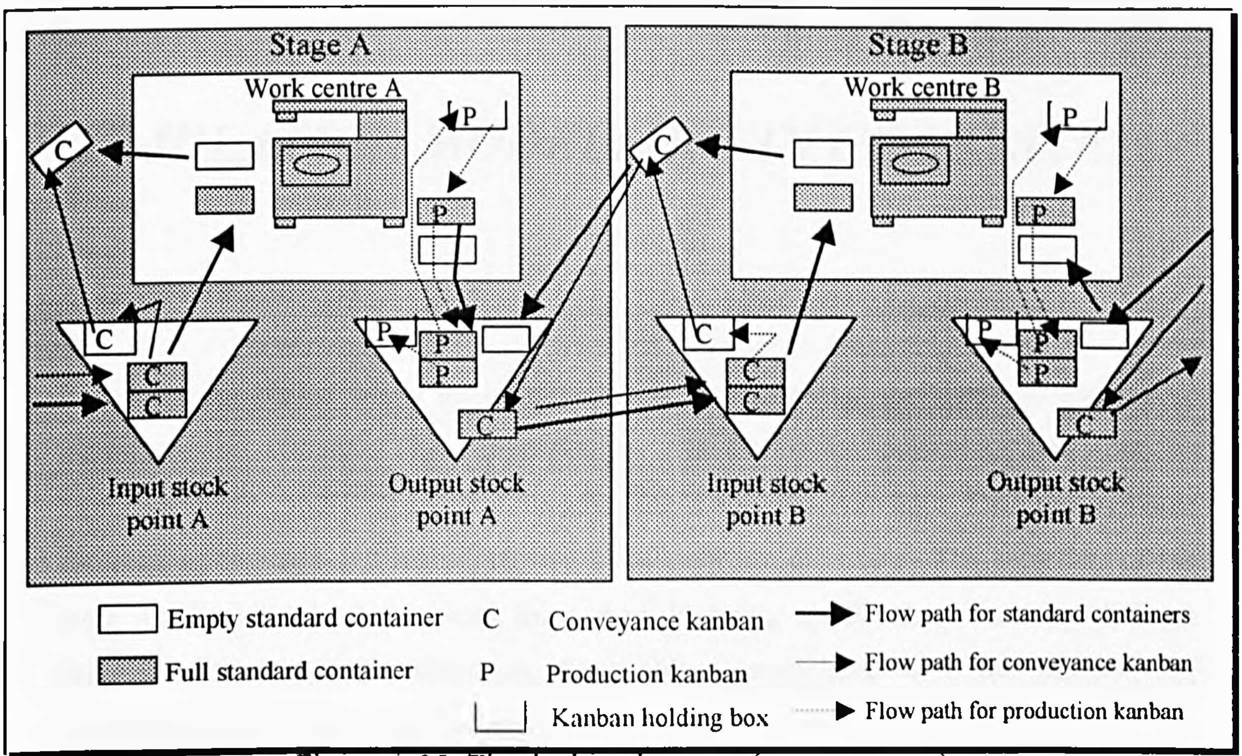


Figure A.35. The dual-card system (Slack et al. 1995).

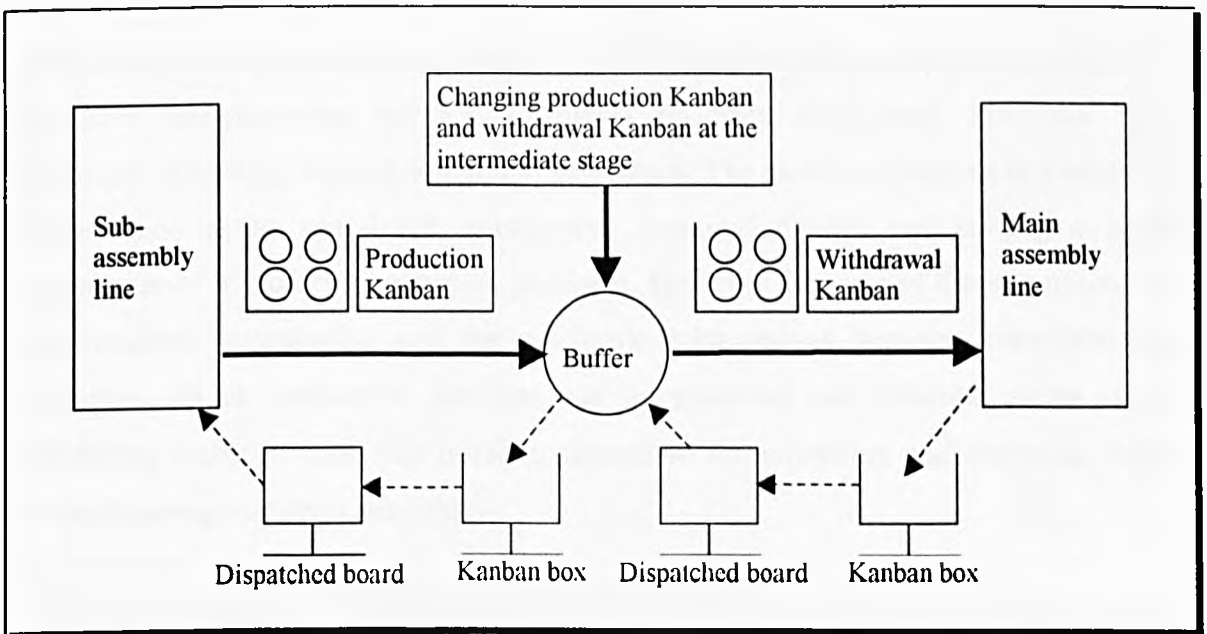


Figure A.36. The semi-dual Kanban system (Huang and Kusiak 1995).

APPENDIX-B

SIMULATION MODELLING IN CIM SYSTEMS

B.1. Introduction

Computer simulation plays an important role in modelling the dynamic aspects of manufacturing systems. This Appendix reviews simulation modelling concepts, its importance as a modelling method, its advantages and disadvantages. Different types of simulation language and manufacturing simulators are discussed. The selection of the most appropriate simulation tool for a manufacturing application is a very difficult decision. This Appendix discusses the problems associated with the selection of simulation tools for CIM applications.

B.2. Simulation

Simulation is very powerful tool which is widely used for the analysis and design of complex manufacturing systems. Chapter-4 reviewed conceptual, functional and structural modelling methodologies and techniques. The methods reviewed in Chapter 4 were found to be useful for constructing structural models and solving a large proportion of manufacturing system problems. However, because of the importance of performance, complexity, and the stochastic relationships between manufacturing activities not all real-world problems can be modelled and presented using static modelling methods. The only possible alternative for modelling and analysing these manufacturing aspects is simulation.

The term 'simulation', according to the Oxford Advanced Learner's Dictionary, means "the deliberate making of certain conditions that could exist in reality". Schriber (1987 - cited by Pegden et al. 1995) reported that "simulation involves the modelling of a process or system in such a way that the model mimics the response of the actual system to events that take place over time". Pegden et al. (1995) themselves defined simulation as "the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies to the operation of the system".

In general, simulation is a technique of imitating behavioural aspects of systems (economic, military, mechanical, etc.) using means of analogous situation, model or approaches, either to gain information more conveniently or to perform training programs (Carrie 1988). It is the most common approach for constructing models that include random and dynamic aspects and a wide variety of components, and assesses the temporal dynamics of manufacturing systems. In Chapter-2, a distinction was made between two types of manufacturing systems, namely, continuous and discrete systems. This distinction becomes important when simulation is used as there are discrete event and continuous simulation models. In a discrete simulation model, changes in the system's behaviour occur at discrete moments in time. In a continuous simulation model, time is increased through a time step mechanism using state variables which are computed at the end of each step. Discrete simulation is particularly useful at advanced stages of production design for investigation into the behaviour of various manufacturing aspects such as feasible scheduling, operating processes and layout formulation (Nicholson 1991). Simulation can be used for many applications such as system analysis and design, as well as evaluation alternative elements of manufacturing systems. In systems analysis, simulation determines the performance of a given manufacturing system. In systems design, simulation evaluates system alternatives and gives numerical outputs which can be used to select the best design alternative. A simulation model can also be used to control a manufacturing process by operating in parallel with the production system. The plant parameters are identified and used by the model according to an algorithmic or search-oriented optimisation strategy to produce the set point parameters of the manufacturing process (Rembold et al. 1993).

B.3. Why Use Simulation

Chapter-2 has discussed the fierce impact of industrial competition. Due to this competition, manufacturing industries are forced to present and develop more and improved products, many means, strategies, methodologies and tools that can be used to achieve these goals. Simulation is one of the most powerful tools which offers an efficient alternative to traditional analysis and design methods. There are various reasons why simulation proves such an efficient modelling technique (Carter and Huzan 1973, Elmaghraby 1966, Pidd 1984):

- Simulation model allows systems to be assessed without the necessity of actually implementing them in order to prevent changes being made without foreknowledge of the consequences.
- It is much quicker to simulate a period of time than to use it carry out tasks and demonstrates in real time.
- Simulation is good alternative in the absence of any compressed and complete mathematical formulation of the problem.
- Simulation may be the only possible way of experimentation because of the difficulties of conducting the experiments and observing the phenomena in reality.
- Simulation may actually support the process of innovation and discovery by imparting 'feel' for the problem.
- Simulation can be used for analysis, design and control for manufacturing systems.
- Simulation models aid the evaluation and creations of relationships between system variables.

Pritsker (1992) reported several points related to the functional levels to which simulation modelling has been applied: as explanatory devices to understand a system or problem; as a communication vehicle to describe system operations; as an analysis tool to determine critical elements, components and issues and to estimate performance measures; as a design assessor to evaluate a proposed solution and synthesise new alternative solutions; as a scheduler to develop on-line operational schedules for jobs, tasks and resources; as a control mechanism for the distribution and routing of materials and resources and finally as a training tool to assist operators in understanding system operations.

The above reasons indicate the importance of simulation modelling particularly in manufacturing systems. Therefore, using this powerful technique leads to the improvement of manufacturing resources utilisation and support decision making activities in manufacturing organisations.

B.4. Advantages and Disadvantages of Simulation

Simulation has become widely accepted for several reasons (some mentioned in the previous section). The following is a brief summary of the main advantages and disadvantages of simulation (Heizer and Render 1996, Pegden et al. 1995). The advantages of simulation include:

- It is relatively flexible and straightforward.
- It can identify bottlenecks in material, information and product flow.
- It can be used to analyse complex systems that cannot be studied by traditional methods.
- It can be used to test hypotheses about how or why certain phenomena occur.
- It allows 'what if' types of questions.
- Time can be compressed with simulation.
- It can be used to identify variables that are most important to performance and how these variables interact.

On the other hand, the disadvantages include:

- Its model construction requires specialised training;
- Good simulation tools can be very expensive;
- Simulation outputs are sometimes difficult to interpret;
- Simulation does not guarantee optimal solutions to problems like other mathematical models;
- Each simulation model is usually for a specific problem;
- Simulation model analysis can be time consuming.

B.5. Process of Simulation

Law (1990), Law and McComas (1990), Gorden (1978) and Pegden et al. (1995) have all recommended various steps of developing simulation models. However, they all agree that there are a number of steps which should be included in construction of a simulation model. These steps include:

1. Problem formulation and planning study. Law (1990) and Law and McComas (1990) reported that the statement of project objectives is one of the most important -but often neglected- aspects of a simulation study. This step of the simulation process will probably make a difference between success and failure (Pegden et al. 1995). Information and data should be collected to provide an adequate understanding of the system to be modelled. Several questions should be answered during this phase of the study:
   - What are the study objectives?
   - What are the performance measures that will be used?

- What are the problems that should be solved?
- What are the information resources that feed the study?

2. Data collection and model formulation. Data and information are collected to solve the problem. Without data, simulation models are meaningless. In the previous step of the process data resources and collection methods are identified. In this step, the actual data collection is launched to specify inputs parameters and probability distributions. This is a complex task as data need to be collected from many sources e.g. in manufacturing systems, data are collected from operators, engineers, supervisors, managers, manuals, systems, etc. for many manufacturing entities such as machines, products, WIP, buffers, capacities, etc. Data accuracy is required in modelling validation. During data collection, there are a number of matters that should be taken into account such as computer system constraints, project objectives and representation of system randomness by appropriate probability distributions.
3. Construction and verification of simulation model. In this step, the analyst must select the software that will be used for the simulation project and decide whether to use a simulation language or manufacturing simulators and construct a system model. It should be noted that the proper choice of simulation software could have a large impact on the success of the simulation study (Law and McComas 1990). The level of model details should be based upon the project objectives identified in the first step and data collected in the second step. Following model constriction, the model should be verified. In model verification, the program is tested to ensure that it works as intended, and provides a correct logical representation of the system.
4. Design of experiments. This step is carried to out to specify the system design configurations. Several aspects should be identified in this step such as:
   - Number and length of simulation runs;
   - Initial conditions of each simulation run;
   - Attribute values that need to be defined at the beginning of the simulation.

The design of simulation experiments to be of a manufacturing system requires careful consideration of such problems as the selection of factor levels and combinations, the order of experimentation and the minimisation of random error as well as other classical computer simulation experimental design problems (Burdick and Naylor 1974).

5. Output data analysis. Performance measures are estimated using the output data. These estimated performance measures indicate the efficiency of the system being studied. The output data is usually presented graphically or in simple tables.
6. Documentation and implementation of results. Methods of presenting simulation results are very important. A model's assumptions and code should be documented and the results should be implemented. The documentation and implementation are very closely connected to each other. Good documentation facilitates modifications and ensures that the model can be used to support decision-making, and present understandable descriptions of current systems (Pegden et al. 1995).

B.6. Simulation Software

This section is concerned with the selection of simulation language because this is one of the most important decisions that a model builder must make to perform a simulation study.

In general, simulation software can be divided in three major categories:

- General-purpose languages;
- Special purpose simulation languages;
- Simulators (also called manufacturing simulators).

Figure B.1 illustrates the classification of these types based upon several factors.

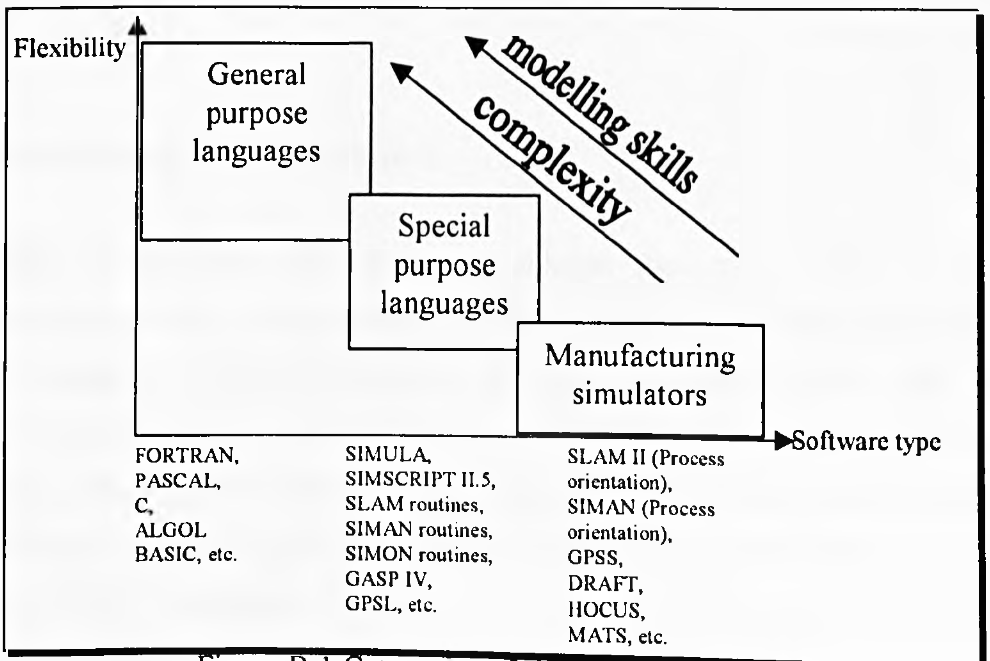


Figure B.1 Categories of simulation software.

B.6.1 General Purpose Simulation Languages

Many simulation projects are still developed using general purpose computer languages such as FORTRAN, C, Pascal, BASIC, etc. However, this requires a great deal of effort to specify entities and attributes of the system under consideration. This type of simulation software has good features such as programming flexibility, short execution time and availability. On the other hand, general-purpose languages have some limitations as they are time consuming and require programming and modelling skills. In addition, this category requires a large number of state-marks in programming (Kidera and Hoff 1974).

B.6.2 Special Purpose Simulation Language

Special purpose simulation languages are computer packages that have special features for system applications such as material handling, workstations and queuing modules. These packages are usually developed using general-purpose simulation languages such as FORTRAN or C. Examples of this category are SIMAN, GPSS/H, SLAM, etc. A simulation model is constructed by writing a computer code that includes system entities, attributes, resources, etc. The main advantages of this type are the ability to model complex systems and the availability of different modelling modules to deal with different aspects of manufacturing systems. There are some limitations to special purpose simulation languages such as the need for programming effort, a lack of communication tools, and the coding and debugging are time consuming.

B.6.3 Manufacturing Simulators

Manufacturing simulators are computer packages that can be used to simulate manufacturing systems without writing model code (Law and McComas 1992). This type of simulation software is easier to learn and has modelling tools more closely related to manufacturing systems. These are predefined simulation templates or an interactive system. Special blocks are used and defined to represent specific features of the modelling environment. Examples of this category are SIMAN/ARENA, SLAM II, SIMFACTORY, ProModel, PCModel, etc. The main advantages of this type of simulation software include:

- Model constructing time is less than for other software categories.
- Better error detection.
- Easier model checking or changes.
- Most of programming features are provided automatically.

However, there are some limitations to these packages such as inflexibility and cost. It should be noted that there are many simulation packages (see for example Anonymous (1988)). In the following sections five types of simulation packages are briefly discussed. The packages selected are SIMAN/ARENA, SIMFACTORY, SLAM II, PCModel and ProModel.

B.6.3.1 SIMAN/ARENA

SIMAN/ARENA is the result of the development of the use of SIMAN/Cinema by adding Object-Based technology and integrating several simulation aspects. It is designed to enable the user to construct a simulation model without writing a program code. A Model is constructed using a set of building blocks called modules that describe system entities. These modules are divided into three groups which make up the ARENA template.

ARENA involves two types of modelling frameworks, a system model framework and an experimental framework. The system model is constructed to describe the process and the experimental model is constructed to supply the parameters and characteristics describing the system operations. This package provides many modelling blocks for designing the system models and experiments. It also has several add-on modules that offer greater flexibility and has a good interactive graphic facility that is used to animate and to represent system entities.

This simulation package has a graphical user interface that automates the model-constructing process and reduces the amount of keyboard input required by allowing the mouse to be used for most tasks. In addition to the method of constructing a simulation model, specific data needs to be specified at the start of simulation and information collected at the end of a simulation run needs to be analysed. SIMAN/ARENA involves two integrated sub-systems, namely; the input and output processors which are used to handle inputs and outputs.

The input processor allows the model builder to take important raw data e.g. time studies, and fit a suitable probability distribution to these data. The distributions generated are directly incorporated into the model. The output processor is an analysis tool that is employed to represent graphically and compare statistically data collected during the simulation run(s). More details about ARENA can be obtained from the ARENA User's Guide (1994).

B.6.3.2 SIMFACTORY

SIMFACTORY is a data-driven simulation model of a generic factory which is used to describe the important components and constraints of the factory under consideration. It uses both graphical tools and data input to construct the model by translating this information into the simulation language SIMSCRIPT II.5. This simulation package does not require programming expertise and can be run with or without animation. SIMFACTORY is composed of four main components, the SIMFACTORY model, the user interface, the layout editor and the icon editor. The SIMFACTORY model is a generic simulation model that considers all manufacturing systems containing the same basic building-blocks such as workstations, resources, queues, etc. Hence, each factory has special block simulation and parameters. The user interface involves several menus which can be used to describe new system models, edit existing models, run simulation models and examine model outputs. The layout editor is used to create and edit the layout of a model using special graphical tools. The icon editor is used to create and edit the graphic icons using a graphical editor. More details of SIMFACTORY can be obtained from Schwab and Nisanci (1992).

B.6.3.3 SLAM II

SLAM II is a FORTRAN-based simulation package. It uses three modelling approaches; process/network, discrete event and continuous, using three different frameworks (network, discrete and continuous) .These can be used independently or combined, depending on the system problems. SLAM II employs special symbols which are contained in a data file and retrieved by a FORTRAN program containing SLAM II routines. The SLAM II model format allows the sub-programs to be added to the main program without recompiling the initial program; hence, the model can be modified and simulated without great effort (Browne and Timon 1991). Standard statistical information is collected after each simulation run.

B.6.3.4 PCModel

The PCModel is a manufacturing simulator which is designed to model the physical flow of manufacturing components through a system. It concentrates on the physical movements and locations of each component. Statistics are collected by monitoring the number of parts which move through a particular physical location and the related times (Browne and Timon 1991). PCModel comprises three main components, the source language processor, the graphical overlay editor, and the run time interpreter.

In PCModel, the model is constructed by describing the details of the system process as a sequence of logical routes, using a graphical overlay to highlight the path taken and the location of each workstation.

PCModel animation and simulation sub-systems are integrated into each other in one system. It has special features which have to be explicitly programmed; hence, models require great effort to give good accuracy. Prior to developing a model, the model builder must decide on the output type and the degree of interaction, required during the simulation run (Browne and Timon 1991).

B.6.3.5 ProModel

The ProModel was produced by the PorModel Corporation. It has a simple top-level menu which assists the user to select model editor, run simulations, view outputs or quit. The model editor assist with a number of tasks such as model definition, scheduling, resource description and handling system design and initiating the model. This package contains all standard statistical distributions used to define model variables and provides interactive animation features. A "what-if-analysis" facility is one of the most attractive features of ProModel and there is no requirement for programming skills during model constructing.

B.7. Selection of Simulation tools

Selecting a simulation tool may not be an easy decision, due to the number of simulation tools, the available different features provided and their widely varying prices (Law and McComas 1992). Several authors e.g. Pidd (1994), Grant and Weiner (1986), Law (1989), Haider and Banks (1986), Ekere and Hannam (1989), Law and McComas (1992), Schwab and Nisanci (1992), and Browne and Timon (1991) have discussed the characteristics that can be used to compare and evaluate simulation tools to guide the user to select the most appropriate package for a particular application. Two studies are reviewed in this section. The first, presented by Law and McComas (1992), presented a general discussion and suggested a number of general features for simulation tools used to design manufacturing systems. These features included:

- Modelling flexibility;
- Ease of model development;
- Execution speed;
- Combined discrete-continuous simulation aspects;
- Ability to model both 'pull' and 'push' systems;
- Ability to accept graphical model inputs.

These features can be used to develop a selection criteria for manufacturing simulation tools as illustrated in Table B.1. Other aspects should be taken into account when selecting simulation tools, such as animation, output reports, statistical capabilities, material handling modules and customer support.

1. Determine the simulation requirements of your organisation, including in your consideration:
   - Issues to be addressed by simulation both now and in the future, paying particular attention to the level of model detail (relating to required modelling flexibility).
   - Background of intended users and the frequency with which simulation will be used (related to software ease of use).
2. Develop a short list of candidate simulation packages based upon your requirements in step one (above), on features of the available software and on cost considerations.
3. Talk to several users of each product on your list to get independent assessments of software strengths and weaknesses.
4. If possible, get a 30-day free trial of each product to see how they perform on applications of particular interest to you. Alternatively, have each software vendor build in your presence a model of a test manufacturing system that is of interest to you.

Table B.1. A procedure activities for selection simulation tool

(Law and McComas 1992)

The second study was presented by Schwab and Nisanci (1992). Their selection methodology is composed of several steps, as shown in Figure B.2.

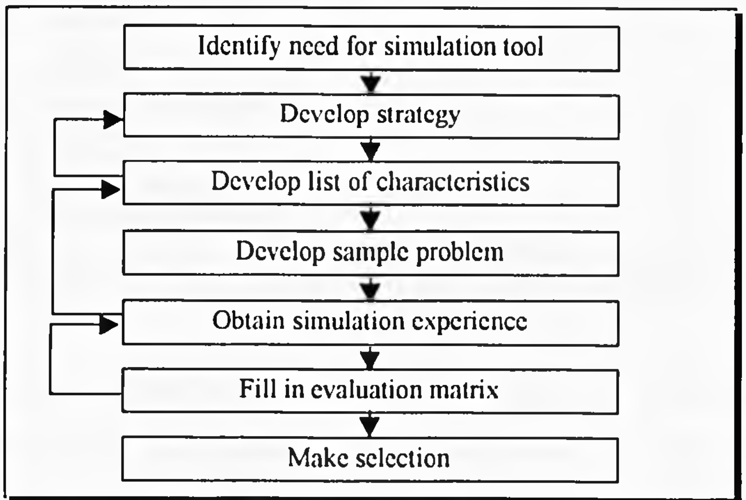


Figure B.2. Simulation software selection flowchart (Schwab and Nisanci 1992).

As illustrated in the figure above, the selection strategy is based upon the needs identified in the first step. This strategy should address the method of how simulation tools will be used, who will use the tool, how the user is trained, how the tool will be applied and how the user will be supported. Following this, a list of tool characteristics is identified based upon the application requirements and the study objectives. The next step is the development of a sample problem to test the tool characteristics and to obtain simulation experience. The evaluation matrix is filled in using the results of preceding steps. Table B.2 illustrates the software evaluation matrix. Finally, the decision can be taken based upon results obtained from the matrix evaluation.

Importance (1)	Desired characteristics	Language 1 Rank (2)	Language 1 Score (1X2)	Language 2 Rank (3)	Language 2 Score (1X3)
	LANGUAGE				
	Probability				
	Readability				
	Documentation				
	Data requirements				
	World views				
	Library of algorithms				
	User written int.				
	Interface w/CAD				
	Compatibility				
	Error recovery				
	OPERATUION				
	Applicability				
	Debugging features				
	Ease of use				
	Intuitive				
	Parameter variation				
	Model revisions				
	Interactive				
	Check points				
	OUTPUT				
	Report				
	Graphics				
	Animation				
	HARDWARE				
	Compatibility]				
	File sharing				
	Output devices				
	COMPANY				
	On-going development				
	Training				
	User support				
	Firm performance				
	Cost/value relationship				
Total scour			Scour 1		Scour 2

Table B.2. Software evaluation matrix (Schwab and Nisanci 1992).

It should be noted that there is no simulation software which is completely convenient and appropriate for all manufacturing applications (Law and McComas 1992). The decision-maker must decide on the requirements of the simulation tool based upon the factors identified. These requirements and factors are different for different applications. For example, a simulation tool selected for layout analysis may be different from that selected for scheduling or on-line control. Therefore, there is no general methodology for selecting a simulation tool identifying that the development of a knowledge-based system for simulation tool selection would be beneficial research.

B.7.1 An Example Comparison of Simulation Packages

Browne and Timon (1991) compared a number of simulation packages based upon the following factors:

1. The modelling capability;
2. The ease of use of the package;
3. The statistical output;
4. The animation capabilities;
5. The ability to link the simulation package to other software products;
6. Extra facilities.

Using these factors, simulation packages discussed in the previous sections have been compared as illustrated in Table B.3.

Factors	SIMAN/ARENA	SIMFACTORY	SLAM II	PCModel
1. Ability to model				
Assembly	Yes	Yes	Yes	Yes
Disassembly	Yes	Yes	Yes	Yes
Part failure	Yes	Yes	Yes	Yes
Machine failure	Yes	Yes	Yes	Yes
Scheduling rules	Yes	Fair	Very good	Program
Transport				
AGV	Very good	Very good	Program	Program
Conveyors	Very good	Very good	Program	Program
Conditional	Program	Yes	Yes	Yes
Non-conditional	Yes	Yes	Yes	Yes
Model limits	Very large	Large	Medium	Large
Run time	Very fast	Very slow	Fair	Fast
2. User-friendly				
Menu driven	Yes	Very good	None	Limited
Online help	None	Yes	None	Limited
Program skills	Medium	None	Medium	Medium
Model building	Hard	Very simple	Hard	Hard
Flexibility	Very high	Flexible	Very high	High
Decision rules	Program	Fair	Program	Program
Ease of debug	Easy	Easy	Hard	Hard
3. Output statistics				
During simulation	Yes	Limited	Yes	Yes
After simulation	Yes	Yes	Yes	Yes
User-defined	Yes	Partially	Yes	Yes
Fixed	No	Yes	Yes	No
Graphical	Yes	Yes	No	Program
Detail	Very high	High	Very high	Fair
Blocking reports	Program	Yes	Yes	Program,
4. Animation				
Real time	Yes	Yes		Yes
Post processor	No	No		No
Ease of construction	Very easy	Easy		Program
Icons	Very good	Good		None
Planning	Yes	Yes		Yes
Zooming	Yes	Yes		
Skip ahead	Yes	Yes		No
Speed	Fast	Very slow		Very fast
Detail	Very good	Good		Fair
Statistical output	Very good	Graphical		Fair
5. Access to external package				
High-level language	FORTRAN	None	FORTRAN	None
Database	Yes	No	Yes	No
Data-files	Yes	No	Yes	Yes
Other packages	Yes	No	Yes	Lotus

Table B.3. Comparison of simulation packages (Browne and Timon 1991).

The example illustrates the differences between simulation packages based upon several characteristics identified. This could be used by decision-makers to select one of these packages based upon results obtained.

B.8. Simulation Results

Simulation results play an important role in decision-making because the results provide performance measures for key variables which need to be identified before taking decisions. Therefore, this section includes a brief discussion of simulation results.

Simulation software should produce different performance statistics such as activity, waiting time and resource utilisation, as illustrated in Table B.4. These items should be included in the simulation report based upon study requirements.

Activity Statistics	Waiting Statistics	Resource Statistics
Average duration	Average length	Average utilisation
Minimum duration	Maximum length	Maximum utilisation
Maximum duration	Minimum wait time	Standard deviation
Entity count	Standard deviation	Current status
Current status	Current status	

Table B.4. Standard statistics for simulation output (Schwab and Nisanci 1992)

The output report of simulation results should be clear and concise. Other aspects of simulation output include graphical representations and animation. The graphical representation includes a wide variety of output charts such as bars, curves, histograms etc. Each output group can be shown using an appropriate chart i.e. a bar chart might be used for the resource utilisation, as shown in Figure B.3.

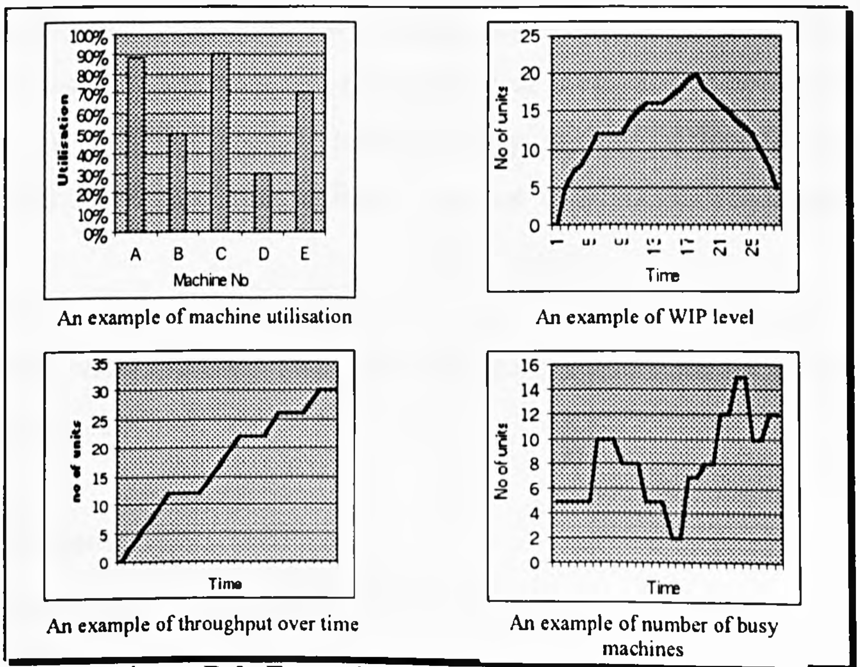


Figure B.3. Examples of simulation output charts.

Animation shows a system changing over time graphically. This feature also helps in validation and verification of simulation models. Movements of model entities on the computer screen give a good representation of the manufacturing resources. This capability can be used when studying alternatives system configurations.

B.9. Simulation Applications in CIM

Simulation has been used in many manufacturing systems applications (Talavage and Hannam 1988). It is an important tool for modelling CIM systems where it has been used for planning, analysis and design purposes. CIM components (CAD, CAM, CAPP, etc.) have also been modelled using simulation tools. Simulation is recommended as a planning tool or for optimising current operations and decisions because traditional analytical methods have proved inadequate for studying complex integrated manufacturing systems (Pegden et al. 1995). Therefore, simulation is a technique and technology that plays an important role in manufacturing systems integration.

Simulation models the behaviour of CIM systems by calculating the movements and interactions of different system elements. Because the computer can perform the moves very rapidly even for a large number of entities, the behaviour complex CIM systems can be described in a matter of minutes (Talavage and Hannam 1988). Pegden et al. (1995) regarded the planning, design, installation and operation of a manufacturing system as hinging on decisions made in the following areas: hard-system configuration decisions, soft-system configuration decisions and real-time control. Simulation is a fundamental tool for designing and analysing these areas. It can be used as a tool for design and analysis for manufacturing facility layout, equipment selection, tool selection, alternative operating policies, process evaluations, scheduling, real time control, etc.

The literature review indicates many manufacturing applications for simulation. These application includes:

- FMS;
- CIM operations;
- Material handling systems (AGV, AS/RS, conveyors, robots, etc.);
- Process Planning;

- Group Technology;
- JIT-Kanban systems;
- Inventory control;
- Management decisions;
- Machine and tool evaluations;
- Layout problem;
- Cell configuration;
- On-line control;
- Manufacturing scheduling;
- Distribution and warehouses;
- Forecasting.

It is clear from many successful applications that simulation will continue to grow in importance and become a truly operational tool for manufacturing and management decisions. Simulation can also contribute to the decision making process at all the three levels of manufacturing systems management, namely, strategic, tactical and operational, and five levels of manufacturing system controls, namely, enterprise, factory, cell, workstation and equipment.

The simulation model will become increasingly integrated with different components of CIM systems. It will be integrated with CAD, MRP II, CAPP, etc. to model the dynamic changes in the manufacturing environment by interconnecting CIM databases with simulation models. For instance, when design changes are made in the CAD system, to rearrange a layout, that information will be automatically reflected in the simulation model through a network computer system (Norman 1992). Simulation will play an important role in solving integration problems in the CIM environment. Exporting simulation output to database systems, spreadsheet, controllers, etc. offers dynamic data integration between system components. This subject is anticipated to receive more attention in future CIM research.

B.10. Discussion on Simulation Modelling

Simulation is a modelling and experimentation technique that represents the structure and dynamic behaviour of manufacturing systems (Askin and Standridge 1993).

Browne and Timon (1991) reported that data driven simulation models together with the capabilities of animation and model debugging, and emerging means of output analysis, have made simulation a useful and widely usable tool.

Some of the limitation of simulation tools for modelling manufacturing systems have been discussed in the literature. It has been found that many simulation languages are designed to support only the modelling and analysis process and material handling systems and do not cater for all CIM requirements. The inability to integrate with other systems such as databases and controllers is a further limitation of simulation languages. Consequently, the dynamic models of manufacturing systems can not use the activity structure definition contained in other static modelling methods. There is clear need to develop a modelling method to provide integration between different modelling methods to evaluate conceptual, structural and dynamic aspects.

B.11. Conclusion

Simulation modelling is one of the most important tools for the analysis and design of the dynamic aspects of manufacturing systems. This Appendix has reviewed the simulation modelling concept, its importance in manufacturing systems and different classes of simulation languages. A number of manufacturing simulators were discussed. General issues surrounding the selection of simulation tools have been discussed. This Appendix has presented two examples of how simulation tools can be selected based upon several requirements. Simulation output plays an important role in manufacturing systems because many decisions can be taken based upon the results. These results include simulation reports, graphical representations and animation facilities.

The research work using this type of modelling is still very active. It has been concluded that simulation models need to be integrated into static modelling of manufacturing systems (conceptual and structural modelling). This integration would provide a clear picture of modelling domains in manufacturing systems and would support decision activity centres at every level of manufacturing management. The integration of static and dynamic modelling methods is discussed in the next chapter.

APPENDIX-C

This appendix presents selected computer codes for the GI--SIM computerised tools. This code is related to the analysis interface used to check the grid rulers.

```
Sub GridRules ()
'you have the following data
'1- W(B) and S(B) for
   'information flow
   'decision flow

'2-Activity names cellname(i,ii)
'3-Hval2(i),Hunit2(i),Pval2(i),and Punit(i)
'4-Hval2D(i),Pval2D(i)

'1:-
'Rule (1)
'There must be at least three levels.  nho>=3
'There must at least seven columns within GRAI grid
OK(1) = True
If nho < 3 Then
OK(1) = False
Beep
Ruleno(1) = "Rule #1:"
Ruleis(1) = "        there must be at least three levels          "
form7.Label1.Caption = Ruleno(1)
form7.Label2.Caption = Ruleis(1)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
End If
OK(10) = True
If nactot < 7 Then
OK(10) = False
Beep
Ruleno(10) = "Rule #10:"
Ruleis(10) = "        There must be at least seven Functions          "
form7.Label1.Caption = Ruleno(10)
form7.Label2.Caption = Ruleis(10)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
End If

'2:-
'Rule2:-
'The horizon of upper level > horizon of the lower level H(i)>H(i+1)
'3:-
'Rule3:-
'The period of upper level > period of the lower level P(i)>P(i+1)
'4:-
'Rule4:-
'The Horizon of upper level > 2* period of the lower level H(i)>2*P(i+1)
DperY = 360
DperM = 30
DperW = 7
For i = 1 To nho - 1
If hunit2(i) = "Y" Then Hval2D(i) = Hval2(i) * DperY
If hunit2(i) = "M" Then Hval2D(i) = Hval2(i) * DperM
If hunit2(i) = "W" Then Hval2D(i) = Hval2(i) * DperW
If hunit2(i) = "D" Then Hval2D(i) = Hval2(i)
If Punit2(i) = "Y" Then Pval2D(i) = Pval2(i) * DperY
If Punit2(i) = "M" Then Pval2D(i) = Pval2(i) * DperM
If Punit2(i) = "W" Then Pval2D(i) = Pval2(i) * DperW
If Punit2(i) = "D" Then Pval2D(i) = Pval2(i)
If hunit2(i + 1) = "Y" Then Hval2D(i + 1) = Hval2(i + 1) * DperY
If hunit2(i + 1) = "M" Then Hval2D(i + 1) = Hval2(i + 1) * DperM
If hunit2(i + 1) = "W" Then Hval2D(i + 1) = Hval2(i + 1) * DperW
If hunit2(i + 1) = "D" Then Hval2D(i + 1) = Hval2(i + 1)
If Punit2(i + 1) = "Y" Then Pval2D(i + 1) = Pval2(i + 1) * DperY
```

```
If Punit2(i + 1) = "M" Then Pval2D(i + 1) = Pval2(i + 1) * DperM
If Punit2(i + 1) = "W" Then Pval2D(i + 1) = Pval2(i + 1) * DperW
If Punit2(i + 1) = "D" Then Pval2D(i + 1) = Pval2(i + 1)
OK(2) = True
If Hval2D(i) <= Hval2D(i + 1) Then
OK(2) = False
Beep
Ruleno(2) = "Rule #2:"
Ruleis(2) = "Horizon of Level #[" & i * 10 & "]must be longer than Horizon of level #[" & (i + 1) * 10 & "]"
form7.Label1.Caption = Ruleno(2)
form7.Label2.Caption = Ruleis(2)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
End If
OK(3) = True
If Pval2D(i) <= Pval2D(i + 1) Then
OK(3) = False
Beep
Ruleno(3) = "Rule #3:"
Ruleis(3) = "Period of Level #[" & i * 10 & "]must be longer than Period of level #[" & (i + 1) * 10 & "]"
form7.Label1.Caption = Ruleno(3)
form7.Label2.Caption = Ruleis(3)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
End If
OK(4) = True
If Hval2D(i) <= 2 * Pval2D(i + 1) Then
OK(4) = False
Beep
Ruleno(4) = "Rule #4:"
Ruleis(4) = "Horizon of Level #[" & i * 10 & "]must be equal at least two periods of level #[" & (i + 1) * 10 & "]"
form7.Label1.Caption = Ruleno(4)
form7.Label2.Caption = Ruleis(4)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
'MsgBox "Horizon of Level #[" & i * 10 & "] must be equal at least two periods of level #[" & (i + 1) * 10 & "]", 16, "GRAI Grid
(Rule #4)"
'Exit Sub
End If
Next i
'5:-
'Rule #5:
'H(I)>P(I) Horizon of any level most be longer then its period.
'6:-
'Rule 6:
'H >0 for Each level.
'7:-
'Rule 7:
'P >0 for Each level except the last one.
'8:-
'Rule 8:
'Every horizon is divided into equal periods.
For i = 1 To nho
If hunit2(i) = "Y" Then Hval2D(i) = Hval2(i) * DperY
If hunit2(i) = "M" Then Hval2D(i) = Hval2(i) * DperM
If hunit2(i) = "W" Then Hval2D(i) = Hval2(i) * DperW
If hunit2(i) = "D" Then Hval2D(i) = Hval2(i)
If Punit2(i) = "Y" Then Pval2D(i) = Pval2(i) * DperY
If Punit2(i) = "M" Then Pval2D(i) = Pval2(i) * DperM
If Punit2(i) = "W" Then Pval2D(i) = Pval2(i) * DperW
If Punit2(i) = "D" Then Pval2D(i) = Pval2(i)
OK(5) = True
If Hval2D(i) <= Pval2D(i) Then
OK(5) = False
Beep
Ruleno(5) = "Rule #5:"
Ruleis(5) = "Horizon of Level #[" & i * 10 & "] must be longer than its period"
form7.Label1.Caption = Ruleno(5)
form7.Label2.Caption = Ruleis(5)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
```

```
'MsgBox "Horizon of Level #[" & i * 10 & "] must be longer than its period", 16, "GRAI Grid (Rule #5)"
'Exit Sub
End If
OK(6) = True
If Hval2D(i) = 0 Then
Beep
'MsgBox "Horizon of Level #[" & i * 10 & "] must > 0", 16, "GRAI Grid (Rule #6)"
OK(6) = False
'Exit Sub
Ruleno(6) = "Rule #6:"
Ruleis(6) = "Horizon of Level #[" & i * 10 & "] must > 0"
form7.Label1.Caption = Ruleno(6)
form7.Label2.Caption = Ruleis(6)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
End If
OK(7) = True
If i < nho And Pval2D(i) = 0 Then
Beep
'MsgBox "Period  of Level #[" & i * 10 & "] must > 0", 16, "GRAI Grid (Rule #7)"
OK(7) = False
'Exit Sub
Ruleno(7) = "Rule #7:"
Ruleis(7) = "Period  of Level #[" & i * 10 & "] must > 0"
form7.Label1.Caption = Ruleno(7)
form7.Label2.Caption = Ruleis(7)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
End If
'Rule8
OK(8) = True
If Pval2D(i) > 0 Then
If (Hval2D(i) Mod Pval2D(i)) > 0 Then
Beep
'MsgBox "Horizon  of Level #[" & i * 10 & "] must be divided into equal periods", 16, "GRAI Grid (Rule #8)"
OK(8) = False
'Exit Sub
Ruleno(8) = "Rule #8:"
Ruleis(8) = "Horizon  of Level #[" & i * 10 & "] must be divided into equal periods"
form7.Label1.Caption = Ruleno(8)
form7.Label2.Caption = Ruleis(8)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
End If
End If
Next i
'9:-
'Rule 9:
'Number of periods per level should be between 5 and 20
For i = 1 To nho
If hunit2(i) = "Y" Then Hval2D(i) = Hval2(i) * DperY
If hunit2(i) = "M" Then Hval2D(i) = Hval2(i) * DperM
If hunit2(i) = "W" Then Hval2D(i) = Hval2(i) * DperW
If hunit2(i) = "D" Then Hval2D(i) = Hval2(i)
If Punit2(i) = "Y" Then Pval2D(i) = Pval2(i) * DperY
If Punit2(i) = "M" Then Pval2D(i) = Pval2(i) * DperM
If Punit2(i) = "W" Then Pval2D(i) = Pval2(i) * DperW
If Punit2(i) = "D" Then Pval2D(i) = Pval2(i)
OK(9) = True
If Pval2D(i) > 0 Then 'And i = nho Then GoTo End_if
If Hval2D(i) / Pval2D(i) < 5 Or Hval2D(i) / Pval2D(i) > 20 Then
OK(9) = False
Beep
'MsgBox "The Number of Periods per horizon of level #[" & i * 10 & "] should be between five and twenty", 16, "GRAI Grid (Rule
#8)"
'Exit Sub
'End_if:
Ruleno(9) = "Rule #9:"
Ruleis(9) = "The Number of Periods per horizon of level #[" & i * 10 & "] should be between five and twenty"
form7.Label1.Caption = Ruleno(9)
form7.Label2.Caption = Ruleis(9)
form7.Show 1
If OKOK = True Then
```

```
Exit Sub
End If
End If
End If
Next i
'11:-
'Rule 11:
'A decision frame should not go from lower level to higher level.
OK(11) = True
For i = 1 To total
Select Case i
Case flagdecis2(i) Or flagdecis3(i) Or flagdecis4(i)
If S(i) < N(i) Then
OK(11) = False
Outlinecolor(i) = red
Else
Outlinecolor(i) = Black
End If
End Select
Next i
If OK(11) = False Then
Beep
Ruleno(11) = "Rule #11:"
Ruleis(11) = "The are decision frame(s) go(es) from lower level to higher level [ Coloured RED ]"
form7.Label1.Caption = Ruleno(11)
form7.Label2.Caption = Ruleis(11)
form7.Show 1
If OKOK = True Then
updateobjects
Exit Sub
End If
'MsgBox "The are decision frame(s) go(es) from lower level to higher level [ Coloured RED ]", 16, "GRAI Grid (Rule #11)"
'updateobjects
'Exit Sub
'Else
End If
updateobjects
'12:-
'Rule 12:
' A function must contain at least one decision centre.
For i = 1 To nactot
OK(12) = False
For ii = 1 To nho
If cellname(i, ii) > "" And celltype(i, ii) = 3 Then
OK(12) = True
End If
Next ii
If OK(12) = False Then
Beep
Ruleno(12) = "Rule #12:"
Ruleis(12) = " Function #[" & i * 10 & "] must contain at least one decision centre"
form7.Label1.Caption = Ruleno(12)
form7.Label2.Caption = Ruleis(12)
form7.Show 1
If OKOK = True Then
Exit Sub
End If
'MsgBox " Function #[" & i * 10 & "] must contain at least one decision centre", 16, "GRAI Grid (Rule #12)"
'Exit Sub
End If
Next i
'13:-
'Rule 13:
'A level must contain at least one decision centre.
'OK(13) = False
For ii = 1 To nho
OK(13) = False
For i = 1 To nactot
If cellname(i, ii) > "" And celltype(i, ii) = 3 Then
OK(13) = True
End If
Next i
If OK(13) = False Then
Beep
Ruleno(13) = "Rule #13:"
Ruleis(13) = " Level #[" & ii * 10 & "] must contain at least one decision centre"
form7.Label1.Caption = Ruleno(13)
form7.Label2.Caption = Ruleis(13)
form7.Show 1
```

```
If OKOK = True Then
Exit Sub
End If
End If
Next ii
'14:-
'Rule 14:
'Avoid decision frame duplication
OK(14) = True
For i = 1 To total
If flagdecis1(i) = i Then
    For ii = 1 To total
    If i = ii Then ii = i + 1
    If flagdecis1(ii) = ii Or flagdecis2(ii) = ii Or flagdecis3(ii) = ii Or flagdecis4(ii) = ii Then
      If (W(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (W(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
And (N(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (N(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho +
1)) And (S(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (S(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho
+ 1)) And (E(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (E(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
Then
      OK(14) = False
      Outlinecolor(ii) = Blue
      End If
    End If
    Next ii
End If
Next i
For i = 1 To total
If flagdecis2(i) = i Then
    For ii = 1 To total
    If i = ii Then ii = i + 1
    If flagdecis1(ii) = ii Or flagdecis2(ii) = ii Or flagdecis3(ii) = ii Or flagdecis4(ii) = ii Then
      If (W(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (W(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
And (N(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (N(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho +
1)) And (S(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (S(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho
+ 1)) And (E(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (E(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
Then
      OK(14) = False
      Outlinecolor(ii) = Blue
      End If
    End If
    Next ii
End If
Next i
For i = 1 To total
If flagdecis3(i) = i Then
    For ii = 1 To total
    If i = ii Then ii = i + 1
    If flagdecis1(ii) = ii Or flagdecis2(ii) = ii Or flagdecis3(ii) = ii Or flagdecis4(ii) = ii Then
      If (W(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (W(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
And (N(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (N(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho +
1)) And (S(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (S(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho
+ 1)) And (E(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (E(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
Then
      OK(14) = False
      Outlinecolor(ii) = Blue
      End If
    End If
    Next ii
End If
Next i
For i = 1 To total
If flagdecis4(i) = i Then
    For ii = 1 To total
    If i = ii Then ii = i + 1
    If flagdecis1(ii) = ii Or flagdecis2(ii) = ii Or flagdecis3(ii) = ii Or flagdecis4(ii) = ii Then
      If (W(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (W(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
And (N(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (N(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho +
1)) And (S(i) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = (S(ii) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho
+ 1)) And (E(i) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = (E(ii) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1))
Then
      OK(14) = False
      Outlinecolor(ii) = Blue
      End If
    End If
    Next ii
End If
Next i
If OK(14) = False Then
```

```
        Beep
        Ruleno(14) = "Rule #14:"
        Ruleis(14) = "Not allowed to duplicate the decision frame from the same source to the same destination [coloured blue]"
        form7.Label1.Caption = Ruleno(14)
        form7.Label2.Caption = Ruleis(14)
        form7.Show 1
        If OKOK = True Then
        updateobjects
        Exit Sub
        End If
Else
For i = 1 To total
If flagdecis1(i) = i Or flagdecis2(i) = i Or flagdecis3(i) = i Or flagdecis4(i) = i Then
If Outlinecolor(i) = Blue Then Outlinecolor(i) = Black
End If
Next i
End If
updateobjects
'15:-
'Rule 15
'Each level of decision must generate at least one decision frame at lower level.
For ii = 1 To nho - 1
OK(15) = False
For i = 1 To nactot
   If celltype(i, ii) = 3 And cellname(i, ii) > "" Then
   For j = 1 To total
   If flagdecis1(j) = j Or flagdecis2(j) = j Or flagdecis3(j) = j Or flagdecis4(i) = j Then
      If (N(j) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = ii And (S(j) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) /
(nho + 1)) > ii Then
      OK(15) = True
      End If
  End If
  Next j
  End If
Next i
        If OK(15) = False Then
        Beep
        Ruleno(15) = "Rule #15:"
        Ruleis(15) = "A decision level #[" & ii * 10 & "] must generate at least one decision frame at a lower level"
        form7.Label1.Caption = Ruleno(15)
        form7.Label2.Caption = Ruleis(15)
        form7.Show 1
        If OKOK = True Then
        Exit Sub
        End If
        End If
Next ii
'16:-
'17:-
'Rule 16 -17
'A source of a decision frame must be a decision activity.
For j = 1 To total
OK(16) = False
OK(17) = False
   If flagdecis1(j) = j Or flagdecis2(j) = j Or flagdecis3(j) = j Or flagdecis4(i) = j Then
      For i = 1 To nactot
      For ii = 1 To nho
      If cellname(i, ii) > "" And celltype(i, ii) = 3 Then

      If (N(j) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = ii And (W(j) - pagex11) \ Abs((pagex22 - pagex11) /
(nactot + 1)) = i Then
      OK(16) = True
      End If

      If (S(j) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) = ii And (E(j) - pagex11) \ Abs((pagex22 - pagex11) / (nactot
+ 1)) = i Then
      OK(17) = True
      End If
      End If
      Next ii
      Next i
   If OK(16) = False Or OK(17) = False Then
   Outlinecolor(j) = Green
      Beep
      Ruleno(16) = "Rule #16-17:"
      Ruleis(16) = " Source and destination of decision frame(s) must be a decision activity [Coloured Green]."
      form7.Label1.Caption = Ruleno(16)
      form7.Label2.Caption = Ruleis(16)
      form7.Show 1
```

```
        If OKOK = True Then
        updateobjects
        Exit Sub
        End If
    Else
    Outlinecolor(j) = Black
    End If
    End If
Next j
updateobjects
'18:-
'Rule 18
'A decision frame should not pass a decision centre at a lower level.
For j = 1 To total
OK(18) = True
    If flagdecis1(j) = j Or flagdecis2(j) = j Or flagdecis3(j) = j Or flagdecis4(i) = j Then
        For i = 1 To nactot
        For ii = 1 To nho
        If cellname(i, ii) > "" And celltype(i, ii) = 3 Then
        If (N(j) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) < ii And (W(j) - pagex11) \ Abs((pagex22 - pagex11) /
(nactot + 1)) = i And (S(j) - pagey11) \ (Abs((pagey22 - pagey11 - 300)) / (nho + 1)) > ii And (E(j) - pagex11) \ Abs((pagex22 -
pagex11) / (nactot + 1)) = i Then
        OK(18) = False
        Beep
        Ruleno(18) = "Rule #18:"
          Ruleis(18) = "A decision frame should not pass a decision centre at level #[" & ii * 10 & "]."
          form7.Label1.Caption = Ruleno(18)
          form7.Label2.Caption = Ruleis(18)
          form7.Show 1
          If OKOK = True Then
          Exit Sub
          End If
        End If
        End If
        Next ii
        Next i
    End If
Next j
'19:-
'Rule 19
'A decision frame should never pass into planning centre.
OK(19) = True
For i = 1 To nac
If Label10(i * 10).Caption = "To Plan" Then
For j = 1 To total
If flagdecis1(j) = j Or flagdecis2(j) = j Or flagdecis3(j) = j Or flagdecis4(i) = j Then
If (E(j) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) = i And (W(j) - pagex11) \ Abs((pagex22 - pagex11) / (nactot + 1)) <> i
Then
OK(19) = False
        Beep
        Ruleno(19) = "Rule #19:"
          Ruleis(19) = "A decision frame should never pass into planning centre (Check level #[" & ((S(j) - pagey11) \ (Abs((pagey22
- pagey11 - 300)) / (nho + 1))) * 10 & "] )."
          form7.Label1.Caption = Ruleno(19)
          form7.Label2.Caption = Ruleis(19)
          form7.Show 1
          If OKOK = True Then
          Exit Sub
        End If
End If
End If
Next j
End If
Next i
MsgBox "All GRAI Grid Rules are checked" & Chr(10) & " " & Chr(10) & " ", 64, "    GRAI  GRID  RULES"
End Sub
```

APPENDIX-D

BROOK HANSEN MOTORS: BACKGROUND

D.1. Introduction

This Appendix gives a background to the industrial company selected for a case study in this research. The company selected is Brook Hansen Motors. The industrial company was established in 1903 but it has been changed and redeveloped various times during its long life. In this Appendix, a brief history of Brook Hansen Motors is presented and relevant details of the Huddersfield manufacturing site are provided. The company has several manufacturing sites, but the Huddersfield site represents the headquarter of this large company today.

The main objective of this Appendix is to describe the industrial case study selected and to lay the foundations for demonstrating the use of the information collected during company visits.

D.2. History and Development

A young electrical engineer Mr E. Brook started up his own business with £300 in 1904. In 1954, he had become the leading alternating current motor manufacturer in the world with an annual turnover of over £4,000,000 and employing over 2000 people. Mr E. Brook wound his first motors in a small first floor warehouse room, in a building in Threadneedle Street, Huddersfield. The first motor was 100 cycles single phase, this being the current supply in Huddersfield at the time. He gradually increased the number of employees to eight and Brook Motors become an accepted part of local industrial life. The business become a private Limited Company, E. Brook Ltd, on the 10th August 1905. In the same year a three phase current supply became available in the town and the staff was increased to 20 persons. More and more orders were obtained;

and a move to a larger premises became necessary to cope with the expanding trade. A desirable place was acquired at "Nelson" Mill, Colne Road; and the works moved there in November 1905. The firm commenced the manufacture of three-phase motors and control gears in June 1906.

Business continued to increase over the next few years. In 1911, a branch workshop and showroom was opened in Leeds for the sale of electric motors. A further branch workshop and showroom was opened in London in 1912. During the period 1912-1914, agents were appointed in Birmingham, Cardiff, Sheffield, Glasgow, Wakefield, Plymouth, and abroad in New Zealand, Australia and Canada. The Colne Road site had become inadequate to meet the demand for products and larger premises were needed to accommodate the work in hand. In 1915, a major decision was taken to build a first-class engineering works and a suitable site was found in St.Thomas Road, Huddersfield. All the machinery was removed from Colne Road and put to work at St. Thomas Road. The new works was completed in 1917. This became the international home of Brook Motors and still remains so today. In 1918, workforce numbered 200, and work was planned to give easy flow through the factory with the least possible loss of time and labour. Motors up to 200-horse power were manufactured, 50 per week of various sizes were being made.

Throughout the war period, 1914-18, the factory worked to full capacity and contribution of E. Brook Limited was substantial in the electrical industry. The large premises in St. Thomas Road had a prospective output of 200 motors a week although in reality it finally handled 800 a week. Output continued to increase and the 20,000th motor was completed in 1918. Thirty thousand motors had been completed in 1921 and 40,000 four years later in 1925. There were 237 employees at that time at Brook Motors.

In 1927, the name of the firm was changed to Brook Motors Limited and a profit sharing scheme was introduced. After the scheme had been in operation a few years, 54 percent of the employees were shareholders and held 67 percent of the total number of the shares issued.

In 1929, other extensions were made and in 1932 conveyor systems were established throughout the factory, and all goods were conveyed from the initial assembly to the final packing at working height without the necessity of lifting on the part of the workforce. In 1933, 100,000 motors were manufactured and in 1935, Brook Motors commenced making their own plastic mouldings for both motors and control gear.

By the end of 1936, the firm was employing over 420 persons and made over 18,000 motors. Three years later the 200,000th motor was made. Following the general trend of individual drives for machines, it became apparent that smaller motors would be needed for industrial machines and the growing number of domestic appliances. Further extensions became necessary and a total floor area of 70,000 square feet was opened on 18th February 1939. The main building was of the two storey types. The ground floor housed three plants manufacturing plastic components, sheet steel fabrication and cotton coverings for wire. The impregnation plant was also housed in this building and further space was occupied by transport and a fully equipped maintenance factory, and sections devoted to the production machining of rough casting.

In 1941, No. 3 Works (this area is still called No. 3 works today) was erected to fulfil the need for floor space to handle the increasing demand and output rose to 1500 motors per week.

A major change in the finish of motors came in 1945 when it was necessary to treat all the parts to withstand tropical conditions. All difficulties were overcome and the treatment became standard for many motors produced. In 1947, due to continuing development, the Huddersfield factory was eventually able to produce 750 standard and 1000 "cub" (fractional) motors each day.

Motor number 1,000,000 rolled off the production line in November 1950. It was an early model from the recently introduced BREMA range of motors specially designed from the Canadian and USA markets. In the same year, Brook Motors became a public company.

Further factory extensions were purchased, a new Shipping and Despatch Department was built (No. 5 Works). The number of employees rose steadily to nearly 500. In 1945, a repair factory (No. 8 works) was established in another building adjacent to the main Huddersfield factory, to cater for an increasing demand for a quick and efficient and repairs service.

By January 1958, 2 million motors had been produced and later in 1959, lamination manufacturing was transferred from Huddersfield to Honley, when a larger and more up to date press shop and an extra bay was added to the main factory. In 1965 the company opened a factory in Rochdale which specialised in the production of large quantities of stator and rotor units for many large companies who buy consignments regularly, usually on a contract basis.

In 1968, a further winding factory at New Rossington (near Doncaster) was opened and eventually provided employment for around 200 people, mainly in the production of wound stators for the smaller range of electric motors.

The 10,000,000th electric motor was produced in 1968 and the Queen's award to Industry for exports was awarded to the company. In 1969, 3300 people were employed by the company across the Huddersfield, Honley, Barnsley and Rochdale manufacturing sites. In 1970 Gryphon motor production (fractional) was between 900 and 1000 a day. In 1970 Brook Motors Ltd, became a wholly owned subsidiary of the Hawker Siddeley Group P.L.C who had earlier acquired Crompton Parkinson Ltd, also manufacturers of AC electric motor. Considerable changes took place in the period from the mid 1970's to the early 1980's, due chiefly to trading difficulties brought about by the world recession and fierce competition.

In order to substantially reduce costs, the manufacturing activities had to be concentrated at fewer sites. Consequently, the Barnsley winding also was closed and the works transferred back to Huddersfield, the Rossington Winding factory closed and joined Doncaster (Crompton Parkinson) and the unit production at the Rochdale factory was moved to Honley. Brook control gear, which occupied part of the Barnsley Site, acquired new premises at Wakefield and commenced trading as a separate company. The Product range was redesigned and the products were marketed more aggressively by way of price, quality and service levels. During 1983/4, a significant slice of the market share was regarded by Brook Motors.

Following BTR's (British Tyre & Rubber Co Ltd.) acquisition of the Hawker Siddeley group in November 1991, the BTR Engineering Group was formed a year later and in 1993 BROOK HANSEN was launched. It brought together under a single banner a well-known international name in electro-mechanical drives and controls.

D.2.1 Crompton Parkinson Limited

Brothers Frank and Albert Parkinson, in neighbouring Guiseley, were simultaneously undertaking pioneering work in the development of AC. motor production, comparable to that of E. Brook. Having expanded into their third factory on the present site in Netherfield Road by 1913, they emerged as F. & A.E. Parkinson Ltd.

In 1927, this company merged with R.E.B Crompton & Co. which had been founded much earlier in 1878 by Colonel Crompton. Another Yorkshire man, he was a pioneer in the development of DC motors. The world's first fully electrically-lit house had been powered by a Crompton dynamo, as had lighting plants for Buckinham Palace, Windsor Castle, Crystal Palace and Kings Cross Station. Also, Col. Crompton's company had built most of the locomotives for the world's first underground railway, in London.

With the formation of Crompton Parkinson Ltd, a period of diversification began through the acquisition of British Electric Transformer Co., Derby Cables, Associated Electric Vehicles Manufacturers and Young Accumulator Co.

D.2.2 Newman Electric Motors Limited

Newman Electric Motors Limited was founded at a later stage by brothers Augusts James and Hedley Newman, who had earlier been involved in the repair of DC tramcar motors, in Bristol. They also moved into AC motor production. Their first units were produced in 1937.

D.2.3 Hawker Siddeley Group plc

The integration of the electric motor activities of Hawker Siddeley and Crompton Parkinson Ltd led to the formation of Brook Crompton Parkinson Motors in 1974. Newman Electric Motors was acquired in 1987. All these developments resulted in the production of electrical rotating machinery being centred on four UK sites: Huddersfield, Guiseley, Doncaster and Honley plus two overseas locations.

D.2.4 Hansen Transmission International

This company's history can be traced back to 1923 and its first complete gear units were manufactured in 1934. David Hansen's success in achieving standardisation in the design and manufacture of industrial gear units, represented by the launch in 1950 of the Hansen Patent I speed reducer, was a 'world first', leading to significant improvements and delivery times. The latest, fourth generation Hansen P4 gear units created another 'industry first' in having been developed right from conceptual design to manufacture in accordance with ISO 9001 quality standards and achieving a further up-rating of some 25% for a given size of unit.

The main centre of research, development and gear manufacture is at Edegem (Antwerp), Belgium. Hansen Transmissions International became part of BTR plc in 1983, with the later group's acquisition of Thomas Tilling Ltd. In 1992, Hansen acquired Parger Inc. of New Orleans, USA, which designs, manufacturers and services gear units used in America's pulp and paper, mining and petrochemical industries.

D.3. Brook Crompton Today

Today, Brook Crompton is part of **Brook Hansen** within the BTR Group which is a world-wide group. Figure D.1 illustrates the Brook Hansen organisation. Based in the UK, it employs over 125,000 people in more than 1,000 operations in some 40 countries. BTR products are made by over 1,200 companies world-wide and fall broadly within the following markets:

- Automotive and aviation components
- Batteries, electronics and electrical motors
- Retail sports equipment and leisure wear
- Paper making equipment
- Transportation
- Construction
- Packing and distribution
- Polymer products
- Industrial engineering products
- Meters and valves

- and many others

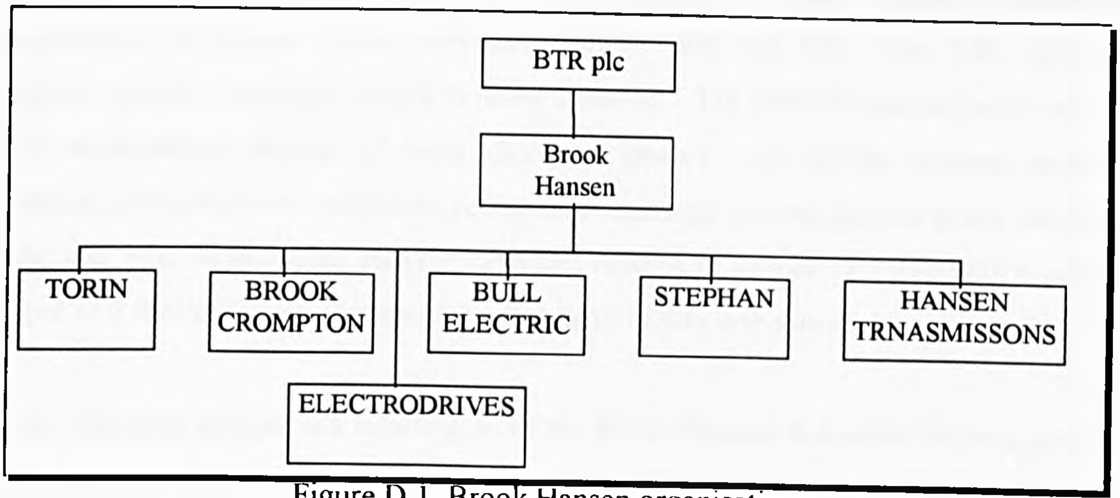


Figure D.1. Brook Hansen organisation.

D.4. Manufacturing Systems of Brook Hansen Motors

Brook Hansen Small Industrial Motors is a division of Brook Hansen International and one of Europe's largest and most successful electric motor manufacturers. Brook Hansen International is rapidly developing a significant global presence as part of BTR's (British Tyre and Rubber Co Ltd) Power-drives group. In addition to 7 UK manufacturing sites, Brook Hansen International has production, assembly and service centres in Europe, Australia, North America, South Africa and India, and markets its products through subsidiary companies in Australia, Canada, USA, Asia and the Far East, France, Belgium, Germany and Sweden. The company also provides local sales and service support through a substantial network of agents and distributors, as shown in Figure D.2.

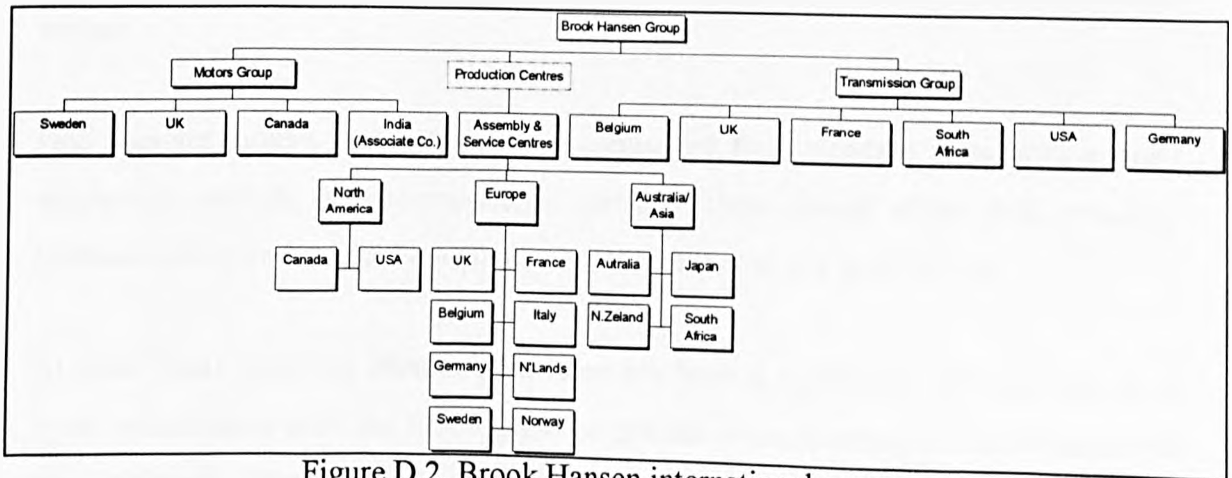


Figure D.2. Brook Hansen international group.

As illustrated in Figure D.2, Brook Hansen is spilt into two main divisions; the motors group and the transmissions group. The motors group is further organised into four geographical divisions; India, Sweden, Canada and the UK. The UK division manufactures AC standard motors in frame sizes 63 - 355 with aluminium, cast iron or steel construction, arrange of hazardous area motors, and special purpose motors. Applications include air movement, refrigeration, pumps, cranes, process plant, machine tools, etc. The division also manufactures DC motors in frames 71 - 710 with a power output of 2,000 kW, special purpose control gear, brakes and thrusters.

Small Industrial Motors is a subdivision of the Brook Hansen Industrial Motors group - manufacturing AC motors in frame sizes 63 - 225 at two sites: Huddersfield and Honley. Huddersfield, the larger of the two sites, manufactures aluminium, cast iron and flameproof motors mainly in frame sizes 100 to 180 - although a small volume of larger motors are manufactured in sizes 200 - 250. In addition, the site provides administrative support to the Honley factory. Honley manufactures aluminium and cast iron motors in frames 63 - 90 and motor interiors.

Both the Huddersfield and Honley sites manufacture in the region of 800 motors per day; approximately half of these motors are sold from stock (standard motors) and half are made to customer order (non-standard motors).

Standard motors are sold from stock and are usually on short delivery. MRP forecasts aim to replace stocks as sold. However, some 40% of the stock motors are modified before despatch i.e. some simple standard parts are modified to configure the motor to customer requirements. In addition, a number of motors are stocked for specific customers (customer standard stock motors) accounting for 20% of total standard motors.

Non-standard motors are predominantly configured from standard parts, with a small number of specially made components; typically, these consist of the shaft, winding, terminal box or end-shield. Non-standard motors are sold at a specific time.

At both Small Industrial Motors sites there has been a significant shift towards short cycle manufacture with the introduction of cellular manufacturing at the Huddersfield site (although Honley's business requirements favour more traditional flow line production methods). Kanban and MRP production systems have been used on the different manufacturing locations in Huddersfield site.

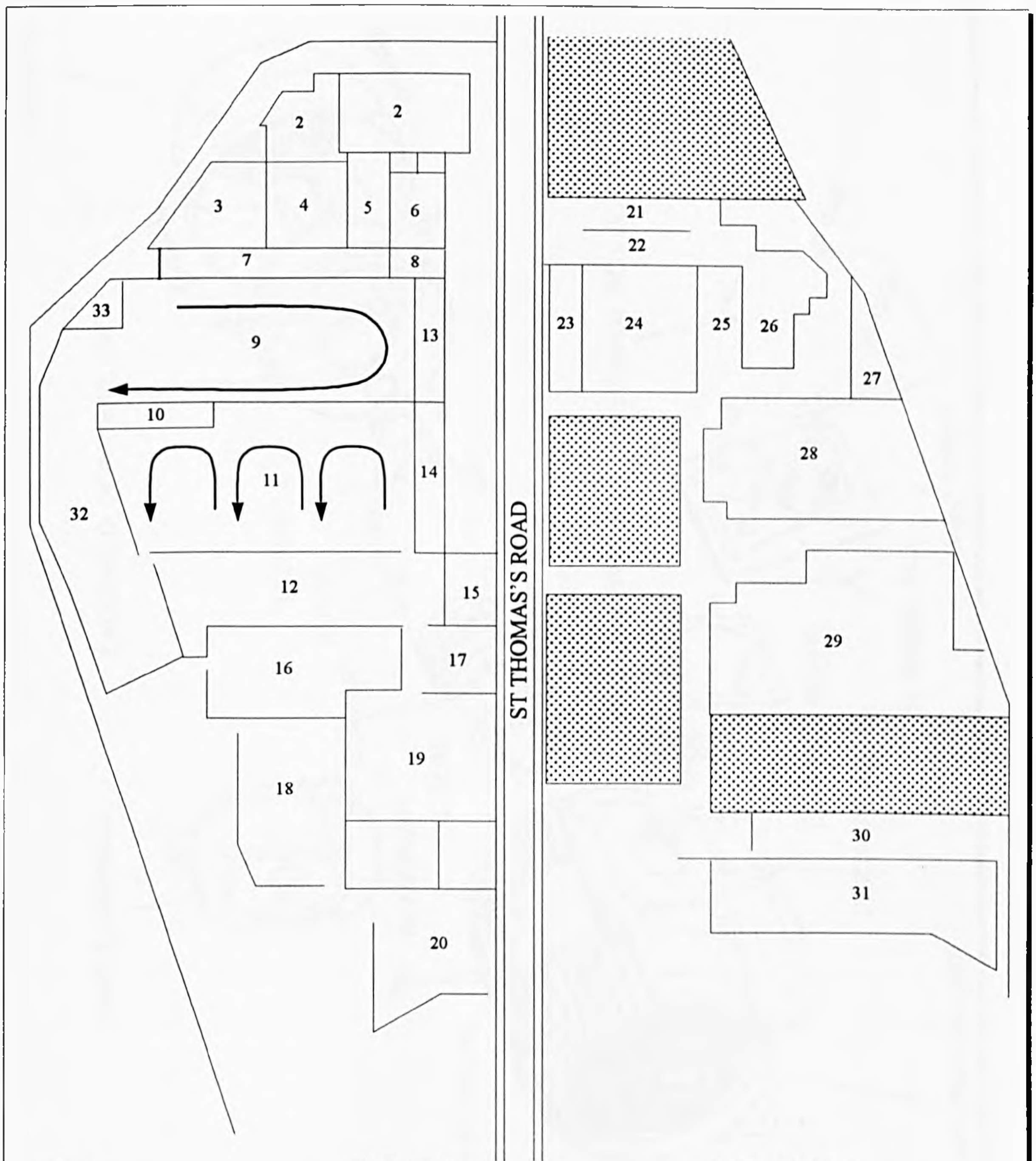
The development and introduction of the group's new range of high efficiency motors, the W range (World Series), is well advanced. This has involved a massive engineering change exercise rationalising product structures and design features. The W product range will simplify both the product and its manufacture in the longer term. However, its phased introduction, alongside existing product types, has created significant operational challenges.

D.4.1 Huddersfield Manufacturing Site

The Huddersfield site is the centre of operations. About 719 employees work in different departments and plants in this manufacturing site. Figure D.3 illustrates the general plan of Brook Hansen Motors in Huddersfield. Five manufacturing cells (100, 112, 132, 160-180 and cast iron motors) represent the major production lines which are fed by the other plants manufacturing motor components and sub-assemblies. Figure D.4 illustrates a general structure of an electric motor.

Two types of production are applied in this site; Make-To-Stock (MTS) and Make-To-Order (MTO). In general, standard motors are sold from stock and manufactured against stock replenishment requirements. Non-standard motors are manufactured to meet customer orders and need to be fully specialised for each new order. The distinction between standard and non-standard motors is not always clear and in addition some non-standard motors are stocked for particular customers. Some motors are produced as stock modifications to achieve some customer specifications. In most cases, standard and non-standard motors are managed using completely separate administrative and computer systems.

The Huddersfield site of small electric motors division consists of eight main organisation departments (purchasing, commercial, site services, manufacturing, engineering, financial, quality, and personal) under a general managing director. Figure D.5 illustrates the management organisation structure of the motor division.



1 - Canteen
2 - No. 2 Rotor Casting
3 - Loose Lams., Insulation Prep & Plastics
4 - Stores & Special. Machined Shafts
5 - Rotor Sub-Assembly
6 - Integrated Shaft Production Unit
7 - High Bay Racking
8 - Weighbridge
9 - 160-180 Assembly
10- Stock Motor Warehouse
11- 100 Cell, 112 Cell And 132 Cell
12- Cast Iron Assembly
13- C.P.O. Drafting Office Computer Department
14- Purchasing & W Series
15- Personal And Industrial Relations
16- Cast Iron Machining
17- Casting Receiving
18- Export Packing, And Warehouse & Despatch
19- Cast Iron Ends Machining
20- Garage
21- Plant Maintenance & Planning Workshops
22- Plant Maintenance & Aluminium Store
23- Sales And Accounts
24- Spares, Repairs & Special Motor Assembly
25- Scrap Storage
26- Jig And Gauge
27- Machining
28- Pressure Die Casting
29- Hansen Transmissions
30- Special Motor Assembly
31- Motor Rewind
32- Despatch
33- Load Test

Figure D.3. Brook Hansen site plan in Huddersfield.

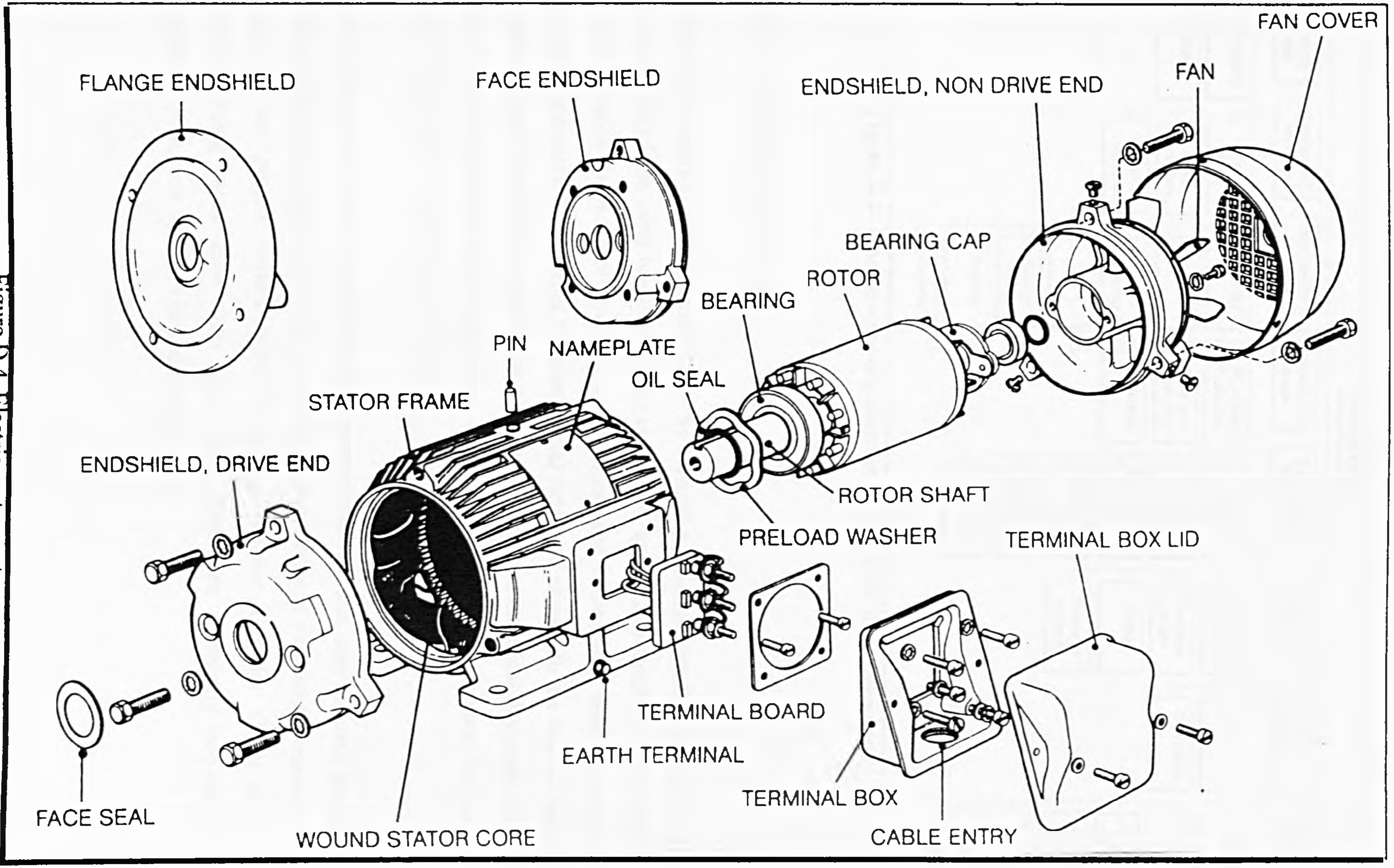


Figure D.4. Electric motor parts.

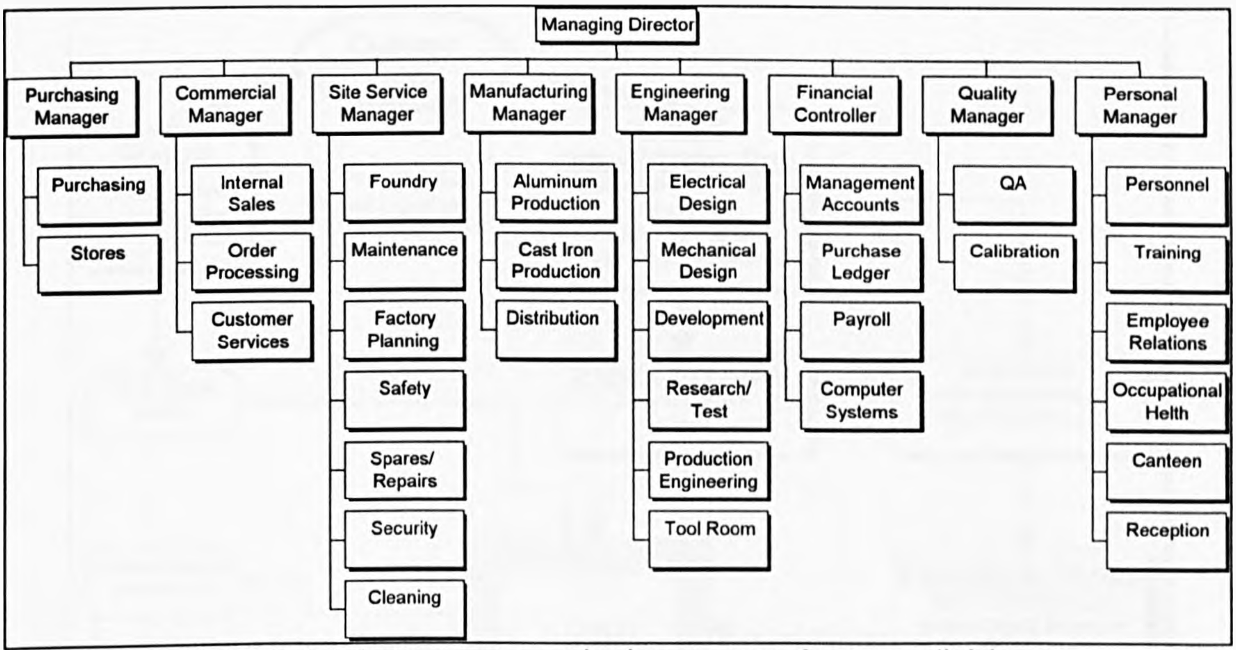


Figure D.5. Management organisation structure for motor divisions.

D.4.1.1 Order Processing

Orders are received at Huddersfield site by Central Administrative Department (order process unit) from sales branches, some customers, agents and stockists. The Central Administrative Department involves an electrical and mechanical team, as well as people responsible for entering the commercial details of orders. In the case of new or difficult requirements, referrals are made to mechanical and electrical designers. Figure D.6 illustrates that customer enquires are handled by the sales department, the project department and the sales branch office.

Motor orders in Brook Hansen are split between standard and non-standard products. The average split tends to be 50/50, with a small proportion of non-stock processed and controlled as 'project' orders, typically supplied against a tender. Order entry is different being based upon order type. Order types are essentially non-standard, standards, premiums, spares, kits and Honley units.

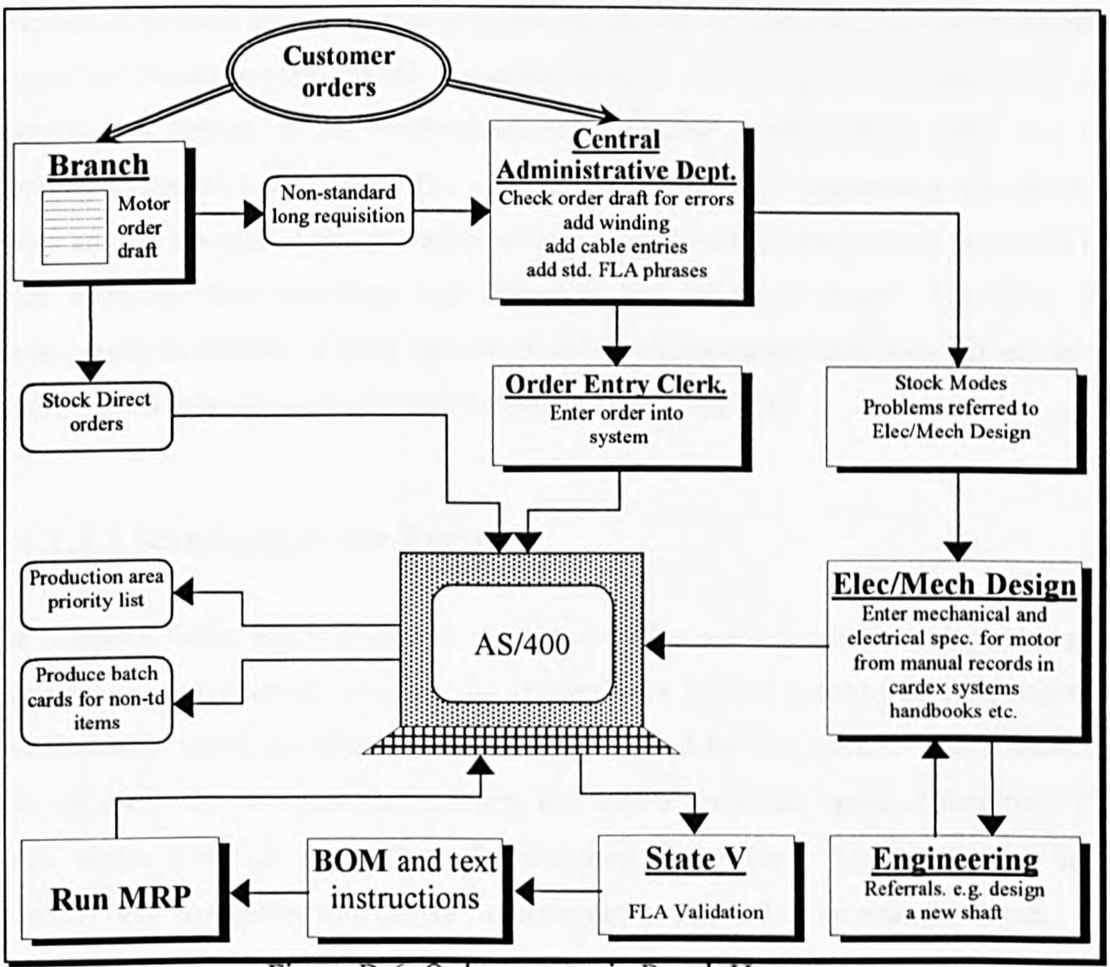


Figure D.6. Order process in Brook Hansen.

D.4.1.1.1 Non-standard Motor Orders

Non-standard orders usually involve complex configurations. These configurations are the responsibility of the relevant manufacturing site. Currently, drafts order are forwarded to the manufacturing site from sales branches. The manufacturing site order specialist completes the following tasks:

1. Mechanically configures the product using First Level Assembly (FLA, an in-house developed configuration tool - IBM AS/400 RPG).
2. Electrically configures the product using various PC base tools and manual methods.

Following this, the configured order is entered on the computer system through the order processing unit. The FLA/order entry system interfaces with the manufacturing department.

The product has to be priced before configuration is completed using a set of commercial product selection rules to price and generate quotations using a printed sales manual or standalone PC based quotation system. The product selection rules are effectively a subset of the mechanical and electrical configuration rules, but are currently maintained separately. The various steps involved in processing non-standard orders and the amount of interpretation involved at the various stages have increased the order administration lead-time and increased the scope of errors. Therefore, the development of OASIS, a front end client server configuration tool was introduced to handle the complex process of motor design and specification.

D.4.1.1.2 Standard Order Entry

The Standard Order Entry is used to order motors that are listed in the sales catalogue. These motors are normally stocked and already have defined mechanical and electrical specifications. Stock modifications can also be entered by this method. Sales Branches have access to the Standard Order Entry, but with a restricted range of facilities. The Quick Order Entry is a substitute for Standard Order Entry for use by the Sales Branches. For straightforward orders, an order can be entered using a single screen.

D.4.1.1.3 FLA (First Level Assembly System)

FLA is a computer-based system which uses the motor frame size and features requested by the customer in the form of approved phrases. These are cross-referenced with the basic frame information to choose the correct parts which meet the features required. This new reference (the FLA code) is then linked to the actual computer part number/s for each component making up the assembly. The description of each component printed on documentation is linked to the computer part number on the main database. The description will always be the same and these are being changed to comply with the "Glossary of Terms" which is approved terminology. A further advantage, as more cross reference details are built up, is that orders will pass through order processing faster. The mechanical specifying operation will become a 'maintenance' and 'create a first time combinations' function rather than handling every order every day; for example, a basic frame for a motor is set up as foot mounted with standard end-shields. If the customer's requirement was for a flange with feet motor, the

FLA system would delete the standard drive end-shield and add a flange end-shield giving a correct bill of materials for that specific motor.

D.4.1.1.4 OASIS System

OASIS handles about 80% of orders processing them in a one step operation; the balance requiring engineering or design input will be automatically referred to the relevant department. OASIS aims to provide a consistent order entry system across all the group sales points. Electrical and mechanical configuration is an integral part of each site's manufacturing system; therefore in practice the system had to interface with various flavours of site FLA configurations. For example, Maintenance of pricing and discounts will be the responsibility of central marketing function, while maintenance of product configuration rules will be the responsibility of the engineering and design team. Customer and customer/product specific information may be maintained centrally by the sales team and/or by each manufacturing site. Customer credit information will be managed centrally. All this information must be updated dynamically and made available to the OASIS system as orders are placed on the manufacturing sites from any sales outlet.

D.4.1.1.5 Predicting Dispatch Lead Times

The sales manual is used to estimate despatch times from the date of order draft. It involves different tables which are used to predict lead times. These tables have been developed by the company to help order processor in this task. The motor frame size is used as a base to calculate despatch time. Customer specifications have been classified into several categories based upon the part that needs special features. Extra time is added to standard time based upon a feature table. Table D.1 illustrates a sample of despatch times from date of order draft.

Frame size		Aluminium cage				Cast Iron cage		Wound Rotor		Ex N EEx e		EEx d/ PTB	
		BC		ED		BC		BC		BC		BC	
		Ds	Ws	Ds	Ws	Ds	Ws	Ds	Ws	Ds	Ws	Ds	Ws
100	3ph	25	5	20	4	20	4			30	6	30	6
	1ph	30	6	20	4	30	6						
132	3ph	40	4	20	4	20	4			30	6	30	6
	1ph	20	4	20	4	30	6						
132	3ph	30	6	20	4	20	4	40	8	30	6	30	6
	1ph	20	4	20	4	30	6						

Table D.1. Despatch times from date of order draft.

D.4.1.1.6 Order Volumes

Each order line on a customer purchase order creates a separate Brook Hansen order. Although the quantities of motors ordered per day have varied over the years, the volume of orders has stayed reasonably constant. The following orders quantities relate to Huddersfield and Honley plants combined:

Total Orders per day	420	
Non-standard Orders per day	140	
Standard Orders per day	280	
Standard Orders:		
Entered by branches		240
Entered by Central Administrative Department		40
Non-standard Orders:		
Entered through Non-Standard Order Entry		20
Entered by FLA		120

D.4.1.2 Engineering and Design

The engineering function is based upon the Huddersfield site and provides engineering services to both the Huddersfield and Honley sites. The function comprises four roles; electrical design, mechanical design, production engineering, and product research and development.

Electrical Design: Customer orders are received from order processing including time and date of orders. The time and date are recorded on a small card attached to the customer order. When the order is complete with the design, it is returned to order process and the time and date are recorded on the card attached to the order. At the same time, the necessary test instructions are specified, particularly for the non-stock and make to order business. The main tasks of electric design are: organise and undertake various tests; produce/maintain internal and customer documentation; monitor test process and data produce documentation, specify new design/improve existing designs and specify electric materials e.g. insulation systems.

Mechanical Design. The order and associated documents such as customer drawings, are hand delivered to specific in-tray(s) in DSN. Certain designers are delegated to ensure the contents of the tray are processed within time scales. The order draft supplement form QA49 (Figure D.7) is used to record the necessary specification and after copying this and associated documents, the original package is placed in a specific out tray from where it is collected by order processing. The copies are retained by DSN because drawing requirements will have been anticipated and drawing numbers allocated; these retained copies become the "work to do" file which is held in a second specific in-tray in DSN. A file of customer drawings is held in DSN and maintained via a control system on the "AS400".

For monitoring purposes each job is booked in and out on another form (Figure D.8) and for prioritising, each job is entered on a white board and thereafter, processed in turn by DSN personnel who append their initials.

If at the design stage a modification is found to be required and tickets have been printed, then a specific form is issued to order processing and any affected orders already in the chain are identified and reviewed.

New special components or modifications to components from an outside supplier are ordered on purchasing via "AS400" and drawing/s issued if relevant.

Production Engineering. The role of engineering and design department in production can be summarised in five points:

1. CNC programming
2. Improve/increase capacity (existing & new equipment / plant / fixtures / layouts)
3. Tooling design / manufacture
4. Develop production method
5. Commission new equipment

Product Research and Development. This task can be carried out by engineering and design department through five items:

1. Collate and disseminate research results
2. Influence standard body

3. Perform tests
4. Undertake research
5. Document external research results

DATE MOTOR NO.	DATE DRAFTED	DRG DUE DATE	DRAFTED BY ISSUED TO (C.A.D)
JOB COMPLETED BY:-	DATE		CHEKED BY:-
SHAFT DRG NO.	SHAFT DESCRIPTION		FRAME SIZE
DRIVE END CUSTOMER DRG			
NON-DRIVE END CUST. DRG		ESTIMATED LEGTH	
DRIVE END FACE OF FLANGE TO 1ST/2ND 3RD SHOULDER/END OF SHAFT			
NON-DRIVE END FACE OF FLANGE TO 1ST/2ND 3RD SHOULDER/END OF SHAFT			
D.E. KEY SIZE ( ): SQUARE/ROUND SQUAR/ROUND ENDED		N.D.E KEY SIZE ( ): ENDED	
DRIVE END DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
N.D.E DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
YOKE DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
TERMINAL BOX DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
FAN COVER DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
ADAPTOR PLATE DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
FAN DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
INSIDE CAP DRG NO	DESCRIPTION	CAN BE OBTAINED FROM	
DRIVE END DRG REFF	DESCRIPTION		
N.D.E DRG REFF	DESCRIPTION		
ANY OTHER COMMENTS OR DETAILS TO BE INCLUDED ON MOTOR TICKET			

Figure D.7. Order draft supplement form

INTERNAL ALTERATION TO ORDER

BROOK CROMPTON

TO HUD/CAD/TG - SK
FROM UD/CDO/
ORGINATOR

DATE

COPIES TO FILE

A.T.O: RECORD ONLY   ALL PAPERWORK   NEW TICKET

CUSTOMER NAME   MOTOR NO.

ALTERATION

Figure D.8. Alterations to order form (QA401)

D.4.1.3 Production Planning and Control

Production planning and control in Brook Hansen is based upon a core MRP system. This system deals with components to be MRP controlled using the modules tailored using control parameters. Components not MRP controlled are dealt with by alternative means such as order point, time-phased order point, Kanban and 2-bin inventory systems. All elements of MRP system are time-phased allowing users to schedule work and predict what may happen in the future. The system operates with daily time buckets. The Master Production Scheduling (MPR) is used in limited locations, when capacity is accounted for by manipulating planned stocks. The major modules of the MRP system are shown in Figure D.9.

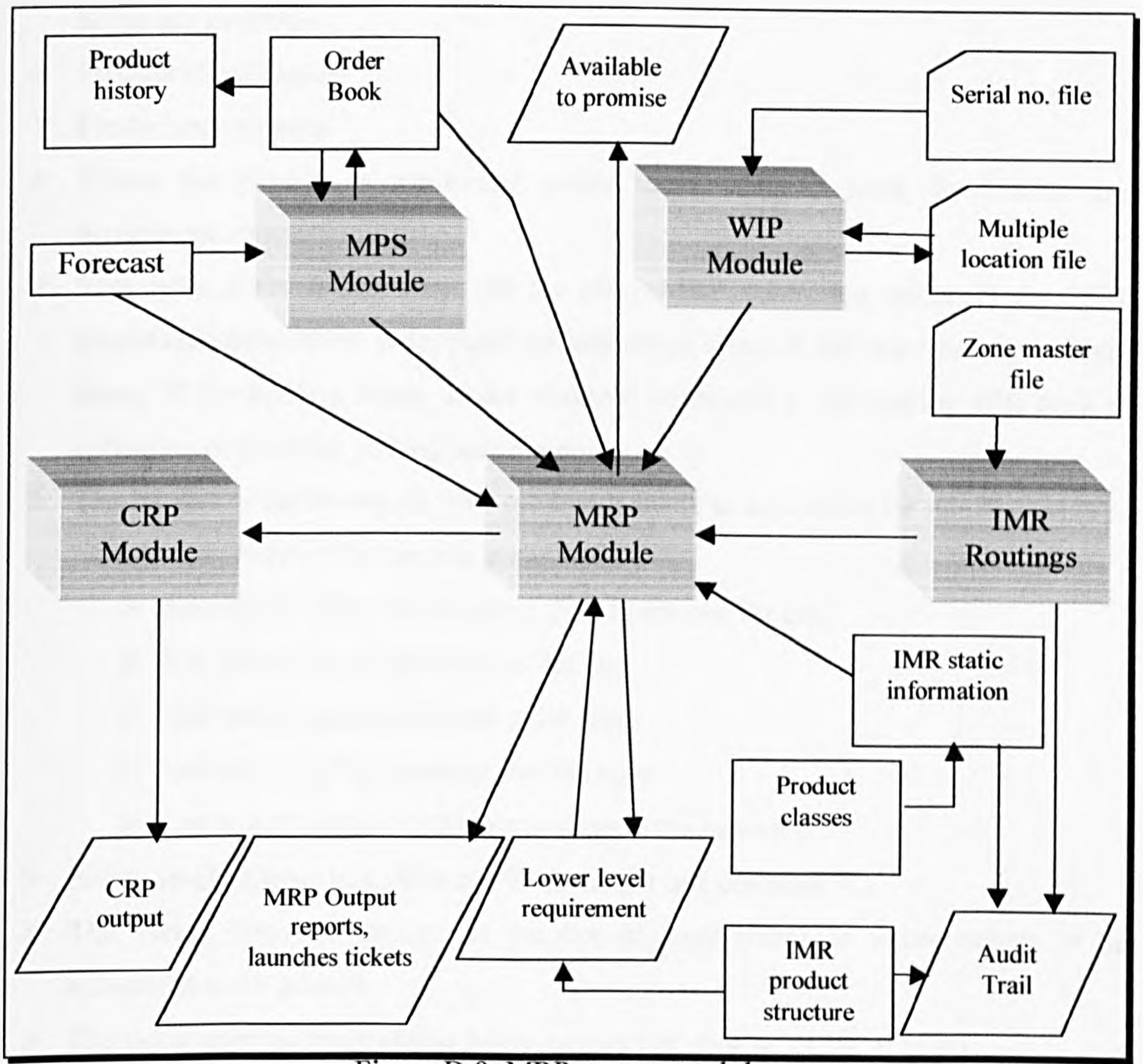


Figure D.9. MRP system modules.

D.4.1.3.1 Item Master Record

The Item Master Record (IMR) is. It is designed to hold manufacturing information on each product. This information is divided into three main categorises, on three separate computer screens:

1. Static information
2. Product structure information
3. Routing and capacity information

Screen (1) of the IMR-static information shows:

- Part number and alternative part number
- Product description, and drawing number

- Level in the product structure. Top level items such as motors are level 1. Up to nine levels are available
- Product classification
- Product sub-classes
- Where the product is made and where the product is used, by factory and department code.
- Item type. **I** for independent, **M** for other MRP controlled items, **N** for MRP purchased items, **O** for order point manufactured items, **P** for order point purchased items, **K** for Kanban items, **H** for phantom items and **C** for costing bills such as collection of parts for costing purpose only.
- The lot size or batch quantity this product is made in and which lot rule is used. The lot rule defines how that lot size is applied.
  - Lot rule 1: apply no lot sizing ,(i.e. Make one for one).
  - Lot rule 2: apply minimum of lot size.
  - Lot rule 3: apply multiples of lot size.
  - Lot rule 4: period cover of lot size days.
  - Lot rule 5: as lot rule 2 but round up to the nearest 5.
- Safety stock information. This can be fixed **(F)** or calculated **(C)**.
- The Ticket Time Fence i.e. the number of days worth of future tickets to be authorised to be printed.
- The ticket printing method (per batch, per unit or none at all -**B, U** or **N**).
- Forecast information.
- The manufacturing time in days. This including all queuing and is thus the average number of days the part actually takes form lunching it to it arriving in stroke.
- The smoothing factor for the Average Weekly Demand (AWD), this can be set between [.00 and .99] and determines how reactive the AWD is to the actual demand, .00 being the least responsive and .99 being the most. For example, if the smoothing factor 0.1: New **AWD = 0.1 x old AWD**.
- Scrap allowance in percent can be entered and MRP will allow for this when reordering parts.
- The status of this record (active **(A)**, locked out from activity **(L)**, or withdrawn **(W)**).
- The Perpetual Physical Inventory (PPI) flag is used to determine what sort of stock monitoring should be applied, A being normal bi-annual count P, being perpetual physical inventory (sometimes known as cycle counting).

- A category, sort number and report group can be entered and to determine how reports or screens will be sequenced.
- The works-loading group is another way of classifying products, primarily for commercial reasons.
- The ABC code is a way of classifying products to determine how they should be controlled. **A** items are high cost and/or difficult to make, down to **C** items which are low cost and/or easy to make.
- Three units of measure may be entered; units for works, count and ordering. Accepted units are:

EA (Each)	MT (Metres)	MM (Millimetres)
IN (Inches)	FT (Feet)	M2 ($Meters^2$)
C2 ($Centimetres^2$)	SH (Sheet)	LT (Litres)
KG (Kilograms)	TN (Tonne)	GM (Grams)
PR (Pair)	PK (Pack)	PI (Piece)
BR (Bar)	LH (Length)	

- There are two principle stock fields:-
  - At the last MRP run (usually overnight)
  - The current stock

  These two fields will only be different in interactive systems, or if the stock has been manually amended for any reason. Figure D.10 illustrates the IMR static information screen.

ITEM MASTER FILE

Mode – Update			
Part No.	99 240408	Alt Pn	
Description	D100LA 4EC	Drawing No.	
Level In Prod. Structure	1	Product Class Code	199162
Price	146.50	Class Des D100L CELL ALUM 3 PH	
Where Made	1C1	Sub Class	
Where Used	105	Item Type	I
Fixed Stock Flag (F Or C)	F	Lot Size	1
Safety Stock	100	Lot Rule	1
Stock Adj. Factor (%)	100.0	Forecast to be Used (Y or N)	Y
Tke. Per Batch / Unit / None	U	Forecast Time Fence	2
Ticket Time Fence	3	Smoothing Factor (.00 - .99)	.10
Manufacturing Time (Days)	2	Scrap Allowance (%)	
Transient Part (Y Or N)	N	PPI Flag (P or A)	A
Status Flag (A, L Or W)	A	Stock Check Code	1
Withdrawal Date	0/00/00	Last Count Date	91/10/13
Replacement Part Number		Works Loading Group 106	
Cat. / Sort No. / Rprt. Grp	01 0048 01		
Ticket Colour	0	ABC Code (A, B, or C)	A
Unit Of Measure –Works		-Count Order	
Conv. Factor Count/Works	.000	Order/Works	.000
Mech. Code	10 100 04 003	Super Stock S	S
Physical Stock	60.00	Physical Stock at last MRP	45.00
CF3 Return/ No	Update, CF23 Delete ,	Enter to Continue,	CF2 Ignore Drg No
Error			

Figure D.10. Screen(1)- IMR static information.

Screen (2) –product structure information:

This screen is used to enter product structure information. The product structure is built up by entering for each part that links to its component parts only. This section also allows the entry of this information by keying in the component part, the quantity required, the unit of measure (UM) and the zone used. Figure D.11 illustrates screen (2) of the IMR.

ITEM MASTER FILE

Product Structure Data

Mode – Update

Part No. 99 240408 Description D100LA 4eC

(A-amend) Alternate Part No.

(D-delete) Level In Product Structure 1

(V-view) Components Used At This Level

Line	Part No.		Qty.Reqd.	UM	Zone used
01	10241210	D100La 4EC	1	EA	11508
02	14240157	D100L Prof KWY	1	EA	11508
03	25249000	D100L M.P.	1	EA	11508
....	....	....	...	....	....
....	....	....	...	....	....
....	....	....	...	....	....
....	....	....	...	....	....
....	....	....	...	....	....
....	....	....	...	....	....

F3 –exit (no update) F2-prevous F4-Zone enquiry F6-Add entry

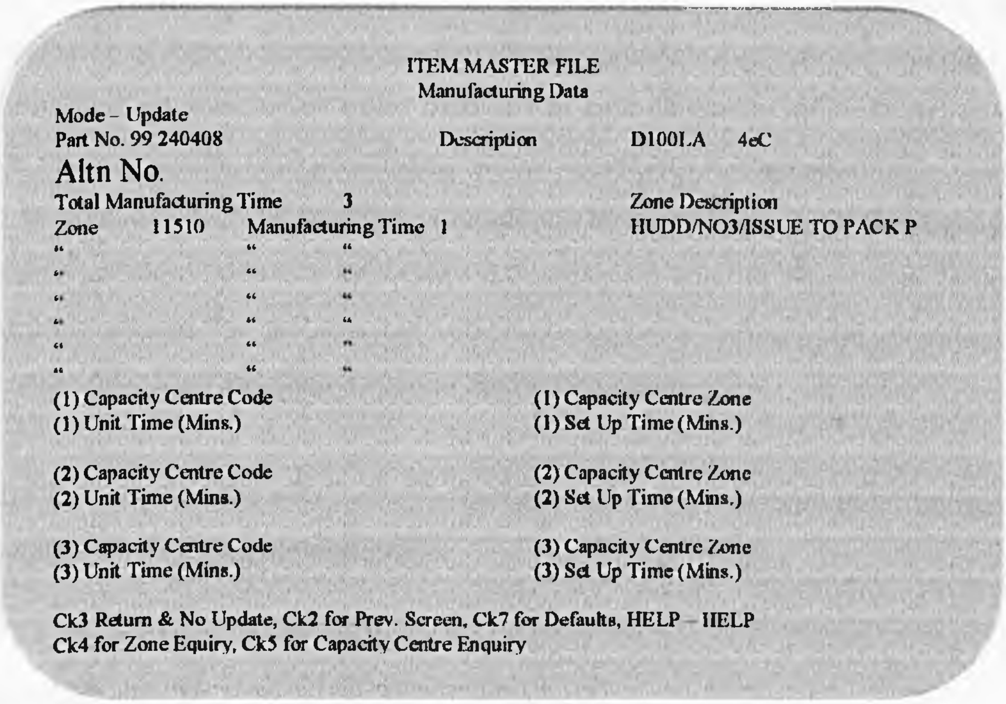
HELP

Figure D.11. Screen (2)-product structure information.

Screen (3), routing and capacity centre information:

This section contains information about the products routing i.e. those zones through which it passes. The zone code must exist in the zone master file and should be entered from top to bottom in the sequence the product passes through. The manufacturing time must also be entered. The total manufacturing time must add up to the value that entered on the first screen.

The creation of Item Masters is assisted by copy facilities. A new IMR can be created by defaulting product class information or by copying another IMR. In the latter case, all data, with the exception of description and safety stock, is copied including the product structure and routing. Figure D.12 illustrates this screen.



ITEM MASTER FILE
Manufacturing Data
Mode – Update
Part No. 99 240408 Description D100LA 4eC
Altn No.
Total Manufacturing Time 3 Zone Description
Zone 11510 Manufacturing Time 1 HUDD/NO3/ISSUE TO PACK P
" " "
" " "
" " "
" " "
" " "
" " "
(1) Capacity Centre Code (1) Capacity Centre Zone
(1) Unit Time (Mins.) (1) Set Up Time (Mins.)
(2) Capacity Centre Code (2) Capacity Centre Zone
(2) Unit Time (Mins.) (2) Set Up Time (Mins.)
(3) Capacity Centre Code (3) Capacity Centre Zone
(3) Unit Time (Mins.) (3) Set Up Time (Mins.)
Ck3 Return & No Update, Ck2 for Prev. Screen, Ck7 for Defaults, HELP – HELP
Ck4 for Zone Equiry, Ck5 for Capacity Centre Enquiry

Figure D.12. Screen (3)-routing and capacity centre infromation.

D.4.1.3.2 Product Class Record

The product class record groups together a number of Item Master records (IMR's) to enable various operations on item masters to be performed more efficiently. The creation of a new IMR will default product class information, if specified. Global changes on a whole class of IMR's can be achieved by changing the class record and initiating a transfer into all those IMR's. Statistics and analysis can be produced to a range of IMR's with the same class. This can be done by holding the confidence limit of supply. The product class determines the scale of the safety stock for each item in the class. Changing this limit will affect all IMR's with this class. The product class record holds similar information to that of the IMR.

D.4.1.3.3 Routings

A product route is described in terms of zones, e.g. assembly, and casting and machine winding. It is limited to 8 per part or level of the BOM. In addition, prior to the first zone is an assumed "Ticket on hand" zone, to show the outstanding requirement not yet in production; and after the last zone is an assumed "In Stock" zone which, in fact, is the physical stock of the finished part. These two zones are, therefore, not explicitly stated in the routing. Each zone requires the length of time (in days) the item will spend in it. The same of days in the route must equal the manufacturing time for this item. The number of days per zone has an upper limit of 9 (a zone should be split into two or more if this is exceeded, otherwise progressing is too vague) and a lower limit of 1 (the zone should be abandoned if less than this is required. Otherwise, data collection becomes a burden). In addition, there is a total limit of 15 days per level in the product structure. Transactions entered for the part are validated using the routing, and quantities are added to the new zone and deducted from the previous zone.

Routings are shown in the item mater, on screen 3. Each zone entered in a routing must have a valid entry in the zone master file.

D.4.1.3.4 Work-In Progress (WIP) Module

After transactions have been validated against the routing, they are used to move WIP to the next zone. In fact this is the only way WIP can move between zones in normal circumstances. Within a zone of more than 1 day, the WIP is moved along by the system day by day until a zone boundary is reached where it will stop until a transaction is entered. Thus the system automatically estimates where the WIP is even when transactions are not being entered. This information is used by 'Available to Promise' systems which need to predict the completion date of WIP. They can then advise when finished products will be available for sale or components will be available for use by the next level up.

D.4.1.3.5 Product Structure

The product structure is the description in the system of the parts included in the product (the motor). This enables MRP to order the relevant parts to put the products together. A product structure is also known as a Bill Of Material (BOM), but it is important to realise that this product structure also conveys the sequence that the parts are put together; it is not just simply a list of parts. It is not intended in which all parts of a product will be acted on by MRP; this would be waste of computer resource and the results would be impractical to action. Instead, many easily made small or cheap parts will be "Order Point" or "Stores" controlled. Thus only major parts will normally be included in the product structure. To value a product, all the material must be tagged to it – this is the function of Item Type C (Costing Bill). This type in a structure will not be treated by MRP, but will include all other items in the product at that level for costing purposes. The product structure is built up level by level. In other words, only those components used at this level are included at this level. Thus each item master screen (2) shows only the next level of parts; to find further information the next item master must be consulted. Thus the item master product structure screen only shows the linkages between this item and its components. The screen also shows the parts description unless the part has an alternative part number.

A product structure display option is available and this will show all the MRP parts in the product right down to the bottom of structure. Alternatively, this option may also be used to show all component of which the item is a component part ("where used"). Along with structure information the quantity required, a unit of measure (UM) and a zone is used. The quantity is the number of the lower level part needed at this point and will often be 1 but could be 2, 3 or even for example 0.64 kilograms of steel. The zone used is the zone in the item's routing that the component is required at; when this item (the parent) passes into this zone, the system will down date the stock of the component by the quantity required. Moving the point usage forwards or backward will result in double or no usage for that particular part. Thus, action must be taken to correct this by subtracting or adding to the lower level part stocks, if this is changed. The finished product is at the top of the structure and purchased items, where MRP controlled, will be at the bottom. Figure D.13 shows a sample BOM for aluminium motors.

LEVEL				PART NO.	DESCRIPTION	QUANTITY USED	UNIT	ITEM TIPE
1				10245001	WD100LB 4R	1	EACH	MRP
	2			11245101	WD100L 4P	1	EACH	MRP
		3		13245101	152.4 X 92.075 X 36W	0.0121	METER	MRP
			4	33164014	MK2 SUPERSTEEL 400 0.5MM AS	0.109	TON	PURCHASED
			4	33164100	SCRAP LAMINATION STEEL (KGS)	0.049	TON	PURCHASED
	2			33067902	100-AL-3-B-24"LEAD	1	EACH	ORDER
		3		33066349	1MM EXAR 350 CABLE (32/.2) RED	0.61	METER	PURCHASED
	2			33091105	0.80MM COPPER WIRE KGS	3.12	KG	PURCHASED
	2			85240010	WD100 WDG INSUL MTLS - CLASS F	1	EACH	ORDER
		3		33070185	DTEX 1100/1X3 POLYESTER THREAD	0.001	ROLS	PURCHASED
		3		33072003	1.5MM TERYLENE SLEEVING YELLOW	1	METER	PURCHASED
		3		33072284	4MM POLYGLASS SLEEVING NATURAL	0.6	METER	PURCHASED
		3		33088451	2/7 NOMEX/MELINEX 914M/M	0.131	METER	ORDER
			4	33088044	PLYESTER FILM .0075 X36	0.25	KG	PURCHASED
		3		33088452	2/10 NOMEX/MELINEX 914M/M	0.046	METER	ORDER
			4	33088234	.010"MYLAR 250MIC.	0.323	KG	PURCHASED
		3		33088503	2-3-2 NOMEX 18"X24"SHEETS	0.077	EACH	PURCHASED
1				14240353	WD100 SRD-CELL	1	EACH	MRP
	2			12246101	WD100LR 4/6	1	EACH	MRP
		3		13246101	92.075 X 34 RT32	0.121	METER	MRP
			4	33164014	MKS SUPERSTEEL 400 0.5MM AS	0.06	TON	PURCHASED
			4	33164100	SCRAP LAMINATION STEEL (KGS)	0.027	TON	PURCHASED
		3		33099000	99.6 ALUMINIUM INGOTS	0.765	KG	PURCHASED
	2			30245500	WD100 STD-CELL	1	EACH	MRP
		3		33163118	060A25 34.5 DIA.	2.594	KG	PURCHASED
1				25245000	WD100L STATOR FRAME	1	EACH	KANBAN
	2			33099004	LM 24 ALUM INGOTS	2.89	KG	PURCHASED
1				27245000	WD100 ALUM D.E.	1	EACH	KANBAN
	2			33011325	WD100/112 BEARING INSERT	1	EACH	PURCHASED
	2			33099004	LM 24 ALUM INGOTS	0.524	KG	PURCHASED
1				27245003	WD100 ALUM N.D.E.	1	EACH	KANBAN
	2			33099004	LM 24 ALUM INGOTS	0.607	KG	PURCHASED
1				33012004	DPSM 30-42-7 OILSEALS	1	EACH	PURCHASED
1				33012282	DPSM 25-37-7 OILSEAL WITHOUT	1	EACH	PURCHASED
1				33014592	6206 2Z C3 BALL BEARING	1	EACH	PURCHASED
1				33014632	6205 2Z C3 HT22 VU082 BEARING	1	EACH	PURCHASED
1				33033199	WD100L BOX/FRAME GASKET	1	EACH	PURCHASED
1				33033200	WD100L BOX/LID GASKET	1	EACH	PURCHASED
1				33042172	THINNERS FOR ALKYD PAINT	0.007	LETER	PURCHASED
1				33042369	BECKERS WASSER BLUE A.D.ENAMEL	0.063	LETER	PURCHASED
1				33055580	WD100L D/C T.BOX 20MM.	1	EACH	ORDER
	2			33099004	AM 24 ALUM INGOTS	0.2	KG	PURCHASED
1				33055847	WD100L D/C TERMINAL BOX LID	1	EACH	ORDER
	2			33099004	LM 24 ALUM INGOTS	0.185	KG	PURCHASED
1				33058939	D.100.T/BOARD.6.PIN.SUB/ASSY.	1	EACH	ORDER
	2			33006251	4MM HEX FULL NUT	6	EACH	PURCHASED
	2			33007214	M4 INT TOOTH LOCK WASHER	6	EACH	PURCHASED
	2			33007297	CLANP WASHER	6	EACH	PURCHASED
	2			33028056	63/100 TERMINAL PINS	6	EACH	PURCHASED
	2			33058928	63 100 FRAME T/BOARDS ITEM B	1	EACH	PURCHASED
1				33061508	WD100L 4P.POLY.FAN F2 24MM.BOR	1	EACH	PURCHASED
1				33063967	WD100L POLY.FAN COVER BLACK	1	EACH	PURCHASED
1				33117074	WD100 BOLT ON FEET	2	EACH	PURCHASED
1				33179834	REF.1167 NAMPLATE	1	EACH	PURCHASED
1				85241010	WD100L FASTENER	1	EACH	ORDER
	2			33003030	M12 DYNAMO EYEBOLT	1	EACH	PURCHASED
	2			33004261	M6X30MM.HEX.WIRTH WASHER HEAD	8	EACH	PURCHASED
	2			33004268	M8X20 HEX.WITH WASHER COREFLEX	4	EACH	PURCHASED
	2			33005353	M4X20 TAPTITE POZ PAN HD SCREW	1	EACH	PURCHASED
	2			33005365	NO.6 X.250 PAN HD SELF TAP SCW	2	EACH	PURCHASED
	2			33005377	M5XM16 TAPTITE POZI SCREWS	8	EACH	PURCHASED
	2			33005412	M6XM16 PAN HD.POZI. T/TITE SCREW	2	EACH	PURCHASED
	2			33005506	M5X8MM.POZI.PAN HD.TAPTITE	3	EACH	PURCHASED
	2			33005253	M6 EARTH CLAMP WASHER	2	EACH	PURCHASED
	2			33007304	BELLEVILLE WASHER DIN 6796	4	EACH	PURCHASED
	2			33009358	DOUBLE RND ENDED PROF KEY	1	EACH	PURCHASED
	2			33013041	D100/112 PER LOAD WASHER	1	EACH	PURCHASED
	2			33029019	25MM. SEEGAR CIRCLIP	1	EACH	PURCHASED
	2			33029057	24MM. SEEGAR CIRCLIP	1	EACH	PURCHASED
	2			33029071	BEARING CIRCLIP N1302/0206	1	EACH	PURCHASED
	2			33037089	90/100 W100/W112 M/S FLANGER	1	EACH	PURCHASED
	2			33058628	D100L DELTA STRIPS IN PACKETS	1	EACH	ORDER
		3		33006251	4MM HEX FULL NUT	6	EACH	PURCHASED
		3		33034046	2.25"X3.00"SEAL AGAIN BAG	1	EACH	PURCHASED
		3		33058991	100L SHORTING LINK	3	EACH	PURCHASED

Figure D.13.AL motor structure.

D.4.1.3.6 Material Requirement Planning (MRP)

The MRP module takes known requirements, forecast requirements and safety stock, and plans how many items to make on a particular day. MRP will be run, at the end of each working day, so that the position each morning is fully up to date and accounts for all new events from the previous day, e.g. new orders, transactions made, scrap and changes to system parameters. Thus it can be seen that the launch list or 'plan' produced by MRP might change every day.

The MRP performs its function based upon the following inputs: -

- Daily requirements

  For end products. This is from order book, it takes into account wagon run data and order type data (precise, stock modification), where appropriate to determine the works due date for the product.

  For components. This is the launch requirement from the next level up plus independent demand (from non-standard motors for example). These are offset earlier by one day from their higher-level launch date to ensure availability on that date.
- Forecast

  This is the Average Weekly Demand (AWD) for the part, which is divided down into an average daily demand.
- Safety Stock
  - This is held in each IMR, and is the stock which MRP will attempt to hold in addition to the above requirements and forecast. Efforts are made to keep safety stock to minimum because of its associated costs. These costs include handling cost, space/warehousing cost, the cost of damage, etc.
- Current physical stock, quantity allocated and WIP
- Lot size and lot sizing rules.
- Shop Calendar

  This removes weekends and holidays, and allows MRP to use working days only.
- Product Structure

  This allows MRP to project launch requirements for items into the next level down in their structure.
- Manufacturing lead time and component offsets (in number of days).

The MRP logic takes the above inputs and effectively calculates when the stock balance will fall below the required safety stock in the future. It then schedules work to be launched "overdue"; despite the fact that this is obviously impossible, these are still shown overdue, to prioritise them over other items. Thus the MRP system output is in the form of launch lists, which detail which items should be made on which days.

Independent demand is passed to MRP when a non-standard higher level part uses an MRP part. For example, a non-standard motor will use some standard parts such as end-shields and wound pack. The quantities and required due dates of these parts will be passed to MRP so that they may be included in the total requirements of those parts, and enable correct prioritisation to be achieved. Similarly, spares, repairs and other independent requirements are passed into the MRP system to ensure that it includes all requirements.

D.4.1.3.7 Capacity Requirement Planning (CRP)

The CRP modules are geared around the capacity centres held in the IMR on screen-3. There are up to three capacity centres allowed per part in the product structure; each one must have a valid entry in the capacity centre Master File and is associated with a zone in the works where the bottleneck is sited. The unit time for each part passing through the bottleneck in this centre will be assumed to determine the total load for the centre. This load will then be compared to the defined capacity for the centre to produce a chart showing overload or under-load for future weeks, enabling action to be taken in advance of the problem.

D.4.1.3.8 Shop Floor Control

Shop floor control involves four main tasks:

1. Issuing work to production
2. WIP
3. Data collection
4. Works documents

1. **Issuing Work to Production.** This revolves around the use of launch reports or screen to advise which items should be launched into production. At this stage of development these are presented with the launch dates as priority columns, rather than as "work-to" lists, to give departments some flexibility production.
2. **WIP.** Once launched, parts are tracked from transactions entered by displaying them in the production zones. These displays may be interactively updated in which case they show up-to-the minute information, or they may be updated in batch mode overnight, in which case the information will be correct only first thing in morning. WIP is tracked and moved in such a way that the system knows when it is completed. When work is launched, all parts in the product structure are allocated to that job. This enables the system to show the remaining free stock of the parts to determine whether more work can be issued against them. As the job passes transaction points, the system checks parts that are used, and down-dates the stock and allocation of the component automatically.
3. **Data Collection.** The principle adopted here is to validate data entered as much as is possible. This is done by:
   i. Entering both the part number and a serial/works reference number. The system can then cross-check these to ensure they are linked, it can also check that this transaction/part number/serial number combination has not been entered before or in the wrong transaction sequence; or
   ii. Entering a serial/works reference number by barcodes which have inherent self-checking facilities.
4. **Works Documents.** Following the MRP reorder decision, the system will print works documents. Whether these are printed one-for-one, as a batch, or not at all depends on the ticket printing flag in the IMR being 'U', 'B', or 'N'. Works documents are supplied with a barcode or tear-off portions where appropriate. Batch-card type feedback may not use either of these methods; if this is the case, the batch card should preferably pass by transaction screens on its route, at which points operators should key in their actions. Where this is not possible, or where no tickets at all are printed, operator-completed job sheets will have to be used for feedback. This, however, is not desirable as their accuracy will be suspect, leading to errors in WIP.

D.4.1.3.9 Product History

A history record is kept for each MRP part for several reasons:

- For further analysis of order pattern, manually or by the computer, for checking seasonality and trends;
- For calculating the AWD, for use as a forecast;
- For calculating the most cost affective and appropriate safety stock.

The independent demand for every item is stored in the product history file, at this time for 52 weeks, though this could be cut to 12 weeks for some items. The dependent demand is determined from the dependent AWD. Both dependent and independent AWD are recalculated weekly using the method described in the IMR. The total AWD is used for forecast purposes for an item if its forecast flag is set to 'Y'.

The last 12 weeks of independent demand are used to recalculate the safety stock weekly. Product history can be copied from an item into its replacement part number, held in the IMR to facilitate changeover from one item to another.

D.4.1.3.10 System Output

Each area under MRP control works to the same methodology even though the output may look different in each case. For example, in some areas this may be report-based, in others screen-based. The main output is in the form of:

1. MRP Standard Item Report;
2. MRP Launch List.

The former gives information about the part, in particular:

- The AWD: this may be split into dependent and independent values so that the user may determine how much demand comes from standard sources and how much from non-standard sources;
- The part number and description;
- The safety stock from the item;
- The manufacturing time for the item: this represents the average time, in days, the item takes, including any queuing time, to reach stock after being launched;

- The WIP situation;
- The requirement situation: the quantities are due to be completed on the days shown. The requirements include all the quantities necessary to enable the next level up to launch all their items shown on their launch list, which includes their stock level and their forecast. They do not include safety stock and forecast at this level.

Thus, the Standard Item Report is an information report which gives information on the status of the part at that time. On the other hand, the second report, the launch list, is an action report. This gives details about the items to be launched at this level by date, in order to meet all known requirements, forecast requirements and safety stock. It also includes any batch quantities and scrap allowances if set.

The launch list, therefore, shows the following items:

- AWD;
- Part numbers and descriptions;
- Possibly information on minor parts made into this item;
- Required launch dates. The spread of these will depend on the space available, but will show some over due days and some forward days with 'TODAY' being placed approximately centrally. Where a wider spread than the space available is required, % days may be 'bucketed' into a week. This means that if all work, up to and including today is launched, all requirements for that item will be satisfied, assuming that they are made in the average manufacturing time set. All requirements include known demand, forecast demand and safety stock. Items launched late (from overdue columns) will cause stock outs; items launched early (from advance columns) will use up parts made for other items;
- The requirement columns are totalled at the end of the report to give a guide as to the workload placed on the section or department.

The above standard report and launch list drive the system. There are many subsidiary reports and screens which are produced to aid the operation of the system, a selection is listed below:

- Standard information list. Details static information about parts: lot size, lot rule, time fence etc.
- Product structure/Where used. Shows bills of materials either down or up the structure.

- Location system. Enables stocks to be carried in more than one place.
- Parts Enquiry. Shows the availability of an item and estimated arrival of new stocks.
- Intermediate priority lists.
- MRP warning messages. This reports circumstances where there may be potential problems. Currently the messages are as follows:

  Safety Stock > 15 * AWD + 5

  Safety Stock < 0.2 * AWD - 2

  Lot Size > 15 * AWD + 10

  Lot Size < 0.2 * AWD - 2

D.4.1.3.11 Production Scheduling

In Brook Hansen Motors, manufacturing scheduling is designed to utilise the company's resources. They take information from both MRP and non-standard order entry to synchronise the winding and component areas. The scheduling system is designed and implemented to reduce unnecessary WIP, allow more efficient use of resources and also reduce bottlenecks at assembly. This system does this in two ways. Firstly, it allows the winding section to create a five-day production plan using capacity planning. Secondly it drives the component areas from this production plan so that they know which orders are due to off and hence which components to make. Working this way means the components will be manufactured to go with the windings in production rather than waiting until assembly to find out which components they may or may not have. The winding schedule is the driving factor behind cell and both winding and component production scheduling, as illustrated in Figure D.14

A production plan can be created from the winding schedule using the planning module to load winding machines. Winding production time can be determined for the selected motors and also changeover time can be predicted. Once the initial production plan has been set up the winding supervisor can then try and launch the days work.

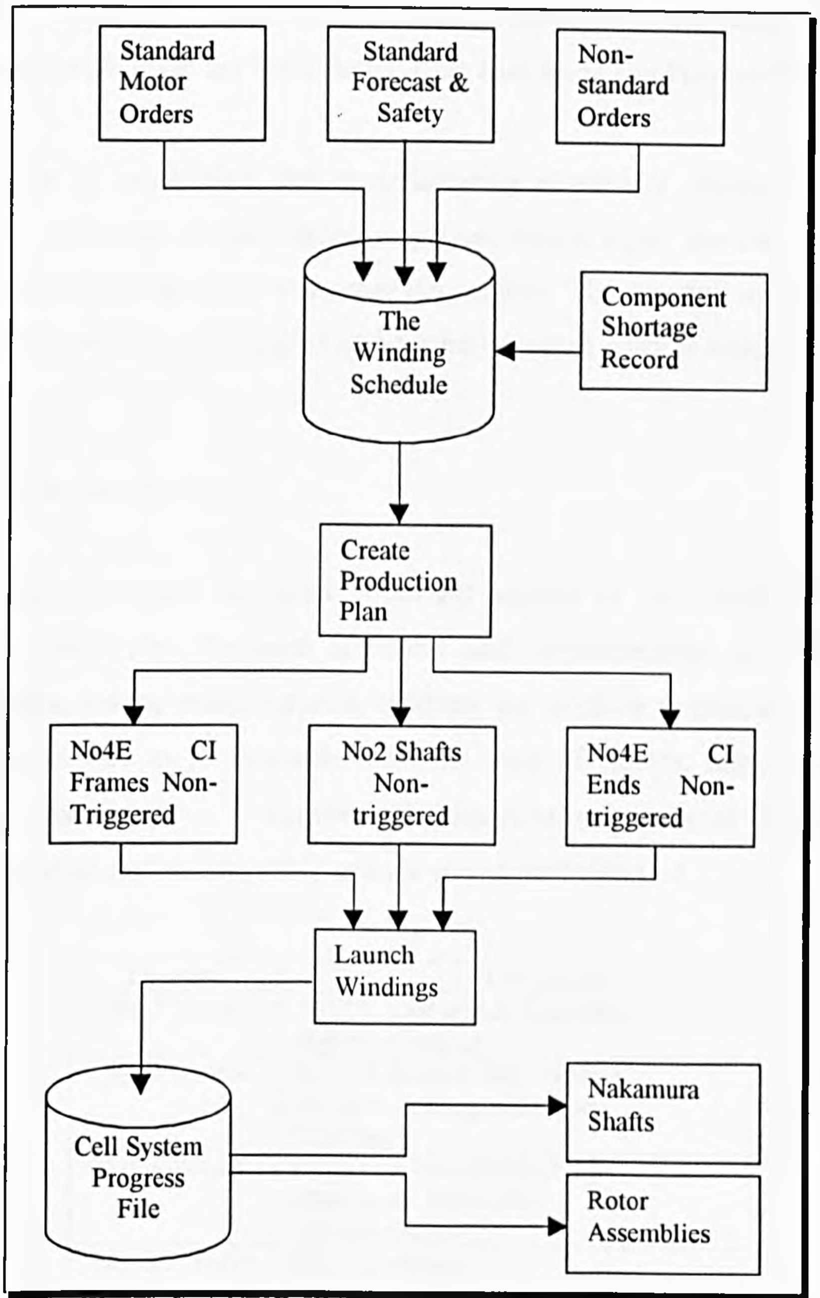


Figure D.14. Winding and its scheduling relationships

Motor components on the cell system can be manufactured at one of two stages. Firstly, there are the triggered components which are made under cell control and triggered after the winding is launched. Secondly, there are those which are made outside cell control and set off per winding launch. The reason for having two stages of manufacture is that components made outside cell control cannot be manufactured within the time it takes the pack to be wound and reach assembly. The only cell made components are Nakamura shafts and rotor assemblies made using these shafts. These items are triggered in to production when the winding is launched, and are thought to be capable of manufacture within the pack lead-time. Although rotor cores still come from outside cell control, their availability is checked on the winding launch and they are assumed to be available at assembly time. There are other four main components that come from outside cell control; these are No 2 shafts, Cast Iron Ends and Frames and Rotor Cores.

In most cases of production, the manufacturing supervisor changes the priority of batches and combines similar orders together, based upon several factors such as machine set-ups, raw materials and order importance. The scheduling problem in Brook Hansen Motors is still very complex and further research work is needed.

D.4.1.4 Manufacturing

Manufacturing represents the most important aspects of any industrial company. It involves the interaction between technical and organisational activities. In Brook Hansen Motors, several manufacturing locations are working to produce different motor components, as well as to assemble different sizes of electric motors. Small electric motors are assembled in a number of manufacturing cells at Huddersfield site. Assembly locations of Huddersfield site are shown in Table D.2.

Location	Description
No. 3 Works	160/180 Cast Iron & Aluminium Slip ring winding
No. 3 Works	160/180 Hand & M/C winding Offshore painting, wire stores Load test
No. 4 Works	D100 Cast Iron DF112 & 113 Flameproof E90-E180 Abbreviated test
No. 8A Works	Slip ring motors Petrol pumps Large motors 200 to 250 frame Premium motors Low canter motors
No. 8 Works	Spares & Repairs
No. 9 Works	Aluminium 100 112 132 M/C & Hand wound, Lead & Insulation preparation 100 & 112 Rotor Assembly Warehouse Stock

Table D.2. Assembly locations in Huddersfield site.

A number of component production departments at both sites feed the assembly areas with the parts required. Component manufacturing locations at Huddersfield site are shown in Table D.3.

Location	Description
No. 9 Works	100 to 180 frame Non-cell machine winding and hand winding 100-132 frame windings 100-112 frame shafts and rotor assemblies
No. 14 Works	Aluminium foundry casting end-shields and stator frames for both Huddersfield and Honley
No. 4E Works	Cast Iron end-shield and stator frame machining
No. 1 Works	Cast Iron special machining Rotor assembly production Shaft machining Rotor core die casting Plastic department

Table D.3. Component production departments at Huddersfield site

Four assembly lines work for aluminium motors. These assembly lines are located in No. 3 and No. 9 Works. No. 3 Works involves assembly works of aluminium and cast iron motors 160/180. No. 9 works involves three manufacturing cells working motors 100, 112 and 132. Figure D.15 illustrates the relationships between the assembly and component production.

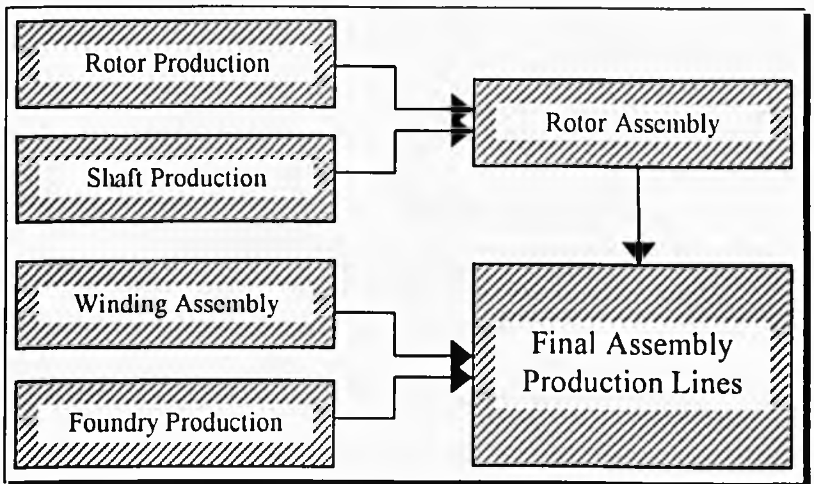


Figure D.15. General relationships between component and assembly production

D.4.1.4.1 Motor Rotor Production

Motor rotor production at Huddersfield site is carried out at area '2' as shown in Figure D.3. This production area involves six casting stations; five are operated manually and one is operated automatically using an industrial robot. The manual casting stations are

1, 2, 3, 4, and 5, and the automatic station is '6' as illustrated in Figure D.16. Figures D.17 and D.18 illustrates the manual and automated rotor stations respectively. Varieties of steel lamination boxes are stocked in the same area near casting stations in addition to aluminium ingots stock. Small batches of finished rotors may be stocked for a short time before passing them to the rotor assembly area. Figure D.19 shows finished rotors ready to be transferred to the rotor assembly plant. These rotor stations feed all the different assembly lines of motor productions at Huddersfield site.

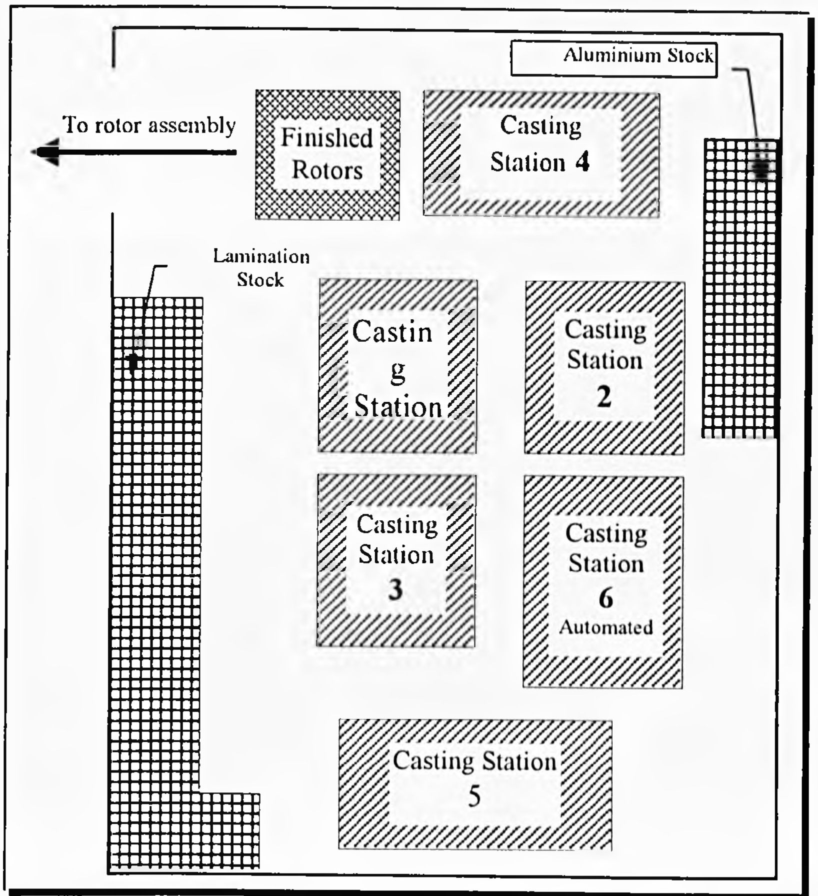


Figure D.16. Motor Rotor Production Department

Each station is used to produce one or more types of motor rotors as:

1. Station-1 is used to produce rotors for motors 100 and 112;
2. Station-2 is used to produce rotors for motors 132;
3. Station-3 is used to produce rotors for motors 132 and 160;
4. Station-4 is used to produce rotors for motors 160;
5. Station-5 is used to produce rotors for motors 160, 180 and C225;
6. Station-6 is used to produce rotors for motors 100, 112 and 132.

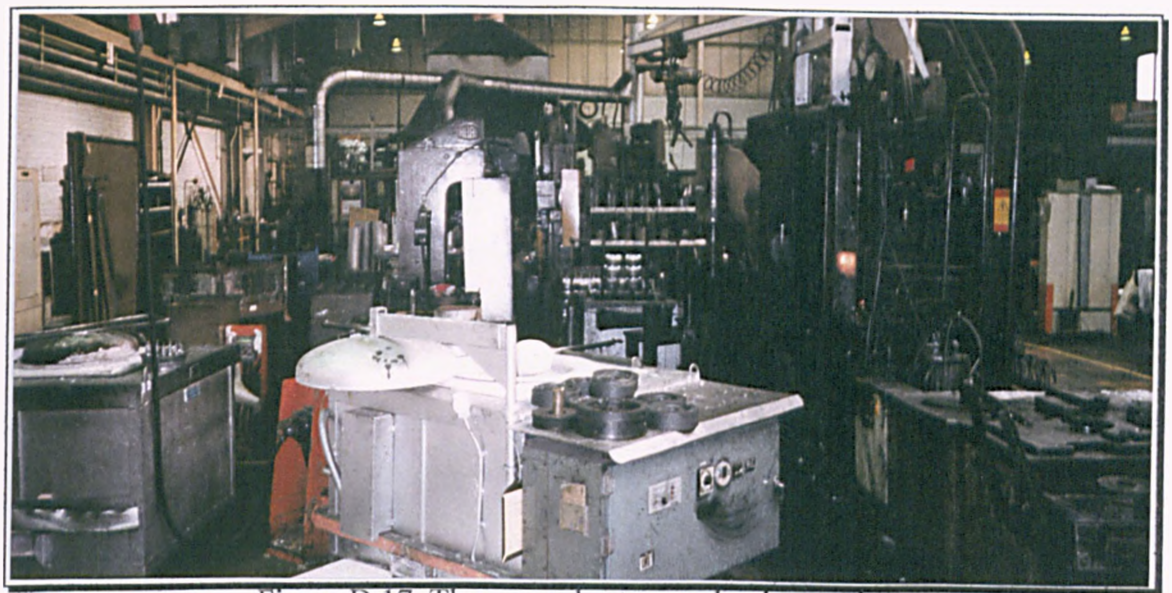


Figure D.17. The manual rotor production station.

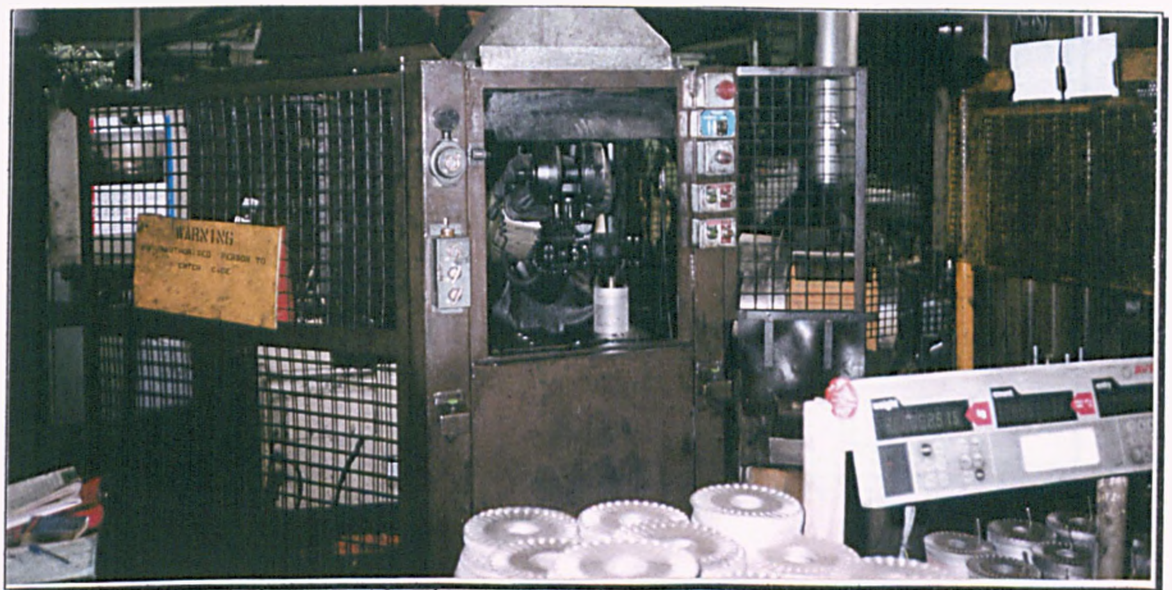


Figure D.18. The automated rotor production station.

Figure D.19. Fished motor rotors.

D.4.1.4.2 Motor Shaft Production

The Shaft department receives orders placed to the company in the form of special and stock order requirements. An electric motor shaft is one of the most common components produced under high production variety owing to different customer specifications related to shaft design and manufacturing processes. The special features associated with the motor shafts are produced using instruction given on batch card and design drawings. Figure D.20 illustrates the main production unit for motor shafts.

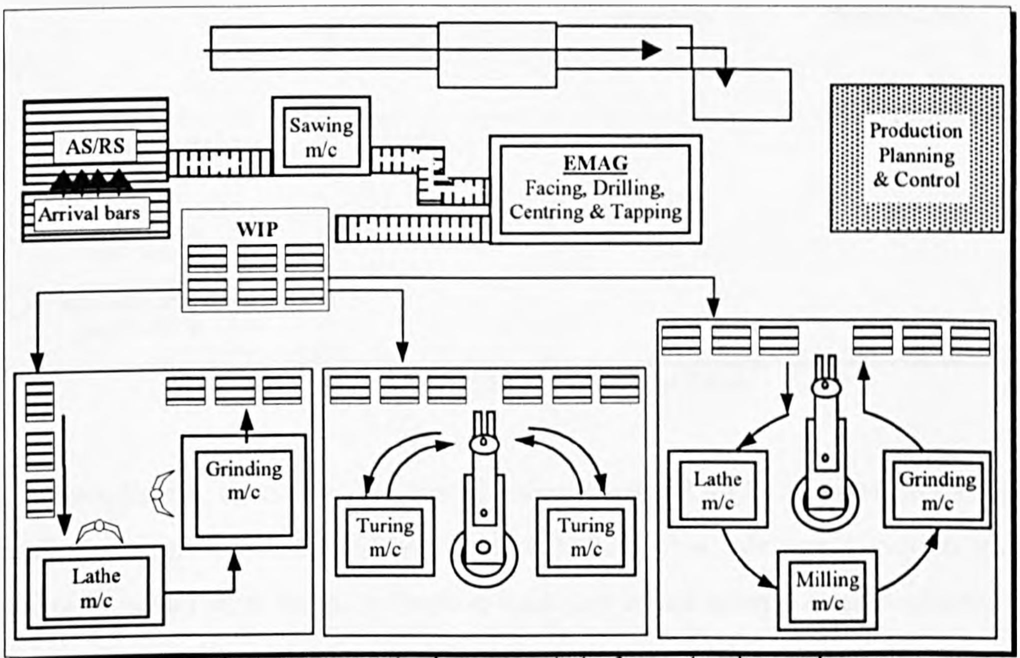


Figure D.20. The integrated shaft production unit.

Owing to the huge order varieties of motor shafts, the company adopted a Computer-Aided Manufacturing (CAM) system in the integrated shaft production unit, as shown in Figure D.20. The location of this unit is represented by area 6 on the general site plan (Figure D.3). There is another machining station located in area 4 in Figure D.3. This station produces special motor shafts in a few batches as these are more complicated; using a traditional manufacturing processes. The integrated unit and the special shaft unit produce orders for assembly for motor frame sizes 132 to 180. Shafts for small sizes (100-112) are produced in the same location as the assembly cells.

The implementation of shaft manufacturing processes is based upon the MRP schedule and special orders. Planners and controllers of shaft production reschedule the orders depending on their operation capacities and batch similarities. Therefore, they print out the production card which includes instructions related to manufacturing and design specifications. Figure D.12 shows a sample shaft production card.

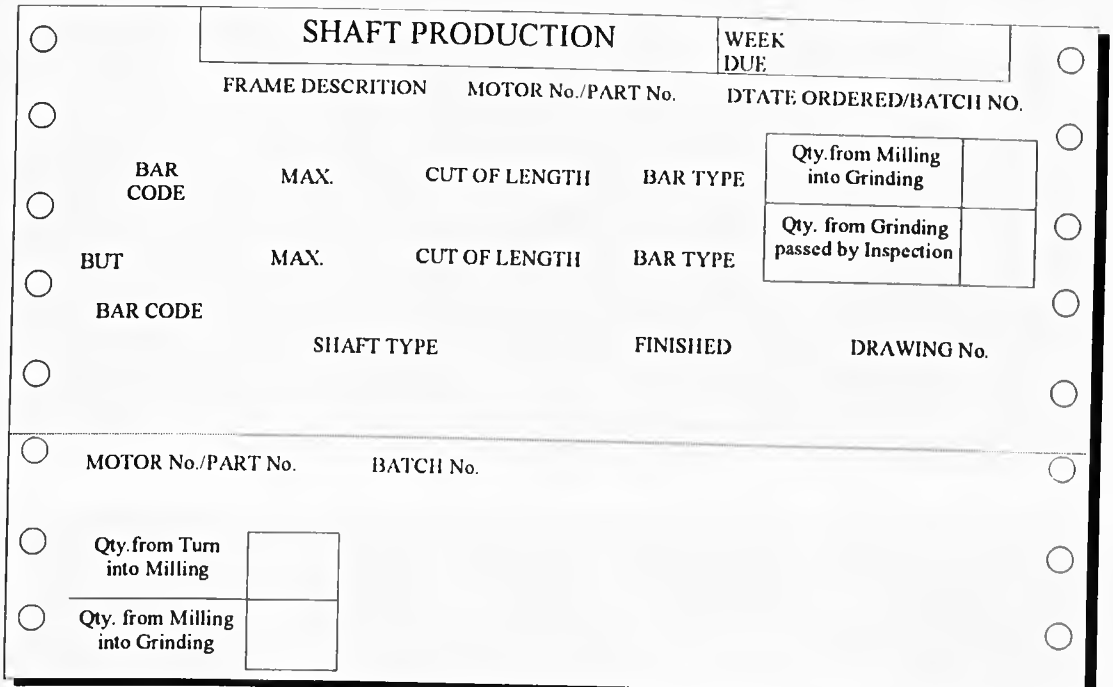


Figure D.21. Shaft production card.

Raw materials, e.g. steel bars, are stored using the AS/RS (Automated Stock/Retrieval System) and computer technologies (Figure D.22). This intelligent system identifies incoming bars and store them in specific locations based upon a classifications such as bar dimensions and material type.

To start the manufacturing processes for the current batch, the AS/RS retrieves the required bars and feeds the programmable sawing machine to cut bar stock into short pieces according to shaft lengths and tolerances. These small bars are conveyed to EMAG (Figure D.23) CNC machine for four types of primary machining operations; facing, drilling, centring and tapping. An industrial robot completes loading and unloading tasks for the EMAG machine. The machined bars are conveyed to a WIP storage area. Four manufacturing stations are employed to complete shaft production, Two stations work automatically (Figure D.24) and two manually. Finished shaft batches are transferred to shaft stock to be used in rotor assembly department.

Figure D.22. AS/RS at motor shaft production.

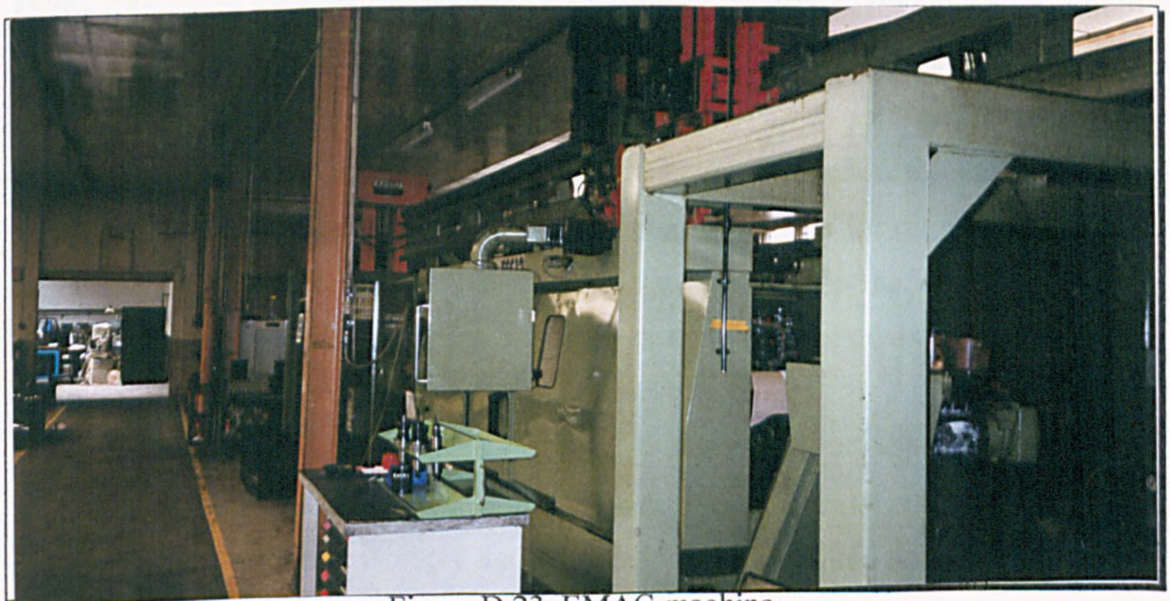


Figure D.23. EMAG machine.

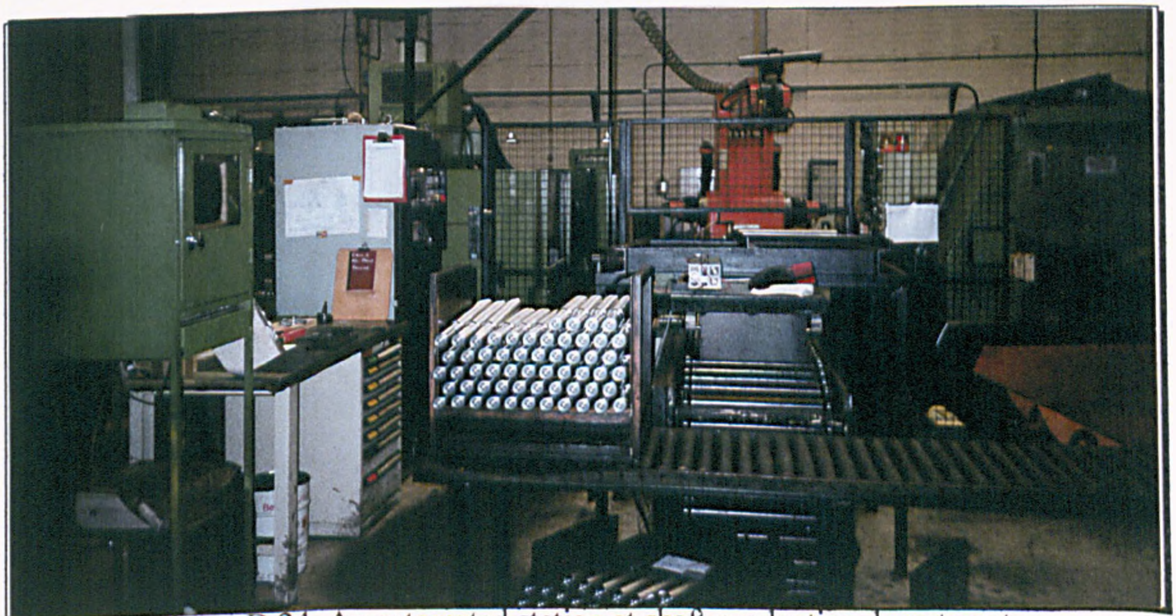


Figure D.24. An automated station at shaft production department.

D.4.1.4.3 Motor Rotors Assembly

According to the MRP system and customer special orders, finished shafts and rotors are assembled based upon the rotor assembly daily schedule. This schedule is updated primarily from the MRP schedule and assembly supervisor experience.

The rotor assembly department is located in area number 5, as shown in Figure D.3. This gives simple access between manufacturing stations of components (shafts and rotors) and the assembly location. The rotor assembly station layout is illustrated in Figure D.25.

Finished rotors are loaded into a conveyer to transfer them into an electric oven (Figure D.26). The rotors are heated to facilitate a shaft fitting operation (the second operation), before the sub-assemblies pass through a cooling process to decrease sub-assemblies temperature. These sub-assemblies are left in production for about three hours to adapt to the normal temperatures before the machining and balancing operations are completed. Lathes are used to complete machining operations for rotor subassemblies (Figure D.27), and the machined sub-assemblies are returned to the WIP area. Finally, balancing operations are carried out for each rotor assembly unit, as shown in Figure D.28.

The rotor assembly planning and control office produces a production card for every order. This card passes with every batch to carry out the required order specifications. This card also involves shafts/rotors numbers, locations, dimensions and quantities. Figure D.29 shows a sample of rotor assembly card.

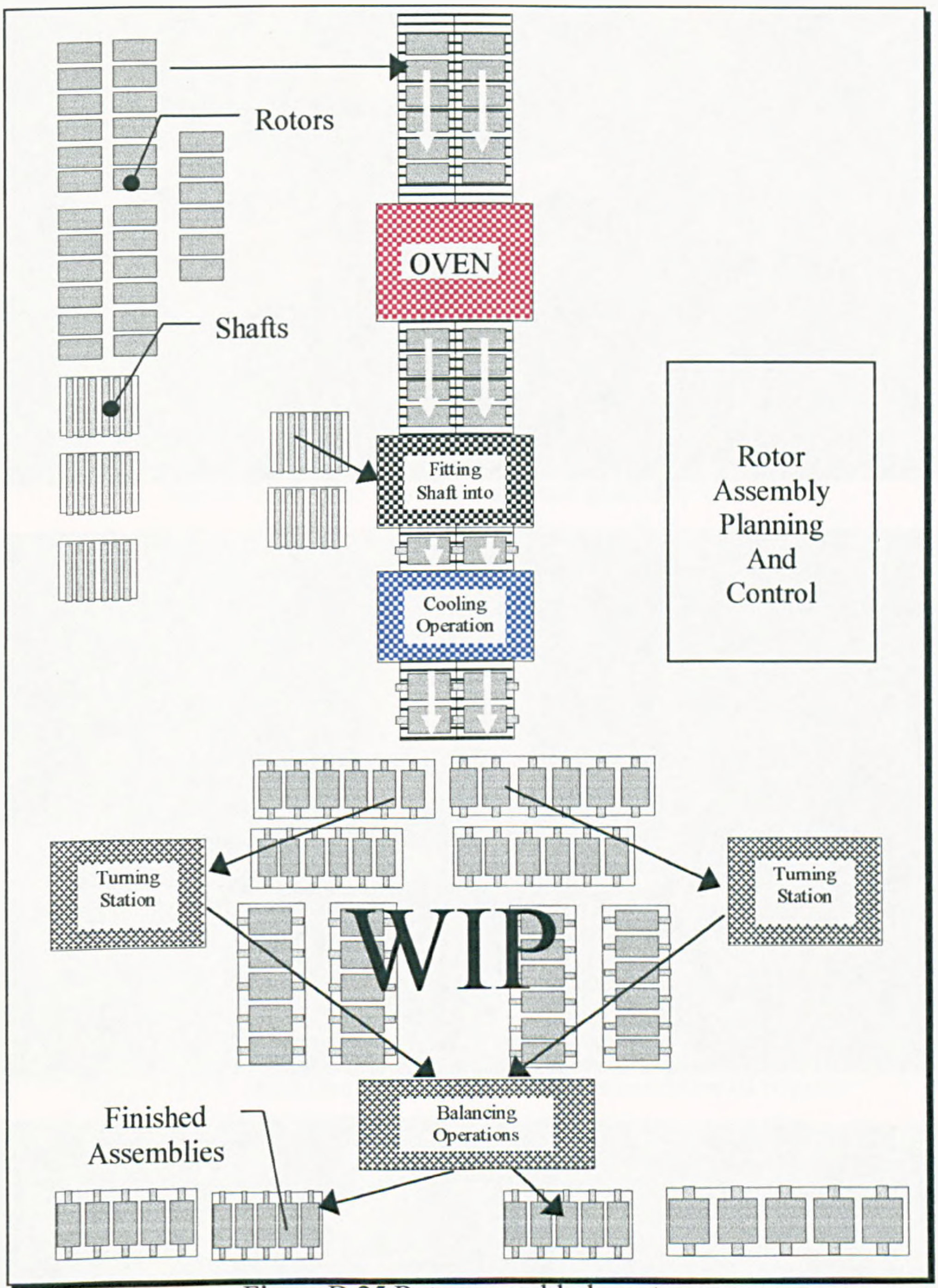


Figure D.25.Rotor assembly layout.

Figure D.26. The first part of rotor assembly line.

Figure D.27. Machining operation for sub-assemblies of rotors.

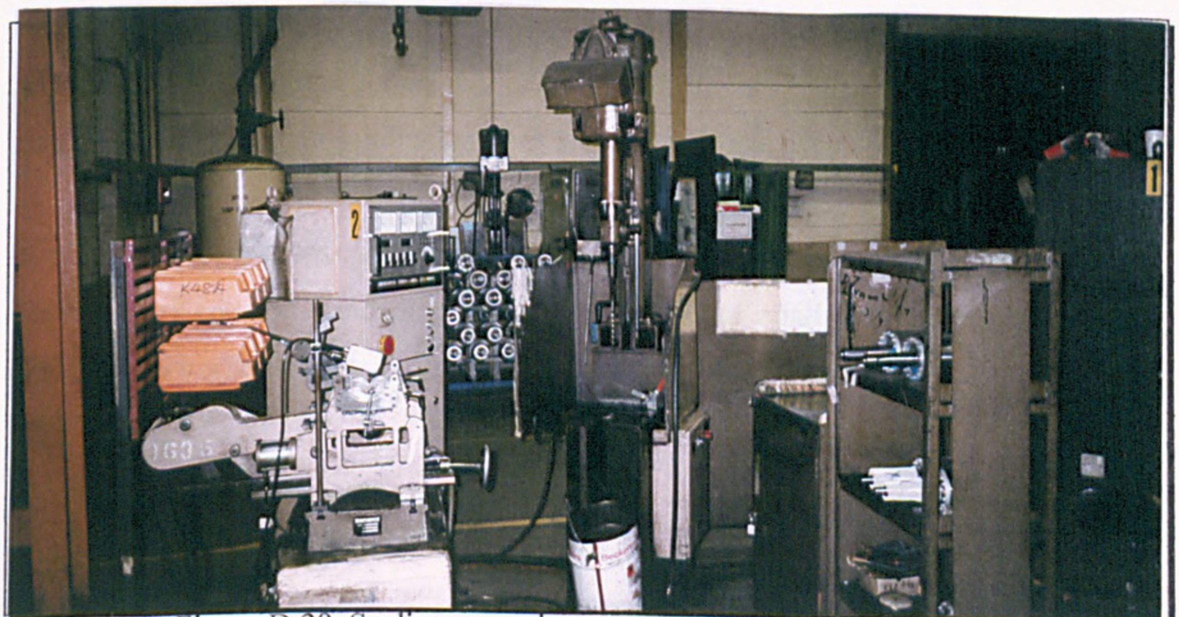


Figure D.28. Scaling operations at rotor assembly department.

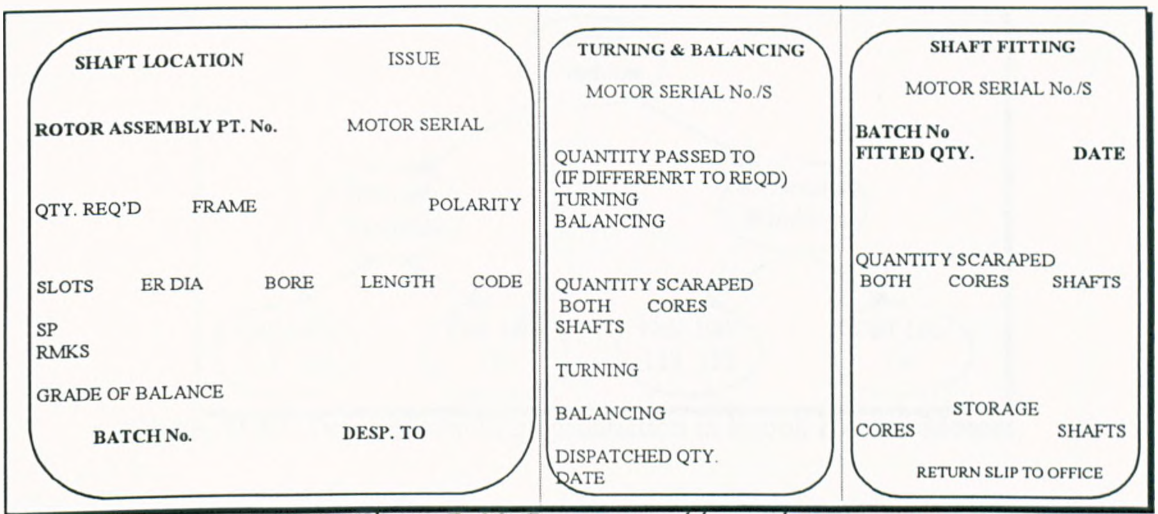


Figure D.29. Rotor assembly card.

D.4.1.4.4 Windings Assembly

The wound stator core is the heart of an electric motor and probably the most complex part to manufacture. The windings are insulated with copper wire and inserted into slots in the stator laminations. These slots have an insulator between the windings and the steel laminations. This is known as the "stator pack". Motor windings are designd to provide the output and speed required. The ends of the winding are brought out of the motor casting to a terminal box mounted on the frame.

Two types of winding assembly processes are used in Brook Hansen Motors: manual and machined (Figure D.30) for every manufacturing motor cell, owing to the range of customer orders relating to motor winding specifications. These customer specifications and requirements include the number of steel laminations, copper wire sizes, number of turns, electric capacities, lamination sizes and heat treatment specifications.

In cell 160/180, a production line is employed to produce the motor wound pack for both standard and non-standard motors. When in cells of small frame sizes, the winding operations are considered as a part of the motor final assembly line.

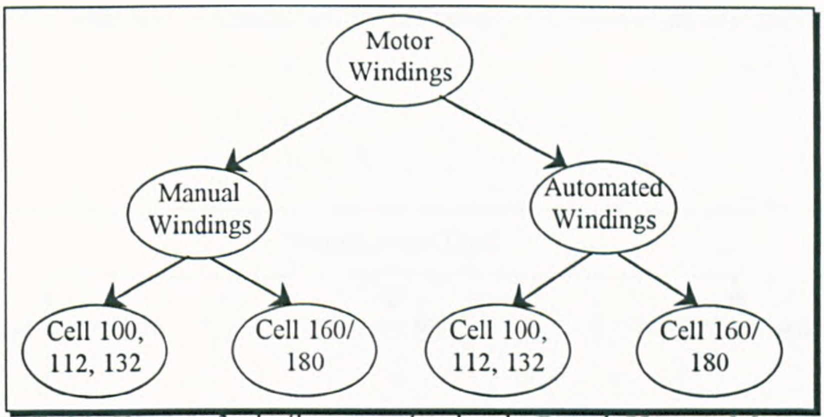


Figure D.30. Types of windings production in Brook Hansen Motors.

The 160/180 Winding supervisor receives the order quantities through the MRP system and special management orders. A primary production schedule is provided by MRP system based upon order due dates and batch quantities. The supervisor reschedules the batches received, depending on several factors such as machine capacities, machine set-ups and WIP levels. A production sheet is issued by the production supervisor for every batch of production using a system terminal located in the winding assembly area. The production sheet accompanies the product related during its production cycle to follow order specifications and manufacturing specifications. This sheet is kept with its specific batch until the final motor assembly is completed.

Two types of production sheets are generated by the supervisor's system for both standard and non-standard motors. The standard sheet is coloured white while the non-standard sheet is coloured pink. Every production sheet is divided into four sections containing:

1. General information about the batch order and customer details;
2. Rating details of motors such as motor output, full load speed, motor phase and hertz, and additional information;
3. Winding details: this section involves motor specifications related to windings such as wire sizes, number of turns, lamination slots, type of windings (m/c or hand), insulation details, connection types and additional information;
4. Assembly details: this section involves specifications of motor components and final assembly requirements and specifications.

The basic Bill of Material (BOM) structure for wound stator core is shown in Figure D.31. Table D.4 illustrates manufacturing processes of windings assembly and their description.

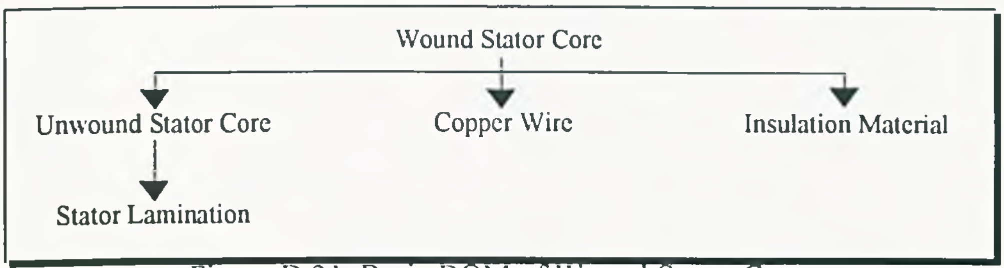


Figure D.31. Basic BOM of Wound Stator Core

Operation	Description
1. Cleater feeding	Operator selects required laminations based upon order specifications and collects them on board, after taking their balance using a scale.
2. Cleating operation	The operator loads the lamination batches to a cleater (machine). The cleater works to press the laminations and ropes then using strips of metal.
3. Cleating inspection	To check cleating operation and return rejected limitations to stock.
4. Insulation operation	An insulator (automated machine) inserts the insulation materials into cleated lamination slots.
5. Winding and Inserting	Automated winders and inserting machines are used to produce wire coils and insert that coils into unwound stator cores.
6. Finishing operations	These operations are completed manually. An insulation material needs to be added to wound stator pack and other finishing operations.
7. Connection operations	Relevant connections are carried out in this station. Connection operations are completed manually.
8. Press operation	After completing wound stator pack connections, they are pressed using pressing machine to reduce winding size.
9. Term and Binding operations	These operations are made manually
10. Electric test	In this station the wound stator pack is tested electrically using a special machine.
11. Heating & treatment operations	All wound stator packs enter the oven and treated for four hours to increase the strength and temperature resistant.
12. Cooling operations	The wound stator packs go through cooling stations. This station uses air for this purpose.
13. Cleaning operations	These operations clean the wound stator pack using two stations of Wyvertic machines.

Table D.4. Windings assembly operations.

Figures D.32, D.33 and D.34 show the windings assembly line.

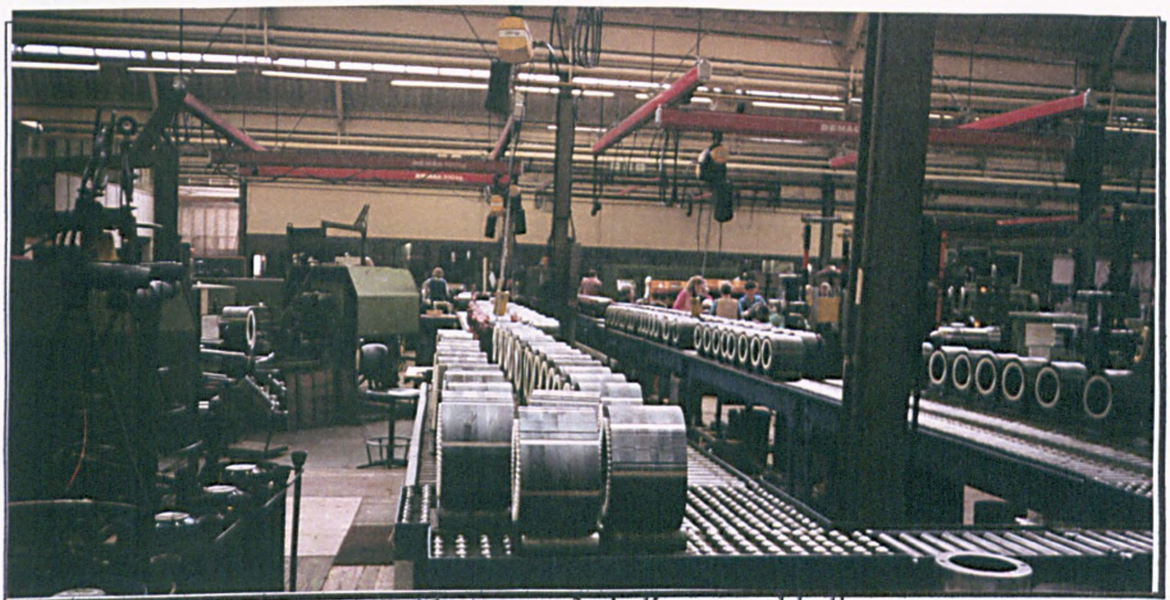


Figure D.32. A part of winding assembly line

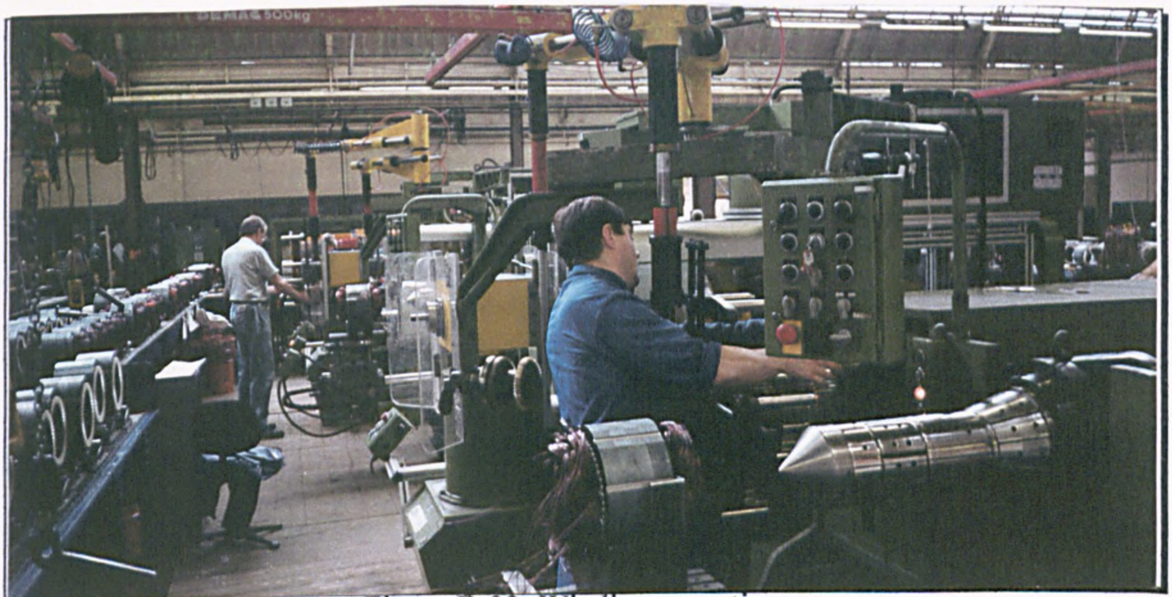


Figure D.33. Winding operations.

Figure D.34. Wyvertic stations.

D.4.1.4.5 Foundry Components

The Aluminium die casting foundry is engaged in the production of aluminium components for use in the manufacturing of electric motors. A facility also exists for machining cast iron components to supply similar manufacturing functions.

Aluminium (AL-LM24) is the raw material used to cast 75 to 80% of components, and the AL-LM24 quantity used is 25 ton/week. The new order of raw material takes about three weeks to arrive. Figure D.35 illustrates Aluminium ingots that are used in die casting operations.

The principal tasks are to cast from raw material (AL ingots) to the required company standards and customer order specifications. Components are supplied in cast and fettled condition and, when required, machined to design specification within the foundry.

Several components of electric motors are produced in foundry department; such as (Figure D.36):

- Motors frames in different sizes (100, 112, 132, 160 and 180 standards for Huddersfield site and 63, 71, 80 and 90 standards for Honley site);
- Terminal boxes;
- Drive and non-drive end-shields;
- Flanges;
- Box lids;
- Bearing caps.

Foundry department receives huge number of orders from both manufacturing sites (Huddersfield and Honley). Planners and controllers develop manual schedules for orders received according to their machine capacities and customer due dates. These production schedules are weekly, and can be divided into two types of schedules: future schedules (type A) and real time schedules (type B). Type A schedules are developed before starting production to a real time schedule reference plan. Type B schedules are basically similar to type A but can be updated from time to time to assimilate changes which have occurred during production.

Kanban has been used in die casting production in the past but did not give results due to several reasons such as a shortage in Kanban baskets, management and control of Kanban system movements and the large range of orders received. The MRP concept has been implemented in the foundry to cope with assembly lines orders but it has been found that some batches can not be controlled by the MRP system.

Latterly, a mixture of Kanban and MRP systems have been implemented. Kanban covers approximately 25% of production and MRP covers approximately 75%.

Table D.5 illustrates the die casting machine types and the components they produce. A general layout of the foundry department is illustrated in Figure D.37. This department also carries out cleaning, machining and painting operations for cast parts.

Figure D.35. AL ingots used by Brook Hansen Motors.

Figure D.36. Some motor components produced by foundry department.

Machine	Component	Size
650 IDRA	Frames	D90L, D90S, D80, D63L, D63S, V90 sets, V80 sets and V63 sets, W71S, W71M, W90L
	End-shields	W160DE, W160NDE, W90DE, D160DE, D160NDE
	Flanges	W100, W112, W132
	Terminal boxes	W100, W112, W160, W71
750 TRIULZI(1)	Frames	W132S, W112, W100, D132S/4, D112, D100, D90
	End-shields	W100DE&NDE, W112DE&NDE, W132DE&NDE, W160DE&NDE, W180DE&NDE, D132DE&NDE
	Flanges	W132, D112
	Terminal boxes & Lids	W100, W112, W160
630 TRIULZI	Frames	D100, D90L, D80, D60S, D63L, D71, V90sets, V80sets, V63sets
	End-shields	D160DE&NDE, W160DE&NDE
	Flanges	W112, W100
	Terminal boxes & Lids	W112, W100, D71
800 TRIULZI	Frames	W132S, W132M, W112, W100, D132, D112, D100
	End-shields	W100DE&NDE, W112DE&NDE, W132DE&NDE, W160DE&NDE, W180DE&NDE
	Flanges	W132
	Terminal boxes & Lids	W112, W160, W100
600 TRIULZI	Frames	D90S, D90L, D80, D63S, D63L, D71, V90sets, V80sets, V63sets
400 WOTAN (2)	End-shields	D90DE, D80DE&NDE, D90DEGen
	Bearing caps	D63-90, D100, D160-180, D132
	Terminal boxes	D112, D160, RS100, ED112
	Pads	D63-90, W100-132, W160-180
	Foot	D&O
480 TRIULZI	End-shields	D90DE&NDE, D80DE&NDE, D100DE&NDE
	Brake adopters	D100, D112, D132
	Pads	D100, D112/132, D160, W100, W112, W132, W160, W180
	Terminal boxes	D160
400 TRIULZI	Terminal boxes	D112, D160, ED112, RS100
	Pads	D63-90, W71, W100-132, W160-180
	Brake adopters	D100, D112, D132, W100, W112, W132
	Others	D80 foot, LC67DE&NDE, D90Ger DE
400 WOTAN (1)	Pads	D63-90
	Bearing caps	D63-90, D100, D160-180, D132
	Others	C&D Box, RS100 Box, ED112 Box
480 TRIULZI (2)	Frames	D90S, D90L, D80, D63S, D63L, D71S, D71L
500 CAST.	End-shields	D100DE&NDE, D90DE, D80DE&NDE
1200 IDRA	Frames	W80, W160, D180, D160, D132, W132S, D132M
	End-shields	W100DE&NDE, W112DE&NDE, W180DE&NDE
600 CAST.	End-shields	D100DE&NDE, D132DE&NDE, D90DE, D80DE&NDE
700 WOTAN	End-shields	D132DE&NDE, W132DE&NDE
950 TRIULZI	Frames	W180, W160, W132S, W132M, D180, D160, D132S/M, D132Int foot
	End-shields	W100DE&NDE, W112DE&NDE, W180DE&NDE, D112DE&NDE
	Flanges	W132
	Others	Int. box &lid, T. body head
750 TRIULZI(2)	Frames	W112, W100, D132S/M, D112, D100, D90
	End-shields	D160DE, W100DE&NDE, W112DE&NDE, W132DE&NDE, W160DE&NDE, W180DE&NDE, D112NDE, D132DE&NDE, D160DE&NDE
	Flanges	W132, W112
	Terminal boxes &Lids	W100, W112, W160

Table D.5. Die casting machines and the motor component produced.

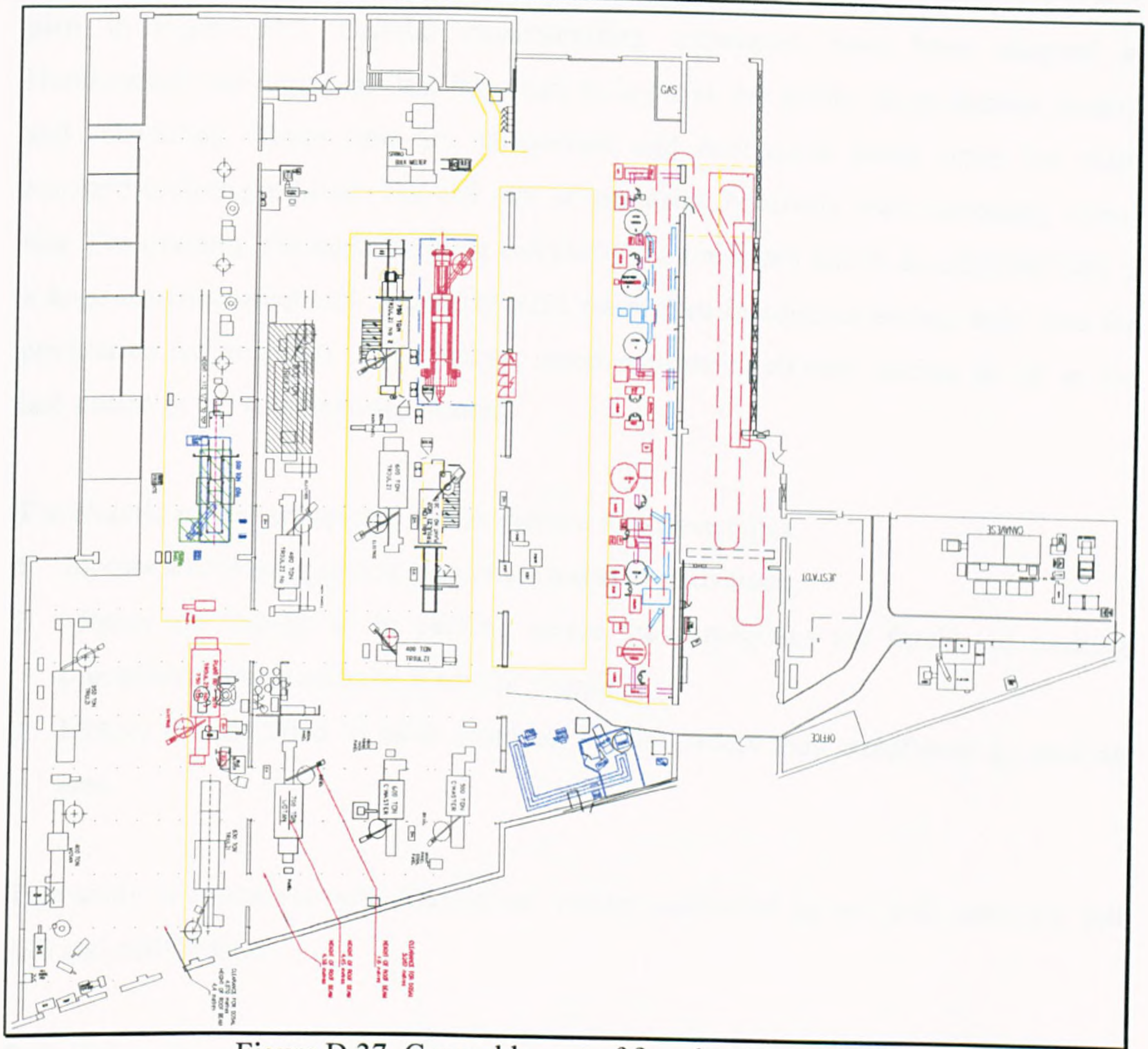


Figure D.37. General layout of foundry department.

D.4.1.4.6 Motor Assembly Cells

Motor assembly lines represent the main manufacturing cells at Huddersfield site. These main cells are fed by motor components which are produced by the other manufacturing departments. The main assembly cells are:

- Cell-160/180;
- Cell-132;
- Cell-112;
- Cell-100;
- CI Cell.

Cell-160/180 is located in area 9, Cell-100, Cell-112, and Cell-132 are located in area 11 and the cast iron production line is located in area 12 as illustrated in the main site plan in Figure D.3. Cellular manufacturing techniques have been adopted at Huddersfield site during the last few years to improve the ability of production control and scheduling. These cells are categorised and configured based upon the main standard motors produced. The cell size is increasing relatively with increasing motor size. For example, the manufacturing cell (100) is a small cell where as cell (160/180) is a large manufacturing cell. Assembly ticks are printed in advance during daily runs for production requirements. These tickets travel with their relevant batches as far as the last station in the assembly department.

The electric motors completed can be divided into three types:

1. Motors which are standard motors are transferred to stock;
2. Motors are transferred to packing where the nameplates are fitted and packing operations completed to be ready for shipping;
3. Motors are modified to meet customer special orders then transferred to packing area.

This study is concerned with Aluminium motors assembled by cell-100, cell-112, cell 132 and cell-160/180.

Cell-100 is located in Works-9 and dedicated to producing small motors (motor size 100). The production line of this cell can be divided into two main parts; winding assembly and motor assembly. The windings are the most complicated parts of electric motors, along with wound pack this is the first part that must be assembled with the motor frame; hence, the winding production line represents the first part of the motor assembly line in every manufacturing cell. Cell-100 also involves rotor production, shaft production and rotor assembly. It is also the same for cell-112 and cell 132. It should be noted that motor shafts that supply cell-132 are manufactured in the shaft department as discussed previously. Figure D.38 illustrates cell-100, cell-112 and cell-132. Figure D.39 illustrates cell-160/180 and the windings assembly for this cell.

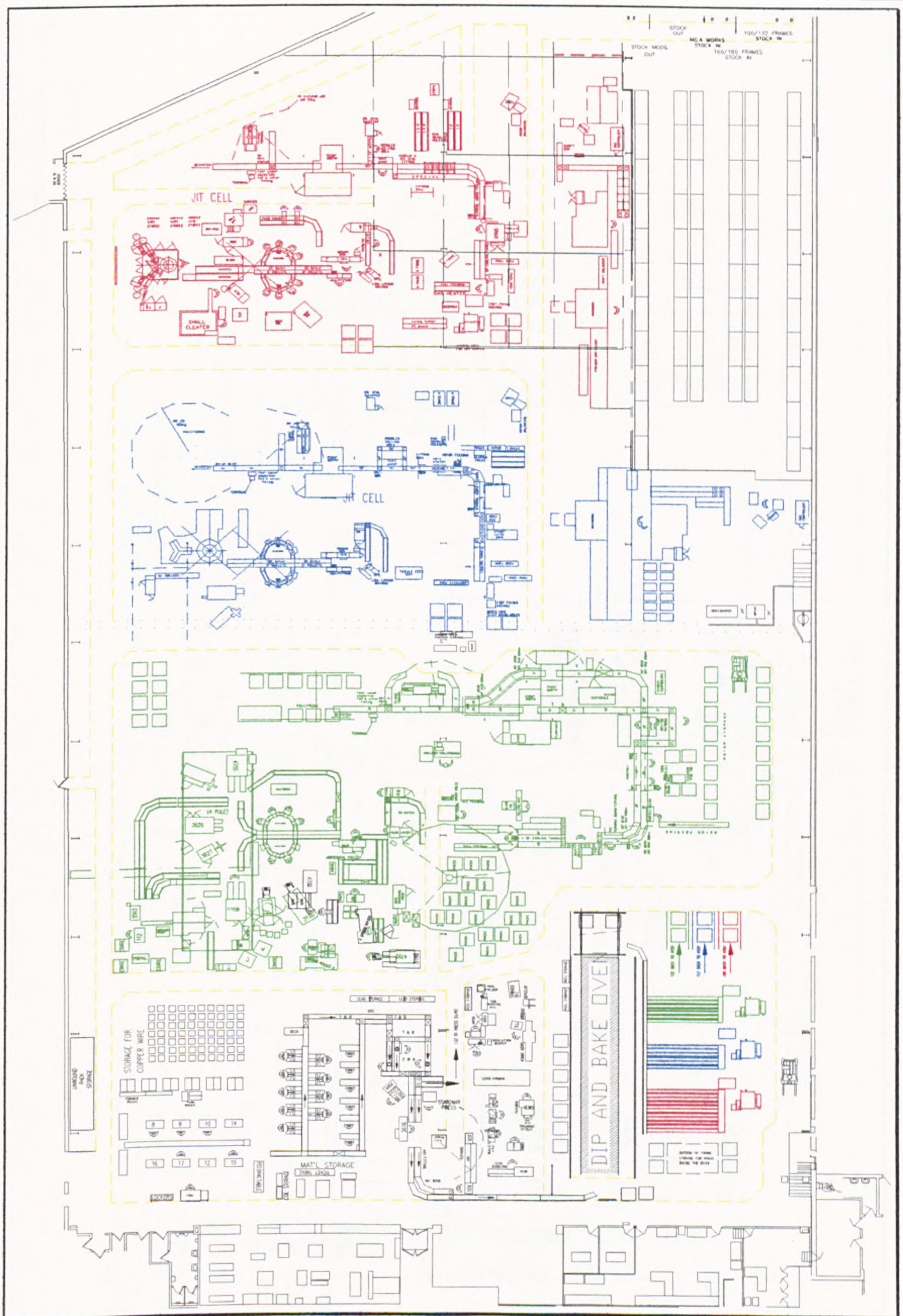


Figure D.38. Cell-100, 112 and 132.

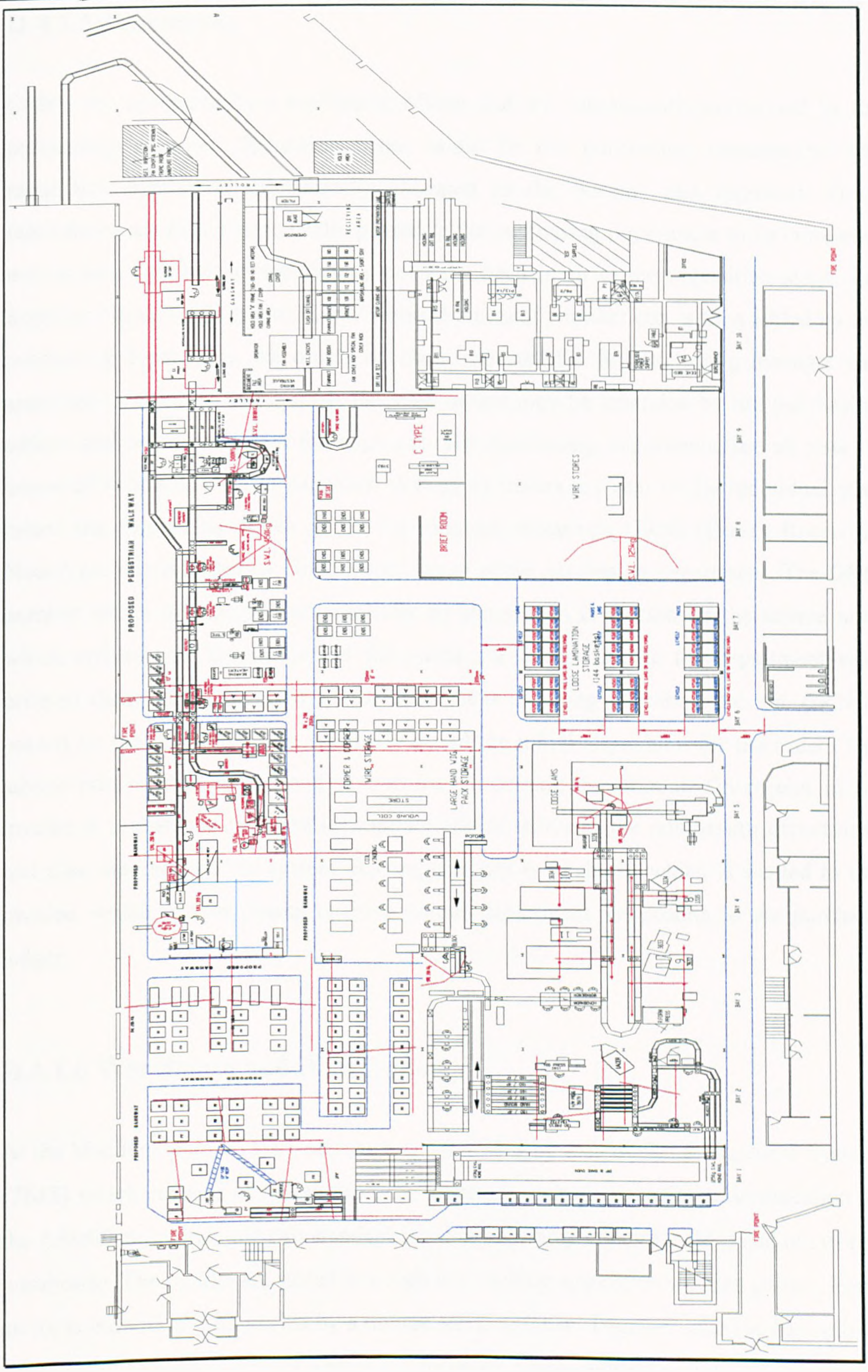


Figure D.39. Cell-160/180.

D.4.1.5 Purchasing

Orders are processed by a purchasing officer and are subsequently authorised by the purchasing manager. Requisitions are raised in the purchasing department. The requisition number is automatically allocated by the system and approved. These requisitions are then electronically passed to the purchasing department to be converted into orders. Information which may not have been input at the requisition stage, i.e. Supplier Name, Organisation Code/Nominal Account Number etc. is then added by the purchasing department. All orders are then inspected by the purchasing manager and approved or rejected as required. Rejected orders may be amended by the purchasing officer and submitted again for approval. The purchasing department fax all new or amended orders directly to suppliers. A copy of orders are sent to the individual who raised the requisition. When goods are received, electronic GRNs (Goods Receiving Notes) are raised, either by Good Receiving or is the purchasing department. The GRN number which is allocated automatically by the system is written on the advice note which arrives with the goods and the goods are then passed to the department who ordered them, along with the advice note. After checking quantities etc. the GRN is passed by the individual who originally raised the initial requisition for the order. The advice note is then signed and sent to the purchasing department. On receipt of an invoice it is matched with the appropriate advice note/s in the purchasing department and then entered into the system. A computer slip is produced which is stapled to the invoice, which is then passed to the accounts department for posting to the purchase ledger.

D.4.1.6 Warehouse and Distribution

At the Huddersfield site the warehouse is controlled by the Truck Management System (TMS) which consists of two trucks with terminals, linked via radio communication to the AS/400 computer, and two standard terminals for booking motors into and out of the warehouse. The motors are stored in a high bay racking system on wooden pallets. Each motor is individually identified by a unique serial number. Together with the bar-coding technology, this gives a high degree of accuracy when auctioning into and/or out of stock transactions.

Stock motors arrive in the warehouse from the assembly locations. Motors in shortage are placed into and taken out of stock, otherwise they are placed on pallets ready to go into the racks. The TMS handles the location and retrieval of the motors using a complex set of picking rules and control features. Truck one is designed to enable the picking of individual motors from pallets whilst they are in the tracks; this truck cannot remove or place complete pallets. Motors which are too heavy to man handle must be removed using truck two. Truck one has a height restriction which means it cannot safely reach the top two or bottom two locations.

In order to provide the necessary of delivery reliability, the UK is split into a number of separate delivery areas. Vehicles are then loaded on set nights and despatched on set dates during the week – with special arrangement as and when required to supplement the standard service. Huddersfield motor production tickets/stock tickets show relevant the despatch information and wagon run numbers for physical loading. The ticket is bar-coded at despatch stage to raise the consignment note/invoice and to update progress records on the computer screen. Pallet labels can also be produced for motors palletised by bar-coding despatch label.

In Huddersfield, export distribution motors received into packing contain the inspection ticket/stock label which gives all relevant shipping / packing details. From this the official packing label can be produced for attachment to the box once packed and stored in the Export Warehouse. The packing ticket is then processed by the Export shipping department for consolidation into a physical shipment. Relevant export documentation is then produced via the AS400 system with minimal input. Details are stored in the computer and retrieved automatically and then printed off via a laser printer; this also updates progress information at this stage. Table D.5 illustrates the despatch and progress records for six months.

Description	Huddersfield Production Information	
	Average	Maximum
Standard motors into stock per day	400	800
Non-standard motors inspected	350	800
Out of stock per day	400	800
Total warehouse stock	5,000	10,000
Despatched per day (Home)	360	800
Despatched per day (Export)	390	800

Table D.5 Despatching quantities.

D.5. Conclusion

This Appendix presents a description of the company selected as a case study for this research. It gives the background to manufacturing activities. The activities described have a direct relationships to manufacturing operations. It has been found that Brook Hasten Motors has been working for a long time and has been redeveloped and re-changed several times.

The information presented by this Appendix will be used in Chapter-6 for the analysis of manufacturing systems of Brook Hansen Motors using the GI-SIM modelling method.

APPENDIX-E

E.1. Model Listing for Order Process

```
STD_ARR    CREATE,     1:EXPO(5):MARK(ARRIVAL_T);
30$        ASSIGN:     ORDER TYPE=DISC(.5,1,1,2);
QA11       QUEUE,      A1_Q;
29$        SEIZE,      1:A1_R,1;
28$        DELAY:      TRIA(8,10,12,1);
27$        RELEASE:    A1_R,1;
26$        BRANCH,     1:If,ORDER TYPE ==1,25$,Yes:
                       Else,22$,Yes;
25$        TALLY:      STD_ORDER_IN_SYSTEM,INT(ARRIVAL_T),1;
24$        COUNT:      # OF STD ORDERS,1;
23$        DISPOSE;
22$        QUEUE,      A2_Q;
21$        SEIZE,      1:A2_R,1;
20$        DELAY:      TRIA(10,15,20,2);
19$        RELEASE:    A2_R,1;
18$        QUEUE,      A3_Q;
17$        SEIZE,      1:A3_R,1;
16$        DELAY:      TRIA(10,13,20,4);
15$        RELEASE:    A3_R,1;
14$        QUEUE,      A4_Q;
13$        SEIZE,      1:A4_R,1;
12$        DELAY:      TRIA(10,12,20,5);
11$        RELEASE:    A4_R,1;
10$        QUEUE,      A5_Q;
9$         SEIZE,      1:A5_R,1;
8$         DELAY:      TRIA(10,15,20,6);
7$         RELEASE:    A5_R,1;
6$         QUEUE,      A6_Q;
5$         SEIZE,      1:A6_R,1;
4$         DELAY:      TRIA(10,17,22,7);
3$         RELEASE:    A6_R,1;
2$         TALLY:      NON_STD_ORDER_IN_SYSTEM,INT(ARRIVAL_T),1;
1$         COUNT:      # OF NON_STD ORDERS,1;
0$         DISPOSE;
```

E.2. Experiment Listing for Order Processing

```
PROJECT,      Order Processing,A.m.A,2/12/1998,Yes;

ATTRIBUTES:   1,STANDARD ORDER:
              2,NON_STANDARD ORDER:
              ORDER TYPE:
              ARRIVAL_T;

QUEUES:       A2_Q,FIFO:
              A3_Q,FIFO:
              A4_Q,FIFO:
              A5_Q,FIFO:
              A6_Q,FIFO:
              A1_Q,FIFO;

RESOURCES:    A1_R,Capacity(1,):
              A2_R,Capacity(1,):
              A3_R,Capacity(1,):
              A4_R,Capacity(1,):
              A5_R,Capacity(1,):
              A6_R,Capacity(1,);

COUNTERS:     # OF STD ORDERS,,Replicate,"C:\BH\SIM\OP\STD_CT.001":
              # OF NON_STD ORDERS,,Replicate,"C:\BH\SIM\OP\NONS_CT.001";

TALLIES:      NON_STD_ORDER_IN_SYSTEM,"C:\BH\SIM\OP\NONS_TI.001":
              STD_ORDER_IN_SYSTEM,"C:\BH\SIM\OP\STD_T.001";

DSTATS:       NR(A1_R):
```

```
                NR(A2_R):
                NR(A3_R):
                NR(A4_R):
                NR(A5_R):
                NR(A6_R):
                NQ(A1_Q):
                NQ(A2_Q):
                NQ(A3_Q):
                NQ(A4_Q):
                NQ(A5_Q):
                NQ(A6_Q);

REPLICATE,      4,0.0,2400,Yes,Yes,0.0;
```

E.3. Model Listing for Rotor Production Station-1

```
0$          STATION,     LEAF_ST1;
10$         TRACE,       -1,"-Arrived to station LEAF_ST1\n";
40$         DELAY:       0.;
17$         COUNT:       FINISHED ROTORS(ROTOR TYPE),1;
12$         TALLY:       TIME IN ST1(ROTOR TYPE),Interval(TIME IN),1;
3$          TRACE,       -1,"-Disposing entity\n";
11$         DISPOSE;
1$          STATION,     STATION_1;
70$         TRACE,       -1,"-Arrived to station STATION_1\n";
190$        STORE:       STATION_1_S1;
67$         TRACE,       -1,"-Delay for unloading time NORM(.1,.03,1)\n";
191$        DELAY:       NORM(.1,.03,1);
175$        UNSTORE:     ;
63$         TRACE,       -1,"-Waiting for resource STATION_1_R\n";
172$        QUEUE,       STATION_1_R_Q;
169$        SEIZE,       1:STATION_1_R,1;
97$         STORE:       STATION_1_S3;
            TRACE,       -1,"-Delay for processing time NORM(2,.2,1)\n";
156$        DELAY:       NORM(2,.2,1);
133$        UNSTORE:     ;
126$        STORE:       STATION_1_S4;
60$         TRACE,       -1,"-Releasing resource\n";
149$        RELEASE:     STATION_1_R,1;
57$         TRACE,       -1,"-Delay for loading time NORM(.2,.02,1)\n";
113$        DELAY:       NORM(.2,.02,1);
110$        UNSTORE:     ;
54$         TRACE,       -1,"-Transferred to station LEAF_ST1\n";
111$        ROUTE:       .2,LEAF_ST1;
216$        CREATE,      BSIZE:NORM(60,2,1):MARK(TIME IN);
215$        ASSIGN:      ROTOR TYPE=DISC(.5,1,1,2);
256$        STATION,     ARR_ST1;
207$        TRACE,       -1,"-Arrived to system at station ARR_ST1\n";
253$        ASSIGN:      Picture=Default;
232$        DELAY:       0.;
219$        ASSIGN:      Picture=R_PIC(ROTOR TYPE);
201$        TRACE,       -1,"-Transferred to station STATION_1\n";
230$        ROUTE:       .1,STATION_1;
```

E.4. Experiment Listing for Rotor Production Station-1

```
PROJECT,        ;

ATTRIBUTES:     ROTOR TYPE:
                TIME IN;

STORAGES:       STATION_1_S4:
                STATION_1_S3:
                STATION_1_S1;

QUEUES:         STATION_1_R_Q,FIFO;

PICTURES:       Default:
                R100_PIC:
                R112_PIC;

RESOURCES:      STATION_1_R,Capacity(1,);
```

```
STATIONS:       ARR_ST1:
                STATION_1:
                LEAF_ST1;

COUNTERS:       ROTORS_100 AT ST1,,Replicate,"C:\BH\SIM\ROTOR\C100S1.ST1":
                ROTORS_112 AT ST1,,Replicate,"C:\BH\SIM\ROTOR\C112S1.ST1";

TALLIES:        T_R_100_AT_ST1,"C:\BH\SIM\ROTOR\T100S1.ST1":
                T_R_112_AT_ST1,"C:\BH\SIM\ROTOR\T112S1.ST1";

DSTATS:         MR(STATION_1_R),STATION_1_R Available:
                NR(STATION_1_R),STATION_1_R Busy,"C:\BH\SIM\ROTOR\S1UT.ST1":
                NQ(STATION_1_R_Q),# in STATION_1_R_Q:
                BSIZE,,"C:\BH\SIM\ROTOR\S1BS.ST1";

REPLICATE,      1,0.0,2265,Yes,Yes;

EXPRESSIONS:    BSIZE,AINT(NORM(35,5,1));

SETS:           FINISHED ROTORS,ROTORS_100 AT ST1,ROTORS_112 AT ST1:
                TIME IN ST1,T_R_100_AT_ST1,T_R_112_AT_ST1:
                R_PIC,R100_PIC,R112_PIC;
```

E.5. Model Listing for Shaft Production (Integrated Production Unit)

```
2$          STATION,        TO_CELL3_FROM4;
79$         TRACE,          -1,"-Arrived to station TO_CELL3_FROM4\n";
99$         STORE:          TO_CELL3_FROM4_S1;
100$        DELAY:          0.;
85$         UNSTORE:        ;
80$         DELAY:          0.000:NEXT(1$);

1$          TRACE,          -1,"-Splitting group representative\n";
101$        SPLIT:          A(*),M,NS,IS:NEXT(0$);

0$          DELAY:          0.00;
109$        TRACE,          -1,"-Delay for loading time 1\n";
132$        DELAY:          1;
105$        TRACE,          -1,"-Transferred to station CELL_3\n";
130$        ROUTE:          10,CELL_3;
6$          STATION,        CELL4_BATCH;
150$        TRACE,          -1,"-Arrived to station CELL4_BATCH\n";
170$        STORE:          CELL4_BATCH_S1;
171$        DELAY:          0.;
156$        UNSTORE:        ;
151$        DELAY:          0.000:NEXT(5$);

5$          DELAY:          0.00;
174$        TRACE,          -1,"-Waiting for batch size of 5\n";
182$        GROUP,          :5,Last;
176$        TRACE,          -1,"-Entity created to represent temporary batch of size
5\n";
180$        DELAY:          0.000:NEXT(3$);

3$          DELAY:          0.00;
213$        DELAY:          0.;
186$        TRACE,          -1,"-Transferred to station TO_CELL3_FROM4\n";
211$        ROUTE:          2,TO_CELL3_FROM4;
7$          STATION,        F_CELL3;
232$        TRACE,          -1,"-Arrived to station F_CELL3\n";
262$        DELAY:          0.;
240$        COUNT:          F_CELL3_C,1;
235$        TALLY:          F_CELL3_Ta,Interval(ARRIVAL TIME),1;
225$        TRACE,          -1,"-Disposing entity\n";
233$        DISPOSE;
8$          STATION,        F_CELL2;
270$        TRACE,          -1,"-Arrived to station F_CELL2\n";
300$        DELAY:          0.;
264$        TRACE,          -1,"-Exiting conveyor\n";
290$        EXIT:           ;
278$        COUNT:          F_CELL2_C,1;
273$        TALLY:          F_CELL2_Ta,Interval(ARRIVAL TIME),1;
263$        TRACE,          -1,"-Disposing entity\n";
271$        DISPOSE;
9$          STATION,        F_CELL1;
308$        TRACE,          -1,"-Arrived to station F_CELL1\n";
```

```
338$    DELAY:     0.;
302$    TRACE,     -1,"-Exiting conveyor\n";
328$    EXIT:      ;
316$    COUNT:     F_CELL1_C,1;
311$    TALLY:     F_CELL1_Ta,Interval(ARRIVAL TIME),1;
301$    TRACE,     -1,"-Disposing entity\n";
309$    DISPOSE;
11$     STATION,   CELL_4;
368$    TRACE,     -1,"-Arrived to station CELL_4\n";
488$    STORE:     CELL_4_S1;
489$    DELAY:     0.;
473$    UNSTORE:   ;
361$    TRACE,     -1,"-Waiting for resource CELL_4_R\n";
470$    QUEUE,     CELL_4_R_Q;
467$    SEIZE,     1:CELL_4_R,1;
395$    STORE:     CELL_4_S3;
        TRACE,     -1,"-Delay for processing time TRIA(4,5,7)\n";
454$    DELAY:     TRIA(4,5,7);
431$    UNSTORE:   ;
424$    STORE:     CELL_4_S4;
358$    TRACE,     -1,"-Releasing resource\n";
447$    RELEASE:   CELL_4_R,1;
411$    DELAY:     0.;
408$    UNSTORE:   ;
352$    TRACE,     -1,"-Transferred to station CELL4_BATCH\n";
409$    ROUTE:     1,CELL4_BATCH;
14$     STATION,   CELL_3;
519$    TRACE,     -1,"-Arrived to station CELL_3\n";
639$    STORE:     CELL_3_S1;
640$    DELAY:     0.;
624$    UNSTORE:   ;
512$    TRACE,     -1,"-Waiting for resource CELL_3_R\n";
621$    QUEUE,     CELL_3_R_Q;
618$    SEIZE,     1:CELL_3_R,1;
546$    STORE:     CELL_3_S3;
        TRACE,     -1,"-Delay for processing time TRIA(10,15,20)\n";
605$    DELAY:     TRIA(10,15,20);
582$    UNSTORE:   ;
575$    STORE:     CELL_3_S4;
509$    TRACE,     -1,"-Releasing resource\n";
598$    RELEASE:   CELL_3_R,1;
562$    DELAY:     0.;
559$    UNSTORE:   ;
503$    TRACE,     -1,"-Transferred to station F_CELL3\n";
560$    ROUTE:     1,F_CELL3;
17$     STATION,   CELL_2;
670$    TRACE,     -1,"-Arrived to station CELL_2\n";
790$    STORE:     CELL_2_S1;
791$    DELAY:     0.;
775$    UNSTORE:   ;
663$    TRACE,     -1,"-Waiting for resource CELL_2_R\n";
772$    QUEUE,     CELL_2_R_Q;
769$    SEIZE,     1:CELL_2_R,1;
697$    STORE:     CELL_2_S3;
        TRACE,     -1,"-Delay for processing time TRIA(7,7.5,8)\n";
756$    DELAY:     TRIA(7,7.5,8);
733$    UNSTORE:   ;
726$    STORE:     CELL_2_S4;
660$    TRACE,     -1,"-Releasing resource\n";
749$    RELEASE:   CELL_2_R,1;
657$    TRACE,     -1,"-Delay for loading time 1\n";
713$    DELAY:     1;
710$    UNSTORE:   ;
654$    TRACE,     -1,"-Transferred to station F_CELL2\n";
705$    CONVEY:    ,F_CELL2;
21$     STATION,   CELL_1;
821$    TRACE,     -1,"-Arrived to station CELL_1\n";
941$    STORE:     CELL_1_S1;
942$    DELAY:     0.;
926$    UNSTORE:   ;
814$    TRACE,     -1,"-Waiting for resource CELL_1_R\n";
923$    QUEUE,     CELL_1_R_Q;
920$    SEIZE,     1:CELL_1_R,1;
848$    STORE:     CELL_1_S3;
        TRACE,     -1,"-Delay for processing time TRIA(7,7.5,8)\n";
907$    DELAY:     TRIA(7,7.5,8);
884$    UNSTORE:   ;
877$    STORE:     CELL_1_S4;
811$    TRACE,     -1,"-Releasing resource\n";
900$    RELEASE:   CELL_1_R,1;
```

```
808$        TRACE,       -1,"-Delay for loading time 1\n";
864$        DELAY:       1;
861$        UNSTORE:     ;
805$        TRACE,       -1,"-Transferred to station F_CELL1\n";
856$        CONVEY:      ,F_CELL1;
27$         STATION,     TO_CELL4;
949$        TRACE,       -1,"-Arrived to station TO_CELL4\n";
969$        STORE:       TO_CELL4_S1;
970$        DELAY:       0.;
955$        UNSTORE:     ;
950$        DELAY:       0.000:NEXT(23$);

23$         TRACE,       -1,"-Splitting group representative\n";
971$        SPLIT:       A(*),M,NS,IS:NEXT(13$);

13$         DELAY:       0.00;
1002$       DELAY:       0.;
975$        TRACE,       -1,"-Transferred to station CELL_4\n";
1000$       ROUTE:       2,CELL_4;
28$         STATION,     TO_CELL3;
1020$       TRACE,       -1,"-Arrived to station TO_CELL3\n";
1040$       STORE:       TO_CELL3_S1;
1041$       DELAY:       0.;
1026$       UNSTORE:     ;
1021$       DELAY:       0.000:NEXT(24$);

24$         TRACE,       -1,"-Splitting group representative\n";
1042$       SPLIT:       A(*),M,NS,IS:NEXT(16$);

16$         DELAY:       0.00;
1050$       TRACE,       -1,"-Delay for loading time 1\n";
1073$       DELAY:       1;
1046$       TRACE,       -1,"-Transferred to station CELL_3\n";
1071$       ROUTE:       1,CELL_3;
29$         STATION,     TO_CELL2;
1091$       TRACE,       -1,"-Arrived to station TO_CELL2\n";
1111$       STORE:       TO_CELL2_S1;
1112$       DELAY:       0.;
1097$       UNSTORE:     ;
1092$       DELAY:       0.000:NEXT(25$);

25$         TRACE,       -1,"-Splitting group representative\n";
1113$       SPLIT:       A(*),M,NS,IS:NEXT(19$);

19$         DELAY:       0.00;
1124$       TRACE,       -1,"-Waiting for conveyor CONV_CELL21\n";
1115$       QUEUE,       TO_CELL2_Q;
1146$       ACCESS:      CONV_CELL21,1;
1121$       TRACE,       -1,"-Delay for loading time 1\n";
1144$       DELAY:       1;
1117$       TRACE,       -1,"-Transferred to station CELL_2\n";
1136$       CONVEY:      ,CELL_2;
30$         STATION,     TO_CELL1;
1162$       TRACE,       -1,"-Arrived to station TO_CELL1\n";
1182$       STORE:       TO_CELL1_S1;
1183$       DELAY:       0.;
1168$       UNSTORE:     ;
1163$       DELAY:       0.000:NEXT(26$);

26$         TRACE,       -1,"-Splitting group representative\n";
1184$       SPLIT:       A(*),M,NS,IS:NEXT(20$);

20$         DELAY:       0.00;
1195$       TRACE,       -1,"-Waiting for conveyor CONV_CELL11\n";
1186$       QUEUE,       TO_CELL1_Q;
1217$       ACCESS:      CONV_CELL11,1;
1192$       TRACE,       -1,"-Delay for loading time 1\n";
1215$       DELAY:       1;
1188$       TRACE,       -1,"-Transferred to station CELL_1\n";
1207$       CONVEY:      ,CELL_1;
38$         STATION,     ENT_BATCHING;
1233$       TRACE,       -1,"-Arrived to station ENT_BATCHING\n";
1253$       STORE:       ENT_BATCHING_S1;
1254$       DELAY:       0.;
1239$       UNSTORE:     ;
1234$       DELAY:       0.000:NEXT(44$);

44$         TRACE,       -1,"-Updating counter EMAGED_SHAFTS(BATCH TYPE) \n";
1255$       COUNT:       EMAGED_SHAFTS(BATCH TYPE),1:NEXT(35$);
```

```
35$             TRACE,          -1,"-Updating counter SHAFTS COMP_BY_EMAG \n";
1258$           COUNT:          SHAFTS COMP_BY_EMAG,1:NEXT(34$);

34$             DELAY:          0.00;
1261$           TRACE,          -1,"-Waiting for batch size of AINT(EXPO(40))+30\n";
1269$           GROUP,          :AINT(EXPO(40))+30,Last;
1263$           TRACE,          -1,"-Entity created to represent temporary batch of size
AINT(EXPO(40))+30\n";
1267$           DELAY:          0.000:NEXT(32$);

32$             TRACE,          -1,"-Selecting Min of 4 stations\n";
1271$           FINDJ,          1,4:Min(NE(ST PICK PS(1,J))+NQ(ST PICK PS(2,J)));
1270$           ASSIGN:         ST CHOICE=ST PICK PS(1,J):NEXT(31$);

31$             DELAY:          0.00;
1286$           ASSIGN:         M=ENT_BATCHING;
1302$           DELAY:          0.;
1275$           TRACE,          -1,"-Transferred to station ST CHOICE\n";
1300$           ROUTE:          10,ST CHOICE;
46$             STATION,        EMAG OPERATOR;
1343$           TRACE,          -1,"-Arrived to station EMAG OPERATOR\n";
1463$           STORE:          EMAG OPERATOR_S1;
1464$           DELAY:          0.;
1448$           UNSTORE:        ;
1336$           TRACE,          -1,"-Waiting for resource EMAG OPERATOR_R\n";
1445$           QUEUE,          EMAG OPERATOR_R_Q;
1442$           SEIZE,          1:EMAG OPERATOR_R,1;
1370$           STORE:          EMAG OPERATOR_S3;
                TRACE,          -1,"-Delay for processing time TRIA(.13,.15,.22)\n";
1429$           DELAY:          TRIA(.13,.15,.22);
1406$           UNSTORE:        ;
1399$           STORE:          EMAG OPERATOR_S4;
1333$           TRACE,          -1,"-Releasing resource\n";
1422$           RELEASE:        EMAG OPERATOR_R,1;
1386$           DELAY:          0.;
1383$           UNSTORE:        ;
1327$           TRACE,          -1,"-Transferred to station ENT_BATCHING\n";
1384$           ROUTE:          .2,ENT_BATCHING;
1537$           CREATE,         1,0.0;
1536$           ASSIGN:         EMAG ROBOT_N=EMAG ROBOT_N+1:
                                NS=EMAG ROBOT_N;
EMAG ROBOT_Home SCAN:           LT(EMAG ROBOT,NS)<>EMAG;
1535$           ALLOCATE,       9999:EMAG ROBOT(NS);
1534$           MOVE,           :,EMAG;
1533$           FREE:           :NEXT(EMAG ROBOT_Home);

53$             STATION,        E_EMAG;
1544$           TRACE,          -1,"-Arrived to station E_EMAG\n";
1564$           STORE:          E_EMAG_S1;
1565$           DELAY:          0.;
1550$           UNSTORE:        ;
1545$           DELAY:          0.000:NEXT(52$);

52$             TRACE,          -1,"-Choosing from 2 options\n";
1566$           BRANCH,         1:If,BATCH TYPE == OBT,50$,Yes:
                                Else,50$,Yes;
50$             TRACE,          -1,"-Making assignments\n";
1567$           ASSIGN:         EMAG ST=TRIA(1,2,2.3):
                                EMAG_OT=TRIA(2,3,4):NEXT(36$);

36$             TRACE,          -1,"-Updating counter # OF EMAG SETUPS \n";
1569$           COUNT:          # OF EMAG SETUPS,1:NEXT(41$);

41$             DELAY:          0.00;
1570$           TRACE,          -1,"-Waiting for resources\n";
1575$           SEIZE,          :EMAG_R,1;
1573$           DELAY:          0.000:NEXT(51$);

51$             TRACE,          -1,"-Delaying for time EMAG ST\n";
1578$           DELAY:          EMAG ST:NEXT(39$);

39$             TRACE,          -1,"-Releasing resources\n";
1579$           RELEASE:        EMAG_R,1:NEXT(48$);

48$             STATION,        EMAG;
1609$           TRACE,          -1,"-Arrived to station EMAG\n";
1730$           DELAY:          0.;
1603$           TRACE,          -1,"-Exiting conveyor\n";
1718$           EXIT:           ;
1602$           TRACE,          -1,"-Waiting for resource EMAG_R\n";
```

```
1711$          QUEUE,        EMAG_R_Q;
1708$          SEIZE,        1:EMAG_R,1;
1586$          TRACE,        -1,"-Releasing overlapping resource\n";
1702$          RELEASE:      EMAG_Q_SPACE,1;
1636$          STORE:        EMAG_S3;
               TRACE,        -1,"-Delay for processing time EMAG_OT\n";
1695$          DELAY:        EMAG_OT;
1672$          UNSTORE:      ;
1665$          STORE:        EMAG_S4;
1599$          TRACE,        -1,"-Releasing resource\n";
1688$          RELEASE:      EMAG_R,1;
1596$          TRACE,        -1,"-Delay for loading time .15\n";
1652$          DELAY:        .15;
1649$          UNSTORE:      ;
1593$          TRACE,        -1,"-Transferred to station EMAG OPERATOR\n";
1650$          ROUTE:        .3,EMAG OPERATOR;
1796$          CREATE,       1,0.0;
1795$          ASSIGN:       ASRST_N=ASRST_N+1:
                             NS=ASRST_N;
ASRST_Home     SCAN:         LT(ASRST,NS)<>ASRS;
1794$          ALLOCATE,     9999:ASRST(NS);
1793$          MOVE,         :,ASRS;
1792$          FREE:         :NEXT(ASRST_Home);

68$            STATION,      E_SAWING;
1803$          TRACE,        -1,"-Arrived to station E_SAWING\n";
1824$          DELAY:        0.;
1798$          TRACE,        -1,"-Freeing transporter\n";
1818$          FREE:         ;
1804$          DELAY:        0.000:NEXT(67$);

67$            TRACE,        -1,"-Choosing from 2 options\n";
1825$          BRANCH,       1:If,BATCH TYPE==OBT,69$,Yes:
                             Else,55$,Yes;
69$            STATION,      SAWING;
1855$          TRACE,        -1,"-Arrived to station SAWING\n";
1975$          STORE:        SAWING_S1;
1976$          DELAY:        0.;
1960$          UNSTORE:      ;
1848$          TRACE,        -1,"-Waiting for resource SAWING_R\n";
1957$          QUEUE,        SAWING_R_Q;
1954$          SEIZE,        1:SAWING_R,1;
1882$          STORE:        SAWING_S3;
               TRACE,        -1,"-Delay for processing time SAWING_OT\n";
1941$          DELAY:        SAWING_OT;
1918$          UNSTORE:      ;
1911$          STORE:        SAWING_S4;
1856$          ASSIGN:       STATE(SAWING_R)=BLOCKED;
1833$          TRACE,        -1,"-Waiting for resource EMAG_Q_SPACE\n";
1938$          SEIZE,        1:EMAG_Q_SPACE,1;
1845$          TRACE,        -1,"-Releasing resource\n";
1934$          RELEASE:      SAWING_R,1;
1898$          DELAY:        0.;
1895$          UNSTORE:      ;
1838$          TRACE,        -1,"-Transferred to next module\n";
1891$          DELAY:        0.00:NEXT(66$);

66$            TRACE,        -1,"-Making assignments\n";
1977$          ASSIGN:       JJ=JJ+1:
                             BATCHC=BATCHC+1:NEXT(65$);

65$            TRACE,        -1,"-Choosing from 4 options\n";
1978$          BRANCH,       1:If,BATCH NUMBER ==0,58$,Yes:
                             If,JJ==SHAFT_PER_BAR.AND.BATCHC<BATCH SIZE.AND.(BATCH
NUMBER>0),64$,Yes:
                             If,JJ==SHAFT_PER_BAR.AND.BATCHC==BATCH SIZE.AND.BATCH
NUMBER>0,60$,Yes:
                             Else,56$,Yes;
58$            TRACE,        -1,"-Making assignments\n";
1979$          ASSIGN:       JJ=0:
                             BATCHC=0:
                             BATCH SIZE=AINT(EXPO(40))+1:
                             BATCH TYPE=AINT(UNIF(1.3,10.3)):
                             SHAFT_PER_BAR=AINT(UNIF(7,10))+1:
                             ARRIVAL TIME=TNOW:
                             BATCH NUMBER=BATCH NUMBER+1:
                             H=H+1:NEXT(10$);

10$            TRACE,        -1,"-Making assignments\n";
1980$          ASSIGN:       BATCH_SIZE(H)=BATCH SIZE:
```

```
                         BATCH_TYPE(H)=BATCH TYPE:NEXT(59$);

59$        TRACE,        -1,"-Updating counter BTCHES PER TYPE(BATCH TYPE) \n";
1981$      COUNT:        BTCHES PER TYPE(BATCH TYPE),1:NEXT(61$);

61$        TRACE,        -1,"-Updating counter STEEL BARS USED \n";
1984$      COUNT:        STEEL BARS USED,1:NEXT(71$);

71$        STATION,      ASRS;
2014$      TRACE,        -1,"-Arrived to station ASRS\n";
2134$      STORE:        ASRS_S1;
2135$      DELAY:        0.;
2119$      UNSTORE:      ;
2007$      TRACE,        -1,"-Waiting for resource ASRS_R\n";
2116$      QUEUE,        ASRS_R_Q;
2113$      SEIZE,        1:ASRS_R,1;
2041$      STORE:        ASRS_S3;
           TRACE,        -1,"-Delay for processing time NORM(1.10,.2,1)\n";
2100$      DELAY:        NORM(1.10,.2,1);
2077$      UNSTORE:      ;
2070$      STORE:        ASRS_S4;
2004$      TRACE,        -1,"-Releasing resource\n";
2093$      RELEASE:      ASRS_R,1;
1995$      TRACE,        -1,"-Waiting for transporter ASRST\n";
2060$      REQUEST,      1:ASRST(CYC);
2001$      TRACE,        -1,"-Delay for loading time .2\n";
2057$      DELAY:        .2;
2054$      UNSTORE:      ;
1998$      TRACE,        -1,"-Transferred to station E_SAWING\n";
2052$      TRANSPORT,    :,E_SAWING;
64$        TRACE,        -1,"-Duplicating entities\n";
2136$      DUPLICATE:    1,62$:NEXT(45$);

45$        TRACE,        -1,"-Updating counter SAWED SHAFTS(BATCH TYPE) \n";
2137$      COUNT:        SAWED SHAFTS(BATCH TYPE),1:NEXT(43$);

43$        TRACE,        -1,"-Making assignments\n";
2139$      ASSIGN:       BATCH_TYPE_PASS_SAW=BATCH TYPE:NEXT(63$);

63$        DELAY:        0.00;
2154$      ASSIGN:       M=SAWING;
2150$      TRACE,        -1,"-Waiting for conveyor CONV_SAW_EMAG\n";
2172$      ACCESS:       CONV_SAW_EMAG,1;
2170$      DELAY:        0.;
2180$      ASSIGN:       Picture=SAWED SHAFT;
2143$      TRACE,        -1,"-Transferred to station E_EMAG\n";
2162$      CONVEY:       ,E_EMAG;
62$        TRACE,        -1,"-Making assignments\n";
2182$      ASSIGN:       JJ=0:
                         ARRIVAL TIME=TNOW:NEXT(61$);

60$        TRACE,        -1,"-Duplicating entities\n";
2183$      DUPLICATE:    1,58$:NEXT(45$);

56$        TRACE,        -1,"-Choosing from 2 options\n";
2184$      BRANCH,       1:If,BATCHC==BATCH SIZE,60$,Yes:
                         Else,57$,Yes;
57$        TRACE,        -1,"-Duplicating entities\n";
2185$      DUPLICATE:    1,69$:NEXT(45$);

55$        TRACE,        -1,"-Making assignments\n";
2186$      ASSIGN:       SAWING ST=EXPO(3,1):
                         SAWING_OT=TRIA(.45,1,1.5):NEXT(54$);

54$        TRACE,        -1,"-Delaying for time SAWING ST\n";
2187$      DELAY:        SAWING ST:NEXT(42$);

42$        TRACE,        -1,"-Making assignments\n";
2188$      ASSIGN:       OBT=BATCH TYPE:NEXT(37$);

37$        TRACE,        -1,"-Updating counter # SAW SETUPS \n";
2190$      COUNT:        # SAW SETUPS,1:NEXT(69$);

2215$      CREATE,       1:1,1;
2214$      ASSIGN:       BATCH TYPE=1:
                         BATCH SIZE=1:
                         OBT=0:
                         BATCH NUMBER=0:
                         JJ=0:
                         SAWING ST=5:
```

```
                                   SAWING_OT=1;
2255$          STATION,            ARRIVAL BARS;
2206$          TRACE,              -1,"-Arrived to system at station ARRIVAL BARS\n";
2252$          ASSIGN:             Picture=Default;
2231$          DELAY:              0.;
2201$          TRACE,              -1,"-Transferred to next module\n";
2225$          DELAY:              0.00:NEXT(71$);
```

E.6. Experiment Listing for Shaft Production (Integrated Production Unit)

```
PROJECT,        ;

ATTRIBUTES:     BATCHC:
                ARRIVAL TIME:
                BATCH SIZE:
                BATCH TYPE:
                SHAFT_PER_BAR:
                BATCH NUMBER:
                OBT:
                SAWING_OT:
                SAWING ST:
                EMAG ST:
                EMAG_OT:
                BATCH_TYPE_PASS_SAW:
                H:
                ST CHOICE;

SCHEDULES:      SCH1,1*894,0*306;

STORAGES:       ASRS_S4:
                ASRS_S3:
                ASRS_S1:
                SAWING_S4:
                SAWING_S3:
                SAWING_S1:
                E_EMAG_S1:
                EMAG_S4:
                EMAG_S3:
                EMAG_OPERATOR_S4:
                EMAG OPERATOR_S3:
                EMAG OPERATOR_S1:
                ENT_BATCHING_S1:
                TO_CELL1_S1:
                TO_CELL2_S1:
                TO_CELL3_S1:
                TO_CELL4_S1:
                CELL_1_S4:
                CELL_1_S3:
                CELL_1_S1:
                CELL_2_S4:
                CELL_2_S3:
                CELL_2_S1:
                CELL_3_S4:
                CELL_3_S3:
                CELL_3_S1:
                CELL_4_S4:
                CELL_4_S3:
                CELL_4_S1:
                CELL4_BATCH_S1:
                TO_CELL3_FROM4_S1;

VARIABLES:      ASRST_N:
                JJ:
                EMAG ROBOT_N:
                BATCH_SIZE(1000),0:
                BATCH_TYPE(1000),0;

QUEUES:         ASRS_R_Q,FIFO:
                SAWING_R_Q,FIFO:
                EMAG_R_Q,FIFO:
                EMAG_OPERATOR_R_Q,FIFO:
                EMAG_Q_SPACE_Q,FIFO:
                CELL_1_R_Q,FIFO:
                CELL_2_R_Q,FIFO:
                CELL_3_R_Q,FIFO:
```

```
                CELL_4_R_Q,FIFO:
                TO_CELL1_Q,FIFO:
                TO_CELL2_Q,FIFO;

PICTURES:       Default:
                SAWED SHAFT;

STATESETS:      SERVER STATES,BLOCKED;

RESOURCES:      ASRS_R,Capacity(1,):
                SAWING_R,Capacity(1,),SERVER STATES:
                EMAG_R,Capacity(1,):
                EMAG OPERATOR_R,Capacity(1,):
                EMAG_Q_SPACE,Capacity(18,):
                CELL_1_R,Schedule(SCH1,Ignore):
                CELL_2_R,Schedule(SCH1,Ignore):
                CELL_3_R,Schedule(SCH1,Ignore):
                CELL_4_R,Schedule(SCH1,Ignore);

STATIONS:       ASRS:
                SAWING:
                ARRIVAL BARS:
                E_SAWING:
                E_EMAG:
                EMAG:
                EMAG OPERATOR:
                ENT_BATCHING:
                TO_CELL1:
                TO_CELL2:
                TO_CELL3:
                TO_CELL4:
                CELL_1:
                F_CELL1:
                CELL_2:
                F_CELL2:
                CELL_3:
                F_CELL3:
                CELL_4:
                CELL4_BATCH:
                TO_CELL3_FROM4;

DISTANCES:      ASRST_Dst,ASRS-E_SAWING-10:
                EMAG ROBOT_Dst,EMAG-EMAG OPERATOR-3;

TRANSPORTERS:   ASRST,1,DISTANCE(ASRST_Dst),1.:
                EMAG ROBOT,1,DISTANCE(EMAG ROBOT_Dst),1.---,()-Active;

SEGMENTS:       CONV_SAW_EMAG_Seg,SAWING,E_EMAG-6:
                CONV_CELL11_Seg,TO_CELL1,CELL_1-25,
                F_CELL1-25:
                CONV_CELL21_Seg,TO_CELL2,CELL_2-25,
                F_CELL2-25;

CONVEYORS:      CONV_SAW_EMAG,CONV_SAW_EMAG_Seg,1.,1,Active,1,Accumulating,.2:
                CONV_CELL11,CONV_CELL11_Seg,1.,1,Active,1,Accumulating:
                CONV_CELL21,CONV_CELL21_Seg,1.,1,Active,1,Accumulating;

COUNTERS:       STEEL BARS USED:
                # BATCHES OF TYPE 1:
                # BATCHES OF TYPE 2:
                # BATCHES OF TYPE 3:
                # BATCHES OF TYPE 4:
                # BATCHES OF TYPE 5:
                # BATCHES OF TYPE 6:
                # BATCHES OF TYPE 7:
                # BATCHES OF TYPE 8:
                # BATCHES OF TYPE 9:
                # BATCHES OF TYPE 10:
                SAWED_SHAFTS_TYPE 1:
                SAWED_SHAFTS_TYPE 2:
                SAWED_SHAFTS_TYPE 3:
                SAWED_SHAFTS_TYPE 4:
                SAWED_SHAFTS_TYPE 5:
                SAWED_SHAFTS_TYPE 6:
                SAWED_SHAFTS_TYPE 7:
                SAWED_SHAFTS_TYPE 8:
                SAWED_SHAFTS_TYPE 9:
                SAWED_SHAFTS_TYPE 10:
                PRM_M_SHAFT TYPE 1:
                PRM_M_SHAFT TYPE 2:
```

```
                PRM_M_SHAFT TYPE 3:
                PRM_M_SHAFT TYPE 4:
                PRM_M_SHAFT TYPE 5:
                PRM_M_SHAFT TYPE 6:
                PRM_M_SHAFT TYPE 7:
                PRM_M_SHAFT TYPE 8:
                PRM_M_SHAFT TYPE 9:
                PRM_M_SHAFT TYPE 10:
                # SAW SETUPS:
                # OF EMAG SETUPS:
                SHAFTS COMP_BY_EMAG:
                F_CELL1_C:
                F_CELL2_C:
                F_CELL3_C;

TALLIES:        F_CELL1_Ta:
                F_CELL2_Ta:
                F_CELL3_Ta;

DSTATS:         MR(ASRS_R),ASRS_R Available:
                NR(ASRS_R),ASRS_R Busy:
                NQ(ASRS_R_Q),# in ASRS_R_Q:
                NQ(SAWING_R_Q),# in SAWING_R_Q:
                MT(ASRST),ASRST Active:
                NT(ASRST),ASRST Busy:
                MR(EMAG_R),EMAG_R Available:
                NR(EMAG_R),EMAG_R Busy:
                NQ(EMAG_R_Q),# in EMAG_R_Q:
                MR(EMAG OPERATOR_R),EMAG OPERATOR_R Available:
                NR(EMAG OPERATOR_R),EMAG OPERATOR_R Busy:
                NQ(EMAG OPERATOR_R_Q),# in EMAG OPERATOR_R_Q:
                MR(EMAG_Q_SPACE),EMAG_Q_SPACE Available:
                NR(EMAG_Q_SPACE),EMAG_Q_SPACE Busy:
                NQ(EMAG_Q_SPACE_Q),# in EMAG_Q_SPACE_Q:
                MR(CELL_1_R),CELL_1_R Available:
                NR(CELL_1_R),CELL_1_R Busy:
                MR(CELL_2_R),CELL_2_R Available:
                NR(CELL_2_R),CELL_2_R Busy:
                NQ(CELL_2_R_Q),# in CELL_2_R_Q:
                MR(CELL_3_R),CELL_3_R Available:
                NR(CELL_3_R),CELL_3_R Busy:
                NQ(CELL_3_R_Q),# in CELL_3_R_Q:
                MR(CELL_4_R),CELL_4_R Available:
                NR(CELL_4_R),CELL_4_R Busy:
                NQ(CELL_4_R_Q),# in CELL_4_R_Q:
                NQ(TO_CELL1_Q),# in TO_CELL1_Q:
                NQ(TO_CELL2_Q),# in TO_CELL2_Q:
                NQ(CELL_1_R_Q),# in CELL_1_R_Q;

FREQUENCIES:    STATE(SAWING_R);

REPLICATE,      7,0.0,6000,Yes,Yes;

EXPRESSIONS:  ST PICK
PS(4,4),TO_CELL1,TO_CELL1_Q,,0,TO_CELL2,TO_CELL2_Q,,0,TO_CELL3,CELL_3_R_Q,,0,TO_CELL4,CE
LL_4_R_Q,,0;

SETS:           BTCHES PER TYPE,# BATCHES OF TYPE 1,# BATCHES OF TYPE 2,# BATCHES OF TYPE
3,# BATCHES OF TYPE 4,
                # BATCHES OF TYPE 5,# BATCHES OF TYPE 6,# BATCHES OF TYPE 7,# BATCHES OF
TYPE 8,# BATCHES OF TYPE 9,
                # BATCHES OF TYPE 10:
                SAWED SHAFTS,SAWED_SHAFTS_TYPE 1,SAWED_SHAFTS_TYPE 2,SAWED_SHAFTS_TYPE
3,SAWED_SHAFTS_TYPE 4,SAWED_SHAFTS_TYPE 5,
                SAWED_SHAFTS_TYPE 6,SAWED_SHAFTS_TYPE 7,SAWED_SHAFTS_TYPE
8,SAWED_SHAFTS_TYPE 9,SAWED_SHAFTS_TYPE 10:
                EMAGED_SHAFTS,PRM_M_SHAFT TYPE 1,PRM_M_SHAFT TYPE 2,PRM_M_SHAFT TYPE
3,PRM_M_SHAFT TYPE 4,PRM_M_SHAFT TYPE 5,
                PRM_M_SHAFT TYPE 6,PRM_M_SHAFT TYPE 7,PRM_M_SHAFT TYPE 8,PRM_M_SHAFT TYPE
9,PRM_M_SHAFT TYPE 10;
```

E.7. Model Listing for Rotor Assembly (Line-1)

```
3$          STATION,      E_SCALING STATION_2;
65$         TRACE,        -1,"-Arrived to station E_SCALING STATION_2\n";
85$         STORE:        E_SCALING STATION_2_S1;
```

```
86$          DELAY:        0.;
71$          UNSTORE:      ;
66$          DELAY:        0.000:NEXT(2$);

2$           TRACE,        -1,"-Splitting group representative\n";
87$          SPLIT:        A(*),M,NS,IS:NEXT(0$);

0$           STATION,      SCALING STATION_2;
117$         TRACE,        -1,"-Arrived to station SCALING STATION_2\n";
237$         STORE:        SCALING STATION_2_S1;
238$         DELAY:        0.;
222$         UNSTORE:      ;
110$         TRACE,        -1,"-Waiting for resource SCALING STATION_2_S\n";
219$         QUEUE,        SCALING STATION_2_S_Q;
215$         SEIZE,        1:SELECT(SCALING STATION_2_S,CYC,),1;
144$         STORE:        SCALING STATION_2_S3;
             TRACE,        -1,"-Delay for processing time TRIA(2,3,4)\n";
203$         DELAY:        TRIA(2,3,4);
180$         UNSTORE:      ;
173$         STORE:        SCALING STATION_2_S4;
107$         TRACE,        -1,"-Releasing resource\n";
195$         RELEASE:      SELECT(SCALING STATION_2_S,,),1;
160$         DELAY:        0.;
157$         UNSTORE:      ;
101$         TRACE,        -1,"-Transferred to station FINISHED R_ASSEMBLIES\n";
158$         ROUTE:        .3,FINISHED R_ASSEMBLIES;
4$           STATION,      FINISHED R_ASSEMBLIES;
246$         TRACE,        -1,"-Arrived to station FINISHED R_ASSEMBLIES\n";
276$         DELAY:        0.;
253$         COUNT:        FINISHED ASSEMBLES(ASSEMBLY TYPE),1;
248$         TALLY:        TIME IN SYS(ASSEMBLY TYPE),Interval(TIME IN),1;
239$         TRACE,        -1,"-Disposing entity\n";
247$         DISPOSE;
8$           STATION,      E_SCALING STATION_1;
297$         TRACE,        -1,"-Arrived to station E_SCALING STATION_1\n";
317$         STORE:        E_SCALING STATION_1_S1;
318$         DELAY:        0.;
303$         UNSTORE:      ;
298$         DELAY:        0.000:NEXT(7$);

7$           TRACE,        -1,"-Splitting group representative\n";
319$         SPLIT:        A(*),M,NS,IS:NEXT(5$);

5$           STATION,      SCALING STATION_1;
349$         TRACE,        -1,"-Arrived to station SCALING STATION_1\n";
469$         STORE:        SCALING STATION_1_S1;
470$         DELAY:        0.;
454$         UNSTORE:      ;
342$         TRACE,        -1,"-Waiting for resource SCALING STATION_1_S\n";
451$         QUEUE,        SCALING STATION_1_S_Q;
447$         SEIZE,        1:SELECT(SCALING STATION_1_S,CYC,),1;
376$         STORE:        SCALING STATION_1_S3;
             TRACE,        -1,"-Delay for processing time TRIA(1.5,1.8,2.2)\n";
435$         DELAY:        TRIA(1.5,1.8,2.2);
412$         UNSTORE:      ;
405$         STORE:        SCALING STATION_1_S4;
339$         TRACE,        -1,"-Releasing resource\n";
427$         RELEASE:      SELECT(SCALING STATION_1_S,,),1;
392$         DELAY:        0.;
389$         UNSTORE:      ;
333$         TRACE,        -1,"-Transferred to station FINISHED R_ASSEMBLIES\n";
390$         ROUTE:        .5,FINISHED R_ASSEMBLIES;
14$          STATION,      E_TURNING STATION_2;
477$         TRACE,        -1,"-Arrived to station E_TURNING STATION_2\n";
497$         STORE:        E_TURNING STATION_2_S1;
498$         DELAY:        0.;
483$         UNSTORE:      ;
478$         DELAY:        0.000:NEXT(13$);

13$          TRACE,        -1,"-Splitting group representative\n";
499$         SPLIT:        A(*),M,NS,IS:NEXT(33$);

33$          STATION,      TURNING STATION_2;
529$         TRACE,        -1,"-Arrived to station TURNING STATION_2\n";
649$         STORE:        TURNING STATION_2_S1;
650$         DELAY:        0.;
634$         UNSTORE:      ;
522$         TRACE,        -1,"-Waiting for resource TURNING STATION_2_S\n";
631$         QUEUE,        TURNING STATION_2_S_Q;
627$         SEIZE,        1:SELECT(TURNING STATION_2_S,LNB,),1;
```

```
556$          STORE:        TURNING STATION_2_S3;
              TRACE,        -1,"-Delay for processing time TRIA(1.5,2,3)\n";
615$          DELAY:        TRIA(1.5,2,3);
592$          UNSTORE:      ;
598$          ASSIGN:       Picture=E3;
585$          STORE:        TURNING STATION_2_S4;
519$          TRACE,        -1,"-Releasing resource\n";
607$          RELEASE:      SELECT(TURNING STATION_2_S,,),1;
572$          DELAY:        0.;
569$          UNSTORE:      ;
512$          TRACE,        -1,"-Transferred to next module\n";
565$          DELAY:        0.00:NEXT(18$);

18$           DELAY:        0.00;
653$          TRACE,        -1,"-Waiting for batch size of 20\n";
661$          GROUP,        :20,Last;
655$          TRACE,        -1,"-Entity created to represent temporary batch of size
20\n";
659$          DELAY:        0.000:NEXT(10$);

10$           DELAY:        0.00;
676$          ASSIGN:       M=TURNING STATION_2;
692$          DELAY:        0.;
665$          TRACE,        -1,"-Transferred to station E_SCALING STATION_2\n";
690$          ROUTE:        2,E_SCALING STATION_2;
16$           STATION,      E_TURNING STATION_1;
710$          TRACE,        -1,"-Arrived to station E_TURNING STATION_1\n";
730$          STORE:        E_TURNING STATION_1_S1;
731$          DELAY:        0.;
716$          UNSTORE:      ;
711$          DELAY:        0.000:NEXT(15$);

15$           TRACE,        -1,"-Splitting group representative\n";
732$          SPLIT:        A(*),M,NS,IS:NEXT(35$);

35$           STATION,      TURNING STATION_1;
762$          TRACE,        -1,"-Arrived to station TURNING STATION_1\n";
882$          STORE:        TURNING STATION_1_S1;
883$          DELAY:        0.;
867$          UNSTORE:      ;
755$          TRACE,        -1,"-Waiting for resource TURNING STATION_1_S\n";
864$          QUEUE,        TURNING STATION_1_S_Q;
860$          SEIZE,        1:SELECT(TURNING STATION_1_S,LNB,),1;
789$          STORE:        TURNING STATION_1_S3;
              TRACE,        -1,"-Delay for processing time TRIA(2,3,3.2)\n";
848$          DELAY:        TRIA(2,3,3.2);
825$          UNSTORE:      ;
831$          ASSIGN:       Picture=E3;
818$          STORE:        TURNING STATION_1_S4;
752$          TRACE,        -1,"-Releasing resource\n";
840$          RELEASE:      SELECT(TURNING STATION_1_S,,),1;
805$          DELAY:        0.;
802$          UNSTORE:      ;
745$          TRACE,        -1,"-Transferred to next module\n";
798$          DELAY:        0.00:NEXT(12$);

12$           DELAY:        0.00;
886$          TRACE,        -1,"-Waiting for batch size of 20\n";
894$          GROUP,        :20,Last;
888$          TRACE,        -1,"-Entity created to represent temporary batch of size
20\n";
892$          DELAY:        0.000:NEXT(9$);

9$            DELAY:        0.00;
909$          ASSIGN:       M=TURNING STATION_1;
925$          DELAY:        0.;
898$          TRACE,        -1,"-Transferred to station E_SCALING STATION_1\n";
923$          ROUTE:        0.,E_SCALING STATION_1;
27$           STATION,      WIP AREA_2;
971$          TRACE,        -1,"-Arrived to station WIP AREA_2\n";
992$          DELAY:        0.;
965$          TRACE,        -1,"-Exiting conveyor\n";
981$          EXIT:         ;
972$          DELAY:        0.000:NEXT(21$);

21$           TRACE,        -1,"-Waiting for batch size of 20\n";
994$          QUEUE,        WIP_Q2;
1003$         GROUP,        :20,Last;
997$          TRACE,        -1,"-Entity created to represent temporary batch of size
20\n";
```

```
1001$           DELAY:          0.000:NEXT(25$);

25$             TRACE,          -1,"-Delaying for time 180\n";
1004$           DELAY:          180:NEXT(20$);

20$             DELAY:          0.00;
1019$           ASSIGN:         M=WIP AREA_2;
1035$           DELAY:          0.;
1008$           TRACE,          -1,"-Transferred to station E_TURNING STATION_2\n";
1033$           ROUTE:          1,E_TURNING STATION_2;
28$             STATION,        WIP AREA_1;
1053$           TRACE,          -1,"-Arrived to station WIP AREA_1\n";
1074$           DELAY:          0.;
1047$           TRACE,          -1,"-Exiting conveyor\n";
1063$           EXIT:           ;
1054$           DELAY:          0.000:NEXT(23$);

23$             TRACE,          -1,"-Waiting for batch size of 20\n";
1076$           QUEUE,          WIP_Q1;
1085$           GROUP,          :20,Last;
1079$           TRACE,          -1,"-Entity created to represent temporary batch of size
20\n";
1083$           DELAY:          0.000:NEXT(26$);

26$             TRACE,          -1,"-Delaying for time 180\n";
1086$           DELAY:          180:NEXT(19$);

19$             DELAY:          0.00;
1101$           ASSIGN:         M=WIP AREA_1;
1117$           DELAY:          0.;
1090$           TRACE,          -1,"-Transferred to station E_TURNING STATION_1\n";
1115$           ROUTE:          1,E_TURNING STATION_1;
31$             STATION,        E_SHAFT FITTING_2;
1135$           TRACE,          -1,"-Arrived to station E_SHAFT FITTING_2\n";
1156$           DELAY:          0.;
1129$           TRACE,          -1,"-Exiting conveyor\n";
1145$           EXIT:           ;
1136$           DELAY:          0.000:NEXT(29$);

29$             DELAY:          0.00;
1171$           ASSIGN:         M=E_SHAFT FITTING_2;
1167$           TRACE,          -1,"-Waiting for conveyor CONV22\n";
1189$           ACCESS:         CONV22,1;
1187$           DELAY:          0.;
1160$           TRACE,          -1,"-Transferred to station SHAFT FITTING_2\n";
1179$           CONVEY:         ,SHAFT FITTING_2;
32$             STATION,        E_SHAFT FITTING_1;
1205$           TRACE,          -1,"-Arrived to station E_SHAFT FITTING_1\n";
1226$           DELAY:          0.;
1199$           TRACE,          -1,"-Exiting conveyor\n";
1215$           EXIT:           ;
1206$           DELAY:          0.000:NEXT(30$);

30$             DELAY:          0.00;
1241$           ASSIGN:         M=E_SHAFT FITTING_1;
1237$           TRACE,          -1,"-Waiting for conveyor CONV11\n";
1259$           ACCESS:         CONV11,1;
1257$           DELAY:          0.;
1230$           TRACE,          -1,"-Transferred to station SHAFT FITTING_1\n";
1249$           CONVEY:         ,SHAFT FITTING_1;
1293$           CREATE,         1:1:MARK(TIME IN);
1292$           ASSIGN:         ASSEMBLY TYPE=2;
1333$           STATION,        ARRIVAL S_ROTORS;
1284$           TRACE,          -1,"-Arrived to system at station ARRIVAL S_ROTORS\n";
1330$           ASSIGN:         Picture=E1;
1274$           TRACE,          -1,"-Waiting for conveyor CONV2\n";
1311$           ACCESS:         CONV2,1;
1309$           DELAY:          0.;
1278$           TRACE,          -1,"-Transferred to station E_SHAFT FITTING_2\n";
1302$           CONVEY:         ,E_SHAFT FITTING_2;
37$             STATION,        COOLING STATION_2;
1363$           TRACE,          -1,"-Arrived to station COOLING STATION_2\n";
1483$           STORE:          COOLING STATION_2_S1;
1484$           DELAY:          0.;
1468$           UNSTORE:        ;
1356$           TRACE,          -1,"-Waiting for resource COOLING STATION_2_R\n";
1465$           QUEUE,          COOLING STATION_2_R_Q;
1462$           SEIZE,          1:COOLING STATION_2_R,1;
1390$           STORE:          COOLING STATION_2_S3;
                TRACE,          -1,"-Delay for processing time TRIA(.2,.25,.28)\n";
```

```
1449$          DELAY:          TRIA(.2,.25,.28);
1426$          UNSTORE:        ;
1419$          STORE:          COOLING STATION_2_S4;
1353$          TRACE,          -1,"-Releasing resource\n";
1442$          RELEASE:        COOLING STATION_2_R,1;
1406$          DELAY:          0.;
1403$          UNSTORE:        ;
1347$          TRACE,          -1,"-Transferred to station WIP AREA_2\n";
1398$          CONVEY:         ,WIP AREA_2;
39$            STATION,        COOLING STATION_1;
1514$          TRACE,          -1,"-Arrived to station COOLING STATION_1\n";
1634$          STORE:          COOLING STATION_1_S1;
1635$          DELAY:          0.;
1619$          UNSTORE:        ;
1507$          TRACE,          -1,"-Waiting for resource COOLING STATION_1_R\n";
1616$          QUEUE,          COOLING STATION_1_R_Q;
1613$          SEIZE,          1:COOLING STATION_1_R,1;
1541$          STORE:          COOLING STATION_1_S3;
               TRACE,          -1,"-Delay for processing time TRIA(.25,.26,.28)\n";
1600$          DELAY:          TRIA(.25,.26,.28);
1577$          UNSTORE:        ;
1570$          STORE:          COOLING STATION_1_S4;
1504$          TRACE,          -1,"-Releasing resource\n";
1593$          RELEASE:        COOLING STATION_1_R,1;
1557$          DELAY:          0.;
1554$          UNSTORE:        ;
1498$          TRACE,          -1,"-Transferred to station WIP AREA_1\n";
1549$          CONVEY:         ,WIP AREA_1;
41$            STATION,        SHAFT FITTING_2;
1665$          TRACE,          -1,"-Arrived to station SHAFT FITTING_2\n";
1785$          STORE:          SHAFT FITTING_2_S1;
1786$          DELAY:          0.;
1770$          UNSTORE:        ;
1658$          TRACE,          -1,"-Waiting for resource SHAFT FITTING_2_R\n";
1767$          QUEUE,          SHAFT FITTING_2_R_Q;
1764$          SEIZE,          1:SHAFT FITTING_2_R,1;
1692$          STORE:          SHAFT FITTING_2_S3;
               TRACE,          -1,"-Delay for processing time TRIA(.14,.22,.3)\n";
1751$          DELAY:          TRIA(.14,.22,.3);
1728$          UNSTORE:        ;
1734$          ASSIGN:         Picture=E2;
1721$          STORE:          SHAFT FITTING_2_S4;
1655$          TRACE,          -1,"-Releasing resource\n";
1744$          RELEASE:        SHAFT FITTING_2_R,1;
1708$          DELAY:          0.;
1705$          UNSTORE:        ;
1649$          TRACE,          -1,"-Transferred to station COOLING STATION_2\n";
1700$          CONVEY:         ,COOLING STATION_2;
43$            STATION,        SHAFT FITTING_1;
1816$          TRACE,          -1,"-Arrived to station SHAFT FITTING_1\n";
1936$          STORE:          SHAFT FITTING_1_S1;
1937$          DELAY:          0.;
1921$          UNSTORE:        ;
1809$          TRACE,          -1,"-Waiting for resource SHAFT FITTING_1_R\n";
1918$          QUEUE,          SHAFT FITTING_1_R_Q;
1915$          SEIZE,          1:SHAFT FITTING_1_R,1;
1843$          STORE:          SHAFT FITTING_1_S3;
               TRACE,          -1,"-Delay for processing time TRIA(.16,.2,.3)\n";
1902$          DELAY:          TRIA(.16,.2,.3);
1879$          UNSTORE:        ;
1885$          ASSIGN:         Picture=E2;
1872$          STORE:          SHAFT FITTING_1_S4;
1806$          TRACE,          -1,"-Releasing resource\n";
1895$          RELEASE:        SHAFT FITTING_1_R,1;
1859$          DELAY:          0.;
1856$          UNSTORE:        ;
1800$          TRACE,          -1,"-Transferred to station COOLING STATION_1\n";
1851$          CONVEY:         ,COOLING STATION_1;
1962$          CREATE,         1:1:MARK(TIME IN);
1961$          ASSIGN:         ASSEMBLY TYPE=1;
2002$          STATION,        ARRIVAL L_ROTORS;
1953$          TRACE,          -1,"-Arrived to system at station ARRIVAL L_ROTORS\n";
1999$          ASSIGN:         Picture=E1;
1943$          TRACE,          -1,"-Waiting for conveyor CONV1\n";
1980$          ACCESS:         CONV1,1;
1978$          DELAY:          0.;
1947$          TRACE,          -1,"-Transferred to station E_SHAFT FITTING_1\n";
1971$          CONVEY:         ,E_SHAFT FITTING_1;
```

E.8. Experiment Listing for Rotor Assembly (Line-1)

```
PROJECT,        ,AL_AMARI;

ATTRIBUTES:     TIME IN:
                ASSEMBLY TYPE;

STORAGES:       SHAFT FITTING_1_S4:
                SHAFT FITTING_1_S3:
                SHAFT FITTING_1_S1:
                SHAFT FITTING_2_S4:
                SHAFT FITTING_2_S3:
                SHAFT FITTING_2_S1:
                COOLING STATION_1_S4:
                COOLING STATION_1_S3:
                COOLING STATION_1_S1:
                COOLING STATION_2_S4:
                COOLING STATION_2_S3:
                COOLING STATION_2_S1:
                TURNING STATION_1_S4:
                TURNING STATION_1_S3:
                TURNING STATION_1_S1:
                TURNING STATION_2_S4:
                TURNING STATION_2_S3:
                TURNING STATION_2_S1:
                E_TURNING STATION_1_S1:
                E_TURNING STATION_2_S1:
                E_SCALING STATION_1_S1:
                SCALING STATION_1_S4:
                SCALING STATION_1_S3:
                SCALING STATION_1_S1:
                E_SCALING STATION_2_S1:
                SCALING STATION_2_S4:
                SCALING STATION_2_S3:
                SCALING STATION_2_S1;

QUEUES:         SHAFT FITTING_1_R_Q,FIFO:
                SHAFT FITTING_2_R_Q,FIFO:
                COOLING STATION_1_R_Q,FIFO:
                COOLING STATION_2_R_Q,FIFO:
                TURNING STATION_1_S_Q,FIFO:
                TURNING_MACHINE_11_Q,FIFO:
                TURNING_MACHINE_12_Q,FIFO:
                TURNING STATION_2_S_Q,FIFO:
                TURNING_MACHINE_21_Q,FIFO:
                TURNING_MACHINE_22_Q,FIFO:
                SCALING STATION_1_S_Q,FIFO:
                SCALE_11_Q,FIFO:
                SCALE_12_Q,FIFO:
                SCALE_21_Q,FIFO:
                SCALE_22_Q,FIFO:
                SCALING STATION_2_S_Q,FIFO:
                WIP_Q1,FIFO:
                WIP_Q2,FIFO;

PICTURES:       Default:
                E1:
                E2:
                E3;

RESOURCES:      SHAFT FITTING_1_R,Capacity(1,):
                SHAFT FITTING_2_R,Capacity(1,):
                COOLING STATION_1_R,Capacity(1,):
                COOLING STATION_2_R,Capacity(1,):
                TURNING_MACHINE_11,Capacity(1,):
                TURNING_MACHINE_12,Capacity(1,):
                TURNING_MACHINE_21,Capacity(1,):
                TURNING_MACHINE_22,Capacity(1,):
                SCALE_11,Capacity(1,):
                SCALE_12,Capacity(1,):
                SCALE_21,Capacity(1,):
                SCALE_22,Capacity(1,);

STATIONS:       SHAFT FITTING_1:
                SHAFT FITTING_2:
                COOLING STATION_1:
                COOLING STATION_2:
```

```
                ARRIVAL L_ROTORS:
                ARRIVAL S_ROTORS:
                E_SHAFT FITTING_2:
                E_SHAFT FITTING_1:
                WIP AREA_1:
                WIP AREA_2:
                TURNING STATION_1:
                TURNING STATION_2:
                E_TURNING STATION_1:
                E_TURNING STATION_2:
                E_SCALING STATION_2:
                E_SCALING STATION_1:
                SCALING STATION_1:
                SCALING STATION_2:
                FINISHED R_ASSEMBLIES;

SEGMENTS:       CONV2_Seg,ARRIVAL S_ROTORS,E_SHAFT FITTING_2-60:
                CONV1_Seg,ARRIVAL L_ROTORS,E_SHAFT FITTING_1-48:
                CONV11_Seg,E_SHAFT FITTING_1,SHAFT FITTING_1-1,
                COOLING STATION_1-1,
                WIP AREA_1-1:
                CONV22_Seg,E_SHAFT FITTING_2,SHAFT FITTING_2-1,
                COOLING STATION_2-1,
                WIP AREA_2-1;

CONVEYORS:      CONV1,CONV1_Seg,.4,1,Active,1,Accumulating:
                CONV2,CONV2_Seg,.5,1,Active,1,Accumulating:
                CONV11,CONV11_Seg,1.,1,Active,1,Accumulating:
                CONV22,CONV22_Seg,1.,1,Active,1,Accumulating;

COUNTERS:       LARGE_ROTORS,,Replicate,"C:\BH\SIM\ROASS\CL.001":
                SMALL_ROTORS,,Replicate,"C:\BH\SIM\ROASS\CS.001";

TALLIES:        LARGE ASS IN SYS,"C:\BH\SIM\ROASS\TALL1.001":
                SMALL ASS IN SYS,"C:\BH\SIM\ROASS\TALLs.001";

DSTATS:         MR(SHAFT FITTING_1_R),SHAFT FITTING_1_R Available:
                NR(SHAFT FITTING_1_R),SHAFT FITTING_1_R Busy:
                NQ(SHAFT FITTING_1_R_Q),# in SHAFT FITTING_1_R_Q:
                MR(SHAFT FITTING_2_R),SHAFT FITTING_2_R Available:
                NR(SHAFT FITTING_2_R),SHAFT FITTING_2_R Busy:
                NQ(SHAFT FITTING_2_R_Q),# in SHAFT FITTING_2_R_Q:
                MR(COOLING STATION_1_R),COOLING STATION_1_R Available:
                NR(COOLING STATION_1_R),COOLING STATION_1_R Busy:
                NQ(COOLING STATION_1_R_Q),# in COOLING STATION_1_R_Q:
                MR(COOLING STATION_2_R),COOLING STATION_2_R Available:
                NR(COOLING STATION_2_R),COOLING STATION_2_R Busy:
                NQ(COOLING STATION_2_R_Q),# in COOLING STATION_2_R_Q:
                NQ(TURNING STATION_1_S_Q),# in TURNING STATION_1_S_Q:
                MR(TURNING_MACHINE_11),TURNING_MACHINE_11 Available:
                NR(TURNING_MACHINE_11),TURNING_MACHINE_11 Busy:
                NQ(TURNING_MACHINE_11_Q),# in TURNING_MACHINE_11_Q:
                MR(TURNING_MACHINE_12),TURNING_MACHINE_12 Available:
                NR(TURNING_MACHINE_12),TURNING_MACHINE_12 Busy:
                NQ(TURNING_MACHINE_12_Q),# in TURNING_MACHINE_12_Q:
                NQ(TURNING STATION_2_S_Q),# in TURNING STATION_2_S_Q:
                MR(TURNING_MACHINE_21),TURNING_MACHINE_21 Available:
                NR(TURNING_MACHINE_21),TURNING_MACHINE_21 Busy:
                NQ(TURNING_MACHINE_21_Q),# in TURNING_MACHINE_21_Q:
                MR(TURNING_MACHINE_22),TURNING_MACHINE_22 Available:
                NR(TURNING_MACHINE_22),TURNING_MACHINE_22 Busy:
                NQ(TURNING_MACHINE_22_Q),# in TURNING_MACHINE_22_Q:
                NQ(SCALING STATION_1_S_Q),# in SCALING STATION_1_S_Q:
                MR(SCALE_11),SCALE_11 Available:
                NR(SCALE_11),SCALE_11 Busy:
                NQ(SCALE_11_Q),# in SCALE_11_Q:
                MR(SCALE_12),SCALE_12 Available:
                NR(SCALE_12),SCALE_12 Busy:
                NQ(SCALE_12_Q),# in SCALE_12_Q:
                MR(SCALE_21),SCALE_21 Available:
                NR(SCALE_21),SCALE_21 Busy:
                NQ(SCALE_21_Q),# in SCALE_21_Q:
                MR(SCALE_22),SCALE_22 Available:
                NR(SCALE_22),SCALE_22 Busy:
                NQ(SCALE_22_Q),# in SCALE_22_Q:
                NQ(SCALING STATION_2_S_Q),# in SCALING STATION_2_S_Q:
                NQ(WIP_Q1),# in WIP_Q1:
                NQ(WIP_Q2),# in WIP_Q2:
                NEC(CONV1)+NEC(CONV2),,"C:\BH\SIM\ROASS\COVOVN.001";
```

```
REPLICATE,      5,0.0,2850,No,Yes;

SETS:           TURNING STATION_1_S,TURNING_MACHINE_11,TURNING_MACHINE_12:
                TURNING STATION_2_S,TURNING_MACHINE_21,TURNING_MACHINE_22:
                SCALING STATION_1_S,SCALE_11,SCALE_12:
                SCALING STATION_2_S,SCALE_21,SCALE_22:
                EPIC,E1,Default,E2:
                FINISHED ASSEMBLES,LARGE_ROTORS,SMALL_ROTORS:
                TIME IN SYS,LARGE ASS IN SYS,SMALL ASS IN SYS;
```

E.9. Model Listing for Rotor Assembly (Cell-100)

```
1$              STATION,        FINISHED ROTOR ASS;
31$             TRACE,          -1,"-Arrived to station FINISHED ROTOR ASS\n";
61$             DELAY:          0.;
39$             COUNT:          FINISHED ROTOR ASS_C,1;
34$             TALLY:          FINISHED ROTOR ASS_Ta,Interval(TIME IN),1;
24$             TRACE,          -1,"-Disposing entity\n";
32$             DISPOSE;
3$              STATION,        ET_SCALING;
134$            TRACE,          -1,"-Arrived to station ET_SCALING\n";
155$            DELAY:          0.;
129$            TRACE,          -1,"-Freeing transporter\n";
149$            FREE:           ;
135$            DELAY:          0.000:NEXT(2$);

2$              TRACE,          -1,"-Splitting group representative\n";
156$            SPLIT:          A(*),M,NS,IS:NEXT(4$);

4$              STATION,        SCALING;
186$            TRACE,          -1,"-Arrived to station SCALING\n";
306$            STORE:          SCALING_S1;
307$            DELAY:          0.;
291$            UNSTORE:        ;
178$            TRACE,          -1,"-Waiting for transporter SCALE OPERATOR\n";
288$            QUEUE,          SCALE OPERATOR_Q;
282$            REQUEST,        1:SCALE OPERATOR(CYC);
213$            STORE:          SCALING_S3;
                TRACE,          -1,"-Delay for processing time TRIA(1.37,1.93,2.1)\n";
272$            DELAY:          TRIA(1.37,1.93,2.1);
249$            UNSTORE:        ;
242$            STORE:          SCALING_S4;
175$            TRACE,          -1,"-Freeing transporter\n";
262$            FREE:           ;
229$            DELAY:          0.;
226$            UNSTORE:        ;
169$            TRACE,          -1,"-Transferred to next module\n";
222$            DELAY:          0.00:NEXT(0$);

0$              DELAY:          0.00;
322$            ASSIGN:         M=SCALING;
338$            DELAY:          0.;
311$            TRACE,          -1,"-Transferred to station FINISHED ROTOR ASS\n";
336$            ROUTE:          .1,FINISHED ROTOR ASS;
9$              STATION,        WIP_AREA2;
356$            TRACE,          -1,"-Arrived to station WIP_AREA2\n";
377$            DELAY:          0.;
351$            TRACE,          -1,"-Freeing transporter\n";
371$            FREE:           ;
357$            DELAY:          0.000:NEXT(8$);

8$              DELAY:          0.00;
380$            TRACE,          -1,"-Waiting for batch size of 20\n";
388$            GROUP,          :20,Last;
382$            TRACE,          -1,"-Entity created to represent temporary batch of size
20\n";
386$            DELAY:          0.000:NEXT(6$);

6$              DELAY:          0.00;
403$            ASSIGN:         M=WIP_AREA2;
398$            TRACE,          -1,"-Waiting for transporter SCALE OPERATOR\n";
422$            REQUEST,        1:SCALE OPERATOR(POR);
419$            DELAY:          0.;
392$            TRACE,          -1,"-Transferred to station ET_SCALING\n";
414$            TRANSPORT,      :,ET_SCALING;
10$             STATION,        TURNING;
```

```
460$          TRACE,         -1,"-Arrived to station TURNING\n";
581$          DELAY:         0.;
455$          TRACE,         -1,"-Freeing transporter\n";
575$          FREE:          ;
453$          TRACE,         -1,"-Waiting for resource TURNING_R\n";
562$          QUEUE,         TURNING_R_Q;
559$          SEIZE,         1:TURNING_R,1;
487$          STORE:         TURNING_S3;
              TRACE,         -1,"-Delay for processing time TRIA(.5,.8,1)\n";
546$          DELAY:         TRIA(.5,.8,1);
523$          UNSTORE:       ;
516$          STORE:         TURNING_S4;
450$          TRACE,         -1,"-Releasing resource\n";
539$          RELEASE:       TURNING_R,1;
441$          TRACE,         -1,"-Waiting for transporter OPERATOR\n";
506$          REQUEST,       1:OPERATOR(POR);
503$          DELAY:         0.;
500$          UNSTORE:       ;
444$          TRACE,         -1,"-Transferred to station WIP_AREA2\n";
498$          TRANSPORT,     :,WIP_AREA2;
15$           STATION,       WIP_AREA;
588$          TRACE,         -1,"-Arrived to station WIP_AREA\n";
609$          DELAY:         0.;
583$          TRACE,         -1,"-Freeing transporter\n";
603$          FREE:          ;
589$          DELAY:         0.000:NEXT(14$);

14$           TRACE,         -1,"-Splitting group representative\n";
610$          SPLIT:         A(*),M,NS,IS:NEXT(13$);

13$           TRACE,         -1,"-Delaying for time 80\n";
611$          DELAY:         80:NEXT(12$);

12$           DELAY:         0.00;
626$          ASSIGN:        M=WIP_AREA;
621$          TRACE,         -1,"-Waiting for transporter OPERATOR\n";
645$          REQUEST,       1:OPERATOR(POR);
642$          DELAY:         0.;
615$          TRACE,         -1,"-Transferred to station TURNING\n";
637$          TRANSPORT,     :,TURNING;
16$           STATION,       COOLING;
683$          TRACE,         -1,"-Arrived to station COOLING\n";
804$          DELAY:         0.;
678$          TRACE,         -1,"-Freeing transporter\n";
798$          FREE:          ;
676$          TRACE,         -1,"-Waiting for resource COOLING_R\n";
785$          QUEUE,         COOLING_R_Q;
782$          SEIZE,         1:COOLING_R,1;
710$          STORE:         COOLING_S3;
              TRACE,         -1,"-Delay for processing time TRIA(1,1.2,1.3)\n";
769$          DELAY:         TRIA(1,1.2,1.3);
746$          UNSTORE:       ;
739$          STORE:         COOLING_S4;
673$          TRACE,         -1,"-Releasing resource\n";
762$          RELEASE:       COOLING_R,1;
664$          TRACE,         -1,"-Waiting for transporter OPERATOR\n";
729$          REQUEST,       1:OPERATOR(POR);
726$          DELAY:         0.;
723$          UNSTORE:       ;
667$          TRACE,         -1,"-Transferred to station WIP_AREA\n";
721$          TRANSPORT,     :,WIP_AREA;
21$           STATION,       ET_COOLING;
811$          TRACE,         -1,"-Arrived to station ET_COOLING\n";
832$          DELAY:         0.;
805$          TRACE,         -1,"-Exiting conveyor\n";
821$          EXIT:          ;
812$          DELAY:         0.000:NEXT(19$);

19$           DELAY:         0.00;
835$          TRACE,         -1,"-Waiting for batch size of 2\n";
843$          GROUP,         :2,Last;
837$          TRACE,         -1,"-Entity created to represent temporary batch of size
2\n";
841$          DELAY:         0.000:NEXT(20$);

20$           DELAY:         0.00;
858$          ASSIGN:        M=ET_COOLING;
853$          TRACE,         -1,"-Waiting for transporter OPERATOR\n";
845$          QUEUE,         OPERATOR_Q;
877$          REQUEST,       1:OPERATOR(POR);
```

```
874$          DELAY:         0.;
847$          TRACE,         -1,"-Transferred to station COOLING\n";
869$          TRANSPORT,     :,COOLING;
23$           STATION,       E_OVEN;
892$          TRACE,         -1,"-Arrived to station E_OVEN\n";
913$          DELAY:         0.;
887$          TRACE,         -1,"-Freeing transporter\n";
907$          FREE:          ;
893$          DELAY:         0.000:NEXT(22$);

22$           DELAY:         0.00;
928$          ASSIGN:        M=E_OVEN;
924$          TRACE,         -1,"-Waiting for conveyor CONV1\n";
946$          ACCESS:        CONV1,1;
944$          DELAY:         0.;
917$          TRACE,         -1,"-Transferred to station ET_COOLING\n";
936$          CONVEY:        ,ET_COOLING;
980$          CREATE,        3:EXPO(10):MARK(TIME IN);
1020$         STATION,       ARRIVAL ROTORS;
971$          TRACE,         -1,"-Arrived to system at station ARRIVAL ROTORS\n";
1017$         ASSIGN:        Picture=Default;
962$          TRACE,         -1,"-Waiting for transporter OPERATOR\n";
999$          REQUEST,       1:OPERATOR(POR);
996$          DELAY:         0.;
965$          TRACE,         -1,"-Transferred to station E_OVEN\n";
992$          TRANSPORT,     :,E_OVEN;
```

E.10. Experiment Listing for Rotor Assembly (Cell-100)

```
PROJECT,       ;

ATTRIBUTES:    TIME IN;

STORAGES:      COOLING_S4:
               COOLING_S3:
               TURNING_S4:
               TURNING_S3:
               SCALING_S4:
               SCALING_S3:
               SCALING_S1;

QUEUES:        OPERATOR_Q:
               COOLING_R_Q,FIFO:
               TURNING_R_Q,FIFO:
               SCALE OPERATOR_Q,FIFO;

PICTURES:      Default;

RESOURCES:     COOLING_R,Capacity(1,):
               TURNING_R,Capacity(1,);

STATIONS:      ARRIVAL ROTORS:
               E_OVEN:
               ET_COOLING:
               COOLING:
               WIP_AREA:
               TURNING:
               WIP_AREA2:
               SCALING:
               FINISHED ROTOR ASS:
               ET_SCALING;

DISTANCES:     OPERATOR_Dst,ARRIVAL ROTORS-E_OVEN-5,
               ET_COOLING-COOLING-1,
               COOLING-WIP_AREA-4,
               WIP_AREA-TURNING-3,
               TURNING-WIP_AREA2-3,
               WIP_AREA2-WIP_AREA-1,
               COOLING-TURNING-4,
               COOLING-ARRIVAL ROTORS-2,
               TURNING-ARRIVAL ROTORS-6,
               TURNING-ET_COOLING-6,
               E_OVEN-ET_COOLING-5,
               WIP_AREA-ARRIVAL ROTORS-3,
               WIP_AREA2-ARRIVAL ROTORS-5,
               E_OVEN-COOLING-1,
```

```
              WIP_AREA2-ET_COOLING-4,
              E_OVEN-WIP_AREA-5,
              ET_COOLING-WIP_AREA-3,
              COOLING-WIP_AREA2-4,
              E_OVEN-TURNING-4:
              SCALE OPERATOR_Dst,WIP_AREA2-ET_SCALING-5,
              ET_SCALING-SCALING-1,
              SCALING-WIP_AREA2-5;

TRANSPORTERS: OPERATOR,1,DISTANCE(OPERATOR_Dst),30:
              SCALE OPERATOR,1,DISTANCE(SCALE OPERATOR_Dst),16;

SEGMENTS:     CONV1_Seg,E_OVEN,ET_COOLING-12;

CONVEYORS:    CONV1,CONV1_Seg,.6,1,Active,1,Accumulating;

COUNTERS:     FINISHED ROTOR ASS_C,,Replicate,"C:\BH\SIM\ROASS\CC100.001";

TALLIES:      FINISHED ROTOR ASS_Ta,"C:\BH\SIM\ROASS\TL100.001";

DSTATS:       MT(OPERATOR),OPERATOR Active:
              NT(OPERATOR),OPERATOR Busy:
              MR(COOLING_R),COOLING_R Available:
              NR(COOLING_R),COOLING_R Busy:
              NQ(COOLING_R_Q),# in COOLING_R_Q:
              MR(TURNING_R),TURNING_R Available:
              NR(TURNING_R),TURNING_R Busy:
              NQ(TURNING_R_Q),# in TURNING_R_Q:
              MT(SCALE OPERATOR),SCALE OPERATOR Active:
              NT(SCALE OPERATOR),SCALE OPERATOR Busy:
              NQ(SCALE OPERATOR_Q),# in SCALE OPERATOR_Q:
              NEC(CONV1),,"C:\BH\SIM\ROASS\COV100.001";

REPLICATE,    5,0.0,2850,Yes,Yes;
```

E.11. Model Listing for Rotor Assembly (Cell-112)

```
1$            STATION,      FINISHED ROTOR ASS;
31$           TRACE,        -1,"-Arrived to station FINISHED ROTOR ASS\n";
61$           DELAY:        0.;
39$           COUNT:        FINISHED ROTOR ASS_C,1;
34$           TALLY:        FINISHED ROTOR ASS_Ta,Interval(TIME IN),1;
24$           TRACE,        -1,"-Disposing entity\n";
32$           DISPOSE;
3$            STATION,      ET_SCALING;
134$          TRACE,        -1,"-Arrived to station ET_SCALING\n";
155$          DELAY:        0.;
129$          TRACE,        -1,"-Freeing transporter\n";
149$          FREE:         ;
135$          DELAY:        0.000:NEXT(2$);

2$            TRACE,        -1,"-Splitting group representative\n";
156$          SPLIT:        A(*),M,NS,IS:NEXT(4$);

4$            STATION,      SCALING;
186$          TRACE,        -1,"-Arrived to station SCALING\n";
306$          STORE:        SCALING_S1;
307$          DELAY:        0.;
291$          UNSTORE:      ;
178$          TRACE,        -1,"-Waiting for transporter SCALE OPERATOR\n";
288$          QUEUE,        SCALE OPERATOR_Q;
282$          REQUEST,      1:SCALE OPERATOR(CYC);
213$          STORE:        SCALING_S3;
              TRACE,        -1,"-Delay for processing time TRIA(1.37,1.93,2.1)\n";
272$          DELAY:        TRIA(1.37,1.93,2.1);
249$          UNSTORE:      ;
242$          STORE:        SCALING_S4;
175$          TRACE,        -1,"-Freeing transporter\n";
262$          FREE:         ;
229$          DELAY:        0.;
226$          UNSTORE:      ;
169$          TRACE,        -1,"-Transferred to next module\n";
222$          DELAY:        0.00:NEXT(0$);

0$            DELAY:        0.00;
322$          ASSIGN:       M=SCALING;
```

```
338$          DELAY:          0.;
311$          TRACE,          -1,"-Transferred to station FINISHED ROTOR ASS\n";
336$          ROUTE:          .1,FINISHED ROTOR ASS;
9$            STATION,        WIP_AREA2;
356$          TRACE,          -1,"-Arrived to station WIP_AREA2\n";
377$          DELAY:          0.;
351$          TRACE,          -1,"-Freeing transporter\n";
371$          FREE:           ;
357$          DELAY:          0.000:NEXT(8$);

8$            DELAY:          0.00;
380$          TRACE,          -1,"-Waiting for batch size of 20\n";
388$          GROUP,          :20,Last;
382$          TRACE,          -1,"-Entity created to represent temporary batch of size
20\n";
386$          DELAY:          0.000:NEXT(6$);

6$            DELAY:          0.00;
403$          ASSIGN:         M=WIP_AREA2;
398$          TRACE,          -1,"-Waiting for transporter SCALE OPERATOR\n";
422$          REQUEST,        1:SCALE OPERATOR(POR);
419$          DELAY:          0.;
392$          TRACE,          -1,"-Transferred to station ET_SCALING\n";
414$          TRANSPORT,      :,ET_SCALING;
10$           STATION,        TURNING;
460$          TRACE,          -1,"-Arrived to station TURNING\n";
581$          DELAY:          0.;
455$          TRACE,          -1,"-Freeing transporter\n";
575$          FREE:           ;
453$          TRACE,          -1,"-Waiting for resource TURNING_R\n";
562$          QUEUE,          TURNING_R_Q;
559$          SEIZE,          1:TURNING_R,1;
487$          STORE:          TURNING_S3;
              TRACE,          -1,"-Delay for processing time TRIA(1,1.5,2)\n";
546$          DELAY:          TRIA(1,1.5,2);
523$          UNSTORE:        ;
516$          STORE:          TURNING_S4;
450$          TRACE,          -1,"-Releasing resource\n";
539$          RELEASE:        TURNING_R,1;
441$          TRACE,          -1,"-Waiting for transporter OPERATOR\n";
506$          REQUEST,        1:OPERATOR(POR);
503$          DELAY:          0.;
500$          UNSTORE:        ;
444$          TRACE,          -1,"-Transferred to station WIP_AREA2\n";
498$          TRANSPORT,      :,WIP_AREA2;
15$           STATION,        WIP_AREA;
588$          TRACE,          -1,"-Arrived to station WIP_AREA\n";
609$          DELAY:          0.;
583$          TRACE,          -1,"-Freeing transporter\n";
603$          FREE:           ;
589$          DELAY:          0.000:NEXT(14$);

14$           TRACE,          -1,"-Splitting group representative\n";
610$          SPLIT:          A(*),M,NS,IS:NEXT(13$);

13$           TRACE,          -1,"-Delaying for time 90\n";
611$          DELAY:          90:NEXT(12$);

12$           DELAY:          0.00;
626$          ASSIGN:         M=WIP_AREA;
621$          TRACE,          -1,"-Waiting for transporter OPERATOR\n";
645$          REQUEST,        1:OPERATOR(POR);
642$          DELAY:          0.;
615$          TRACE,          -1,"-Transferred to station TURNING\n";
637$          TRANSPORT,      :,TURNING;
16$           STATION,        COOLING;
683$          TRACE,          -1,"-Arrived to station COOLING\n";
804$          DELAY:          0.;
678$          TRACE,          -1,"-Freeing transporter\n";
798$          FREE:           ;
676$          TRACE,          -1,"-Waiting for resource COOLING_R\n";
785$          QUEUE,          COOLING_R_Q;
782$          SEIZE,          1:COOLING_R,1;
710$          STORE:          COOLING_S3;
              TRACE,          -1,"-Delay for processing time TRIA(1.2,1.33,1.4)\n";
769$          DELAY:          TRIA(1.2,1.33,1.4);
746$          UNSTORE:        ;
739$          STORE:          COOLING_S4;
673$          TRACE,          -1,"-Releasing resource\n";
762$          RELEASE:        COOLING_R,1;
```

```
664$            TRACE,          -1,"-Waiting for transporter OPERATOR\n";
729$            REQUEST,        1:OPERATOR(POR);
726$            DELAY:          0.;
723$            UNSTORE:        ;
667$            TRACE,          -1,"-Transferred to station WIP_AREA\n";
721$            TRANSPORT,      :,WIP_AREA;
21$             STATION,        ET_COOLING;
811$            TRACE,          -1,"-Arrived to station ET_COOLING\n";
832$            DELAY:          0.;
805$            TRACE,          -1,"-Exiting conveyor\n";
821$            EXIT:           ;
812$            DELAY:          0.000:NEXT(19$);

19$             DELAY:          0.00;
835$            TRACE,          -1,"-Waiting for batch size of 2\n";
843$            GROUP,          :2,Last;
837$            TRACE,          -1,"-Entity created to represent temporary batch of size
2\n";
841$            DELAY:          0.000:NEXT(20$);

20$             DELAY:          0.00;
858$            ASSIGN:         M=ET_COOLING;
853$            TRACE,          -1,"-Waiting for transporter OPERATOR\n";
845$            QUEUE,          OPERATOR_Q;
877$            REQUEST,        1:OPERATOR(POR);
874$            DELAY:          0.;
847$            TRACE,          -1,"-Transferred to station COOLING\n";
869$            TRANSPORT,      :,COOLING;
23$             STATION,        E_OVEN;
892$            TRACE,          -1,"-Arrived to station E_OVEN\n";
913$            DELAY:          0.;
887$            TRACE,          -1,"-Freeing transporter\n";
907$            FREE:           ;
893$            DELAY:          0.000:NEXT(22$);

22$             DELAY:          0.00;
928$            ASSIGN:         M=E_OVEN;
924$            TRACE,          -1,"-Waiting for conveyor CONV1\n";
946$            ACCESS:         CONV1,1;
944$            DELAY:          0.;
917$            TRACE,          -1,"-Transferred to station ET_COOLING\n";
936$            CONVEY:         ,ET_COOLING;
980$            CREATE,         3:EXPO(10):MARK(TIME IN);
1020$           STATION,        ARRIVAL ROTORS;
971$            TRACE,          -1,"-Arrived to system at station ARRIVAL ROTORS\n";
1017$           ASSIGN:         Picture=Default;
962$            TRACE,          -1,"-Waiting for transporter OPERATOR\n";
999$            REQUEST,        1:OPERATOR(POR);
996$            DELAY:          0.;
965$            TRACE,          -1,"-Transferred to station E_OVEN\n";
992$            TRANSPORT,      :,E_OVEN;
```

E.12. Experiment Listing for Rotor Assembly (Cell-112)

```
PROJECT,        ;

ATTRIBUTES:     TIME IN;

STORAGES:       COOLING_S4:
                COOLING_S3:
                TURNING_S4:
                TURNING_S3:
                SCALING_S4:
                SCALING_S3:
                SCALING_S1;

QUEUES:         OPERATOR_Q:
                COOLING_R_Q,FIFO:
                TURNING_R_Q,FIFO:
                SCALE OPERATOR_Q,FIFO;

PICTURES:       Default;

RESOURCES:      COOLING_R,Capacity(1,):
                TURNING_R,Capacity(1,);
```

```
STATIONS:      ARRIVAL ROTORS:
               E_OVEN:
               ET_COOLING:
               COOLING:
               WIP_AREA:
               TURNING:
               WIP_AREA2:
               SCALING:
               FINISHED ROTOR ASS:
               ET_SCALING;

DISTANCES:     OPERATOR_Dst,ARRIVAL ROTORS-E_OVEN-5,
               ET_COOLING-COOLING-1,
               COOLING-WIP_AREA-4,
               WIP_AREA-TURNING-3,
               TURNING-WIP_AREA2-3,
               WIP_AREA2-WIP_AREA-1,
               COOLING-TURNING-4,
               COOLING-ARRIVAL ROTORS-2,
               TURNING-ARRIVAL ROTORS-6,
               TURNING-ET_COOLING-6,
               E_OVEN-ET_COOLING-5,
               WIP_AREA-ARRIVAL ROTORS-3,
               WIP_AREA2-ARRIVAL ROTORS-5,
               E_OVEN-COOLING-1,
               WIP_AREA2-ET_COOLING-4,
               E_OVEN-WIP_AREA-5,
               ET_COOLING-WIP_AREA-3,
               COOLING-WIP_AREA2-4,
               E_OVEN-TURNING-4:
               SCALE OPERATOR_Dst,WIP_AREA2-ET_SCALING-5,
               ET_SCALING-SCALING-1,
               SCALING-WIP_AREA2-5;

TRANSPORTERS:  OPERATOR,1,DISTANCE(OPERATOR_Dst),25:
               SCALE OPERATOR,1,DISTANCE(SCALE OPERATOR_Dst),16;

SEGMENTS:      CONV1_Seg,E_OVEN,ET_COOLING-12;

CONVEYORS:     CONV1,CONV1_Seg,.6,1,Active,1,Accumulating;

COUNTERS:      FINISHED ROTOR ASS_C,,Replicate,"C:\BH\SIM\ROASS\CC112.001";

TALLIES:       FINISHED ROTOR ASS_Ta,"C:\BH\SIM\ROASS\TL112.001";

DSTATS:        MT(OPERATOR),OPERATOR Active:
               NT(OPERATOR),OPERATOR Busy:
               MR(COOLING_R),COOLING_R Available:
               NR(COOLING_R),COOLING_R Busy:
               NQ(COOLING_R_Q),# in COOLING_R_Q:
               MR(TURNING_R),TURNING_R Available:
               NR(TURNING_R),TURNING_R Busy:
               NQ(TURNING_R_Q),# in TURNING_R_Q:
               MT(SCALE OPERATOR),SCALE OPERATOR Active:
               NT(SCALE OPERATOR),SCALE OPERATOR Busy:
               NQ(SCALE OPERATOR_Q),# in SCALE OPERATOR_Q:
               NEC(CONV1),,"C:\BH\SIM\ROASS\COV112.001";

REPLICATE,     5,0.0,2850,Yes,Yes;
```

E.13. Model Listing for Windings Assembly (Cell-160/180)

```
0$          STATION,      F_WOUND PACK;
53$         TRACE,        -1,"-Arrived to station F_WOUND PACK\n";
83$         DELAY:        0.;
61$         COUNT:        F_WOUND PACK_C,1;
56$         TALLY:        F_WOUND PACK_Ta,Interval(TIME IN),1;
46$         TRACE,        -1,"-Disposing entity\n";
54$         DISPOSE;
1$          STATION,      WYVERTIC;
127$        TRACE,        -1,"-Arrived to station WYVERTIC\n";
247$        STORE:        WYVERTIC_S1;
248$        DELAY:        0.;
232$        UNSTORE:      ;
120$        TRACE,        -1,"-Waiting for resource WYVERTIC MACHINES\n";
```

```
229$            QUEUE,          WYVERTIC MACHINES_Q;
225$            SEIZE,          1:SELECT(WYVERTIC MACHINES,CYC,),1;
154$            STORE:          WYVERTIC_S3;
                TRACE,          -1,"-Delay for processing time TRIA(2.20,3,3.6)\n";
213$            DELAY:          TRIA(2.20,3,3.6);
190$            UNSTORE:        ;
183$            STORE:          WYVERTIC_S4;
117$            TRACE,          -1,"-Releasing resource\n";
205$            RELEASE:        SELECT(WYVERTIC MACHINES,,),1;
170$            DELAY:          0.;
167$            UNSTORE:        ;
111$            TRACE,          -1,"-Transferred to station F_WOUND PACK\n";
168$            ROUTE:          1,F_WOUND PACK;
5$              STATION,        ENT_COOLING;
255$            TRACE,          -1,"-Arrived to station ENT_COOLING\n";
253$            TRACE,          -1,"-Delay for unloading time .1\n";
276$            DELAY:          .1;
249$            TRACE,          -1,"-Exiting conveyor\n";
265$            EXIT:           ;
256$            DELAY:          0.000:NEXT(4$);

4$              TRACE,          -1,"-Splitting group representative\n";
277$            SPLIT:          A(*),M,NS,IS:NEXT(3$);

3$              DELAY:          0.00;
292$            ASSIGN:         M=ENT_COOLING;
308$            DELAY:          0.;
281$            TRACE,          -1,"-Transferred to station WYVERTIC\n";
306$            ROUTE:          2,WYVERTIC;
9$              STATION,        ENT_OVEN;
326$            TRACE,          -1,"-Arrived to station ENT_OVEN\n";
346$            STORE:          ENT_OVEN_S1;
347$            DELAY:          0.;
332$            UNSTORE:        ;
327$            DELAY:          0.000:NEXT(8$);

8$              DELAY:          0.00;
350$            TRACE,          -1,"-Waiting for batch size of 4\n";
358$            GROUP,          :4,Last;
352$            TRACE,          -1,"-Entity created to represent temporary batch of size
4\n";
356$            DELAY:          0.000:NEXT(6$);

6$              DELAY:          0.00;
373$            ASSIGN:         M=ENT_OVEN;
369$            TRACE,          -1,"-Waiting for conveyor DIP_BAKE_CONV\n";
391$            ACCESS:         DIP_BAKE_CONV,1;
366$            TRACE,          -1,"-Delay for loading time .2\n";
389$            DELAY:          .2;
399$            ASSIGN:         Picture=OVEN;
362$            TRACE,          -1,"-Transferred to station ENT_COOLING\n";
381$            CONVEY:         ,ENT_COOLING;
10$             STATION,        REJECTED_WOUND;
408$            TRACE,          -1,"-Arrived to station REJECTED_WOUND\n";
438$            DELAY:          0.;
416$            COUNT:          REJECTED_WOUND_C,1;
401$            TRACE,          -1,"-Disposing entity\n";
409$            DISPOSE;
11$             STATION,        ELEC_TEST;
479$            TRACE,          -1,"-Arrived to Inspect station ELEC_TEST\n";
538$            DELAY:          0.;
594$            QUEUE,          ELEC_TEST_R_Q;
593$            SEIZE,          1:ELEC_TEST_R,1;
                TRACE,          -1,"-Delay for processing time TRIA(1,1.5,2) with failure
probability .05\n";
592$            DELAY:          TRIA(1,1.5,2);
582$            BRANCH,         1:With,.05,471$,Yes:
                                Else,472$,Yes;
471$            TRACE,          -1,"-Entity failed inspection\n";
486$            DELAY:          0.0;
527$            RELEASE:        ELEC_TEST_R,1;
502$            DELAY:          0.;
467$            TRACE,          -1,"-Transferred to station REJECTED_WOUND\n";
584$            ROUTE:          .1,REJECTED_WOUND;
472$            TRACE,          -1,"-Entity passed inspection\n";
487$            DELAY:          0.0;
591$            RELEASE:        ELEC_TEST_R,1;
503$            DELAY:          0.;
461$            TRACE,          -1,"-Transferring to station ENT_OVEN\n";
586$            ROUTE:          .3,ENT_OVEN;
```

```
13$        STATION,     TERMING;
624$       TRACE,       -1,"-Arrived to station TERMING\n";
744$       STORE:       TERMING_S1;
745$       DELAY:       0.;
729$       UNSTORE:     ;
617$       TRACE,       -1,"-Waiting for resource TERMING_R\n";
726$       QUEUE,       TERMING_R_Q;
723$       SEIZE,       1:TERMING_R,1;
651$       STORE:       TERMING_S3;
           TRACE,       -1,"-Delay for processing time TRIA(4,5,8)\n";
710$       DELAY:       TRIA(4,5,8);
687$       UNSTORE:     ;
680$       STORE:       TERMING_S4;
614$       TRACE,       -1,"-Releasing resource\n";
703$       RELEASE:     TERMING_R,1;
667$       DELAY:       0.;
664$       UNSTORE:     ;
608$       TRACE,       -1,"-Transferred to station ELEC_TEST\n";
665$       ROUTE:       .3,ELEC_TEST;
15$        STATION,     PREESERS;
789$       TRACE,       -1,"-Arrived to station PREESERS\n";
909$       STORE:       PREESERS_S1;
910$       DELAY:       0.;
894$       UNSTORE:     ;
782$       TRACE,       -1,"-Waiting for resource PRESSERS_M\n";
891$       QUEUE,       PRESSERS_M_Q;
887$       SEIZE,       1:SELECT(PRESSERS_M,RAN,),1;
816$       STORE:       PREESERS_S3;
           TRACE,       -1,"-Delay for processing time TRIA(.8,1.5,2.3)\n";
875$       DELAY:       TRIA(.8,1.5,2.3);
852$       UNSTORE:     ;
845$       STORE:       PREESERS_S4;
779$       TRACE,       -1,"-Releasing resource\n";
867$       RELEASE:     SELECT(PRESSERS_M,,),1;
832$       DELAY:       0.;
829$       UNSTORE:     ;
773$       TRACE,       -1,"-Transferred to station TERMING\n";
830$       ROUTE:       1,TERMING;
19$        STATION,     CONNECTING;
982$       TRACE,       -1,"-Arrived to station CONNECTING\n";
1103$      DELAY:       0.;
976$       TRACE,       -1,"-Exiting conveyor\n";
1091$      EXIT:        ;
975$       TRACE,       -1,"-Waiting for resource CONN_OPER\n";
1084$      QUEUE,       CONN_OPER_Q;
1080$      SEIZE,       1:SELECT(CONN_OPER,POR,),1;
1009$      STORE:       CONNECTING_S3;
           TRACE,       -1,"-Delay for processing time TRIA(12,18,22)\n";
1068$      DELAY:       TRIA(12,18,22);
1045$      UNSTORE:     ;
1038$      STORE:       CONNECTING_S4;
972$       TRACE,       -1,"-Releasing resource\n";
1060$      RELEASE:     SELECT(CONN_OPER,,),1;
1025$      DELAY:       0.;
1022$      UNSTORE:     ;
966$       TRACE,       -1,"-Transferred to station PREESERS\n";
1023$      ROUTE:       1,PREESERS;
26$        STATION,     FINISHING OFF;
1189$      TRACE,       -1,"-Arrived to station FINISHING OFF\n";
1309$      STORE:       FINISHING OFF_S1;
1310$      DELAY:       0.;
1294$      UNSTORE:     ;
1182$      TRACE,       -1,"-Waiting for resource FINISHING OFF OP\n";
1291$      QUEUE,       FINISHING OFF OP_Q;
1287$      SEIZE,       1:SELECT(FINISHING OFF OP,POR,),1;
1216$      STORE:       FINISHING OFF_S3;
           TRACE,       -1,"-Delay for processing time TRIA(18,20,25)\n";
1275$      DELAY:       TRIA(18,20,25);
1252$      UNSTORE:     ;
1245$      STORE:       FINISHING OFF_S4;
1179$      TRACE,       -1,"-Releasing resource\n";
1267$      RELEASE:     SELECT(FINISHING OFF OP,,),1;
1232$      DELAY:       0.;
1229$      UNSTORE:     ;
1172$      TRACE,       -1,"-Transferred to next module\n";
1225$      DELAY:       0.00:NEXT(18$);

18$        STATION,     EXIT FINI;
1317$      TRACE,       -1,"-Arrived to station EXIT FINI\n";
1338$      DELAY:       0.;
```

```
1311$          TRACE,         -1,"-Exiting conveyor\n";
1327$          EXIT:          ;
1318$          DELAY:         0.000:NEXT(17$);

17$            DELAY:         0.00;
1353$          ASSIGN:        M=EXIT FINI;
1349$          TRACE,         -1,"-Waiting for conveyor FINI CONV\n";
1371$          ACCESS:        FINI CONV,1;
1369$          DELAY:         0.;
1342$          TRACE,         -1,"-Transferred to station CONNECTING\n";
1361$          CONVEY:        ,CONNECTING;
28$            STATION,       WINDER4;
1410$          TRACE,         -1,"-Arrived to station WINDER4\n";
1530$          STORE:         WINDER4_S1;
1531$          DELAY:         0.;
1515$          UNSTORE:       ;
1403$          TRACE,         -1,"-Waiting for resource WINDER4_R\n";
1512$          QUEUE,         WINDER4_R_Q;
1509$          SEIZE,         1:WINDER4_R,1;
1437$          STORE:         WINDER4_S3;
               TRACE,         -1,"-Delay for processing time TRIA(10,15,22)\n";
1496$          DELAY:         TRIA(10,15,22);
1473$          UNSTORE:       ;
1466$          STORE:         WINDER4_S4;
1400$          TRACE,         -1,"-Releasing resource\n";
1489$          RELEASE:       WINDER4_R,1;
1453$          DELAY:         0.;
1450$          UNSTORE:       ;
1394$          TRACE,         -1,"-Transferred to station FINISHING OFF\n";
1445$          CONVEY:        ,FINISHING OFF;
30$            STATION,       WINDER3;
1561$          TRACE,         -1,"-Arrived to station WINDER3\n";
1681$          STORE:         WINDER3_S1;
1682$          DELAY:         0.;
1666$          UNSTORE:       ;
1554$          TRACE,         -1,"-Waiting for resource WINDER3_R\n";
1663$          QUEUE,         WINDER3_R_Q;
1660$          SEIZE,         1:WINDER3_R,1;
1588$          STORE:         WINDER3_S3;
               TRACE,         -1,"-Delay for processing time TRIA(10,15,20)\n";
1647$          DELAY:         TRIA(10,15,20);
1624$          UNSTORE:       ;
1617$          STORE:         WINDER3_S4;
1551$          TRACE,         -1,"-Releasing resource\n";
1640$          RELEASE:       WINDER3_R,1;
1604$          DELAY:         0.;
1601$          UNSTORE:       ;
1545$          TRACE,         -1,"-Transferred to station FINISHING OFF\n";
1596$          CONVEY:        ,FINISHING OFF;
32$            STATION,       WINDER2;
1712$          TRACE,         -1,"-Arrived to station WINDER2\n";
1832$          STORE:         WINDER2_S1;
1833$          DELAY:         0.;
1817$          UNSTORE:       ;
1705$          TRACE,         -1,"-Waiting for resource WINDER2_R\n";
1814$          QUEUE,         WINDER2_R_Q;
1811$          SEIZE,         1:WINDER2_R,1;
1739$          STORE:         WINDER2_S3;
               TRACE,         -1,"-Delay for processing time TRIA(10,15,22)\n";
1798$          DELAY:         TRIA(10,15,22);
1775$          UNSTORE:       ;
1768$          STORE:         WINDER2_S4;
1702$          TRACE,         -1,"-Releasing resource\n";
1791$          RELEASE:       WINDER2_R,1;
1755$          DELAY:         0.;
1752$          UNSTORE:       ;
1696$          TRACE,         -1,"-Transferred to station FINISHING OFF\n";
1747$          CONVEY:        ,FINISHING OFF;
34$            STATION,       WINDER1;
1863$          TRACE,         -1,"-Arrived to station WINDER1\n";
1983$          STORE:         WINDER1_S1;
1984$          DELAY:         0.;
1968$          UNSTORE:       ;
1856$          TRACE,         -1,"-Waiting for resource WINDER1_R\n";
1965$          QUEUE,         WINDER1_R_Q;
1962$          SEIZE,         1:WINDER1_R,1;
1890$          STORE:         WINDER1_S3;
               TRACE,         -1,"-Delay for processing time TRIA(10,15,20)\n";
1949$          DELAY:         TRIA(10,15,20);
1926$          UNSTORE:       ;
```

```
1919$           STORE:          WINDER1_S4;
1853$           TRACE,          -1,"-Releasing resource\n";
1942$           RELEASE:        WINDER1_R,1;
1906$           DELAY:          0.;
1903$           UNSTORE:        ;
1847$           TRACE,          -1,"-Transferred to station FINISHING OFF\n";
1898$           CONVEY:         ,FINISHING OFF;
37$             STATION,        INSULATOR;
2014$           TRACE,          -1,"-Arrived to station INSULATOR\n";
2135$           DELAY:          0.;
2008$           TRACE,          -1,"-Exiting conveyor\n";
2123$           EXIT:           ;
2007$           TRACE,          -1,"-Waiting for resource INSULATOR_R\n";
2116$           QUEUE,          INSULATOR_R_Q;
2113$           SEIZE,          1:INSULATOR_R,1;
2041$           STORE:          INSULATOR_S3;
                TRACE,          -1,"-Delay for processing time TRIA(.3,1,1.1)\n";
2100$           DELAY:          TRIA(.3,1,1.1);
2077$           UNSTORE:        ;
2070$           STORE:          INSULATOR_S4;
2004$           TRACE,          -1,"-Releasing resource\n";
2093$           RELEASE:        INSULATOR_R,1;
2057$           DELAY:          0.;
2054$           UNSTORE:        ;
1997$           TRACE,          -1,"-Transferred to next module\n";
2050$           DELAY:          0.00:NEXT(25$);

25$             TRACE,          -1,"-Choosing from 4 options\n";
2136$           BRANCH,         1:With,.25,24$,Yes:
                                With,.25,23$,Yes:
                                With,.25,22$,Yes:
                                With,.25,21$,Yes;
24$             DELAY:          0.00;
2151$           ASSIGN:         M=INSULATOR;
2147$           TRACE,          -1,"-Waiting for conveyor WINDER CONV1\n";
2169$           ACCESS:         WINDER CONV1,1;
2167$           DELAY:          0.;
2140$           TRACE,          -1,"-Transferred to station WINDER1\n";
2159$           CONVEY:         ,WINDER1;
23$             DELAY:          0.00;
2193$           ASSIGN:         M=INSULATOR;
2189$           TRACE,          -1,"-Waiting for conveyor WINDER CONV2\n";
2211$           ACCESS:         WINDER CONV2,1;
2209$           DELAY:          0.;
2182$           TRACE,          -1,"-Transferred to station WINDER2\n";
2201$           CONVEY:         ,WINDER2;
22$             DELAY:          0.00;
2235$           ASSIGN:         M=INSULATOR;
2231$           TRACE,          -1,"-Waiting for conveyor WINDER CONV3\n";
2253$           ACCESS:         WINDER CONV3,1;
2251$           DELAY:          0.;
2224$           TRACE,          -1,"-Transferred to station WINDER3\n";
2243$           CONVEY:         ,WINDER3;
21$             DELAY:          0.00;
2277$           ASSIGN:         M=INSULATOR;
2273$           TRACE,          -1,"-Waiting for conveyor WINDER CONV4\n";
2295$           ACCESS:         WINDER CONV4,1;
2293$           DELAY:          0.;
2266$           TRACE,          -1,"-Transferred to station WINDER4\n";
2285$           CONVEY:         ,WINDER4;
39$             STATION,        REJECT LAM;
2312$           TRACE,          -1,"-Arrived to station REJECT LAM\n";
2342$           DELAY:          0.;
2306$           TRACE,          -1,"-Exiting conveyor\n";
2332$           EXIT:           ;
2320$           COUNT:          REJECT LAM_C,1;
2305$           TRACE,          -1,"-Disposing entity\n";
2313$           DISPOSE;
40$             STATION,        LAM INSPECT;
2383$           TRACE,          -1,"-Arrived to Inspect station LAM INSPECT\n";
2442$           DELAY:          0.;
2498$           QUEUE,          LAM INSPECT_R_Q;
2497$           SEIZE,          1:LAM INSPECT_R,1;
                TRACE,          -1,"-Delay for processing time TRIA(.17,.25,.30) with
failure probability .1\n";
2496$           DELAY:          TRIA(.17,.25,.30);
2486$           BRANCH,         1:With,.1,2375$,Yes:
                                Else,2376$,Yes;

2375$           TRACE,          -1,"-Entity failed inspection\n";
2390$           DELAY:          0.0;
```

```
2431$         RELEASE:      LAM INSPECT_R,1;
2406$         DELAY:        0.;
2371$         TRACE,        -1,"-Transferred to station REJECT LAM\n";
2488$         ROUTE:        .06,REJECT LAM;
2376$         TRACE,        -1,"-Entity passed inspection\n";
2391$         DELAY:        0.0;
2495$         RELEASE:      LAM INSPECT_R,1;
2407$         DELAY:        0.;
2364$         TRACE,        -1,"-Transferrinf to next module\n";
2489$         DELAY:        0.000:NEXT(36$);

36$           DELAY:        0.00;
2513$         ASSIGN:       M=LAM INSPECT;
2529$         DELAY:        0.;
2502$         TRACE,        -1,"-Transferred to station INSULATOR\n";
2521$         CONVEY:       ,INSULATOR;
42$           STATION,      CLEATER;
2636$         TRACE,        -1,"-Arrived to station CLEATER\n";
2756$         STORE:        CLEATER_S1;
2757$         DELAY:        0.;
2741$         UNSTORE:      ;
2629$         TRACE,        -1,"-Waiting for resource CLEATER_R\n";
2738$         QUEUE,        CLEATER_R_Q;
2735$         SEIZE,        1:CLEATER_R,1;
2663$         STORE:        CLEATER_S3;
              TRACE,        -1,"-Delay for processing time TRIA(.8,1,1.25)\n";
2722$         DELAY:        TRIA(.8,1,1.25);
2699$         UNSTORE:      ;
2692$         STORE:        CLEATER_S4;
2626$         TRACE,        -1,"-Releasing resource\n";
2715$         RELEASE:      CLEATER_R,1;
2679$         DELAY:        0.;
2676$         UNSTORE:      ;
2620$         TRACE,        -1,"-Transferred to station LAM INSPECT\n";
2671$         CONVEY:       ,LAM INSPECT;
44$           STATION,      SCALING;
2787$         TRACE,        -1,"-Arrived to station SCALING\n";
2908$         DELAY:        0.;
2782$         TRACE,        -1,"-Freeing transporter\n";
2902$         FREE:         ;
2779$         TRACE,        -1,"-Waiting for transporter LAM LABOUR\n";
2889$         QUEUE,        LAM LABOUR_Q,10,2819$;
2883$         REQUEST,      1:LAM LABOUR(CYC);
2814$         STORE:        SCALING_S3;
              TRACE,        -1,"-Delay for processing time TRIA(.21,.5,.62)\n";
2873$         DELAY:        TRIA(.21,.5,.62);
2850$         UNSTORE:      ;
2843$         STORE:        SCALING_S4;
2776$         TRACE,        -1,"-Freeing transporter\n";
2863$         FREE:         ;
2767$         TRACE,        -1,"-Waiting for conveyor CLEAT CONV\n";
2832$         ACCESS:       CLEAT CONV,1;
2830$         DELAY:        0.;
2827$         UNSTORE:      ;
2771$         TRACE,        -1,"-Transferred to station CLEATER\n";
2822$         CONVEY:       ,CLEATER;
2819$         DISPOSE;
2933$         CREATE,       1:1:MARK(TIME IN);
2973$         STATION,      ARR_LAM;
2924$         TRACE,        -1,"-Arrived to system at station ARR_LAM\n";
2970$         ASSIGN:       Picture=Default;
2915$         TRACE,        -1,"-Waiting for transporter LAM LABOUR\n";
2952$         REQUEST,      1:LAM LABOUR(CYC);
2949$         DELAY:        0.;
2918$         TRACE,        -1,"-Transferred to station SCALING\n";
2945$         TRANSPORT,    :,SCALING;
```

E.14. Experiment Listing for Windings Assembly (Cell-160/180)

```
PROJECT,        ;

ATTRIBUTES:     TIME IN;

STORAGES:       SCALING_S4:
                SCALING_S3:
                CLEATER_S4:
```

```
                CLEATER_S3:
                CLEATER_S1:
                INSULATOR_S4:
                INSULATOR_S3:
                WINDER1_S4:
                WINDER1_S3:
                WINDER1_S1:
                WINDER2_S4:
                WINDER2_S3:
                WINDER2_S1:
                WINDER3_S4:
                WINDER3_S3:
                WINDER3_S1:
                WINDER4_S4:
                WINDER4_S3:
                WINDER4_S1:
                FINISHING OFF_S4:
                FINISHING OFF_S3:
                FINISHING OFF_S1:
                CONNECTING_S4:
                CONNECTING_S3:
                PREESERS_S4:
                PREESERS_S3:
                PREESERS_S1:
                TERMING_S4:
                TERMING_S3:
                TERMING_S1:
                ENT_OVEN_S1:
                WYVERTIC_S4:
                WYVERTIC_S3:
                WYVERTIC_S1;

QUEUES:         LAM LABOUR_Q,FIFO:
                CLEATER_R_Q,FIFO:
                LAM INSPECT_R_Q,FIFO:
                INSULATOR_R_Q,FIFO:
                WINDER1_R_Q,FIFO:
                WINDER2_R_Q,FIFO:
                WINDER3_R_Q,FIFO:
                WINDER4_R_Q,FIFO:
                FINISHING OFF OP_Q,FIFO:
                F_OFF_OPERATOR1_Q,FIFO:
                F_OFF_OPERATOR2_Q,FIFO:
                F_OFF_OPERATOR3_Q,FIFO:
                F_OFF_OPERATOR5_Q,FIFO:
                F_OFF_OPERATOR6_Q,FIFO:
                F_OFF_OPERATOR7_Q,FIFO:
                F_OFF_OPERATOR8_Q,FIFO:
                CONN_OPER_Q,FIFO:
                CON_OPERATOR1_Q,FIFO:
                CON_OPERATOR2_Q,FIFO:
                CON_OPERATOR3_Q,FIFO:
                CON_OPERATOR4_Q,FIFO:
                CON_OPERATOR5_Q,FIFO:
                CON_OPERATOR6_Q,FIFO:
                PRESSERS_M_Q,FIFO:
                PRESSER_1_Q,FIFO:
                PRESSER_2_Q,FIFO:
                TERMING_R_Q,FIFO:
                ELEC_TEST_R_Q,FIFO:
                WYVERTIC MACHINES_Q,FIFO:
                WYVERTIC_1_Q,FIFO:
                WYVERTIC_2_Q,FIFO;

PICTURES:       Default:
                OVEN;

RESOURCES:      CLEATER_R,Capacity(1,):
                LAM INSPECT_R,Capacity(1,):
                INSULATOR_R,Capacity(1,):
                WINDER1_R,Capacity(1,):
                WINDER2_R,Capacity(1,):
                WINDER3_R,Capacity(1,):
                WINDER4_R,Capacity(1,):
                F_OFF_OPERATOR1,Capacity(1,):
                F_OFF_OPERATOR2,Capacity(1,):
                F_OFF_OPERATOR3,Capacity(1,):
                F_OFF_OPERATOR4:
                F_OFF_OPERATOR5,Capacity(1,):
                F_OFF_OPERATOR6,Capacity(1,):
```

```
              F_OFF_OPERATOR7,Capacity(1,):
              F_OFF_OPERATOR8,Capacity(1,):
              CON_OPERATOR1,Capacity(1,):
              CON_OPERATOR2,Capacity(1,):
              CON_OPERATOR3,Capacity(1,):
              CON_OPERATOR4,Capacity(1,):
              CON_OPERATOR5,Capacity(1,):
              CON_OPERATOR6,Capacity(1,):
              PRESSER_1,Capacity(1,):
              PRESSER_2,Capacity(1,):
              TERMING_R,Capacity(1,):
              ELEC_TEST_R,Capacity(1,):
              WYVERTIC_1,Capacity(1,):
              WYVERTIC_2,Capacity(1,);

STATIONS:     ARR_LAM:
              SCALING:
              CLEATER:
              LAM INSPECT:
              INSULATOR:
              WINDER1:
              FINISHING OFF:
              WINDER2:
              WINDER3:
              WINDER4:
              CONNECTING:
              EXIT FINI:
              PREESERS:
              TERMING:
              ELEC_TEST:
              ENT_OVEN:
              REJECTED_WOUND:
              ENT_COOLING:
              WYVERTIC:
              F_WOUND PACK:
              REJECT LAM;

DISTANCES:    LAM LABOUR_Dst,ARR_LAM-SCALING-2;

TRANSPORTERS: LAM LABOUR,1,DISTANCE(LAM LABOUR_Dst),1.;

SEGMENTS:     CLEAT CONV_Seg,SCALING,CLEATER-5,
              LAM INSPECT-2,
              INSULATOR-3:
              WINDER CONV1_Seg,INSULATOR,WINDER1-5,
              FINISHING OFF-8:
              WINDER CONV2_Seg,INSULATOR,WINDER2-5,
              FINISHING OFF-15:
              WINDER CONV3_Seg,INSULATOR,WINDER3-8,
              FINISHING OFF-5:
              WINDER CONV4_Seg,INSULATOR,WINDER4-8,
              FINISHING OFF-10:
              FINI CONV_Seg,EXIT FINI,CONNECTING-10:
              DIP_BAKE_CONV_Seg,ENT_OVEN,ENT_COOLING-90;

CONVEYORS:    CLEAT CONV,CLEAT CONV_Seg,1.,1,Active,1,Accumulating:
              WINDER CONV1,WINDER CONV1_Seg,1.,1,Active,1,Accumulating:
              WINDER CONV2,WINDER CONV2_Seg,1.,1,Active,1,Accumulating:
              WINDER CONV3,WINDER CONV3_Seg,1.,1,Active,1,Accumulating:
              WINDER CONV4,WINDER CONV4_Seg,1.,1,Active,1,Accumulating:
              FINI CONV,FINI CONV_Seg,1.,1,Active,1,Accumulating:
              DIP_BAKE_CONV,DIP_BAKE_CONV_Seg,1,1,Active,1,Accumulating;

COUNTERS:     REJECTED_WOUND_C:
              F_WOUND PACK_C,,Replicate,"C:\BH\SIM\WIND\CO160.001":
              REJECT LAM_C;

TALLIES:      F_WOUND PACK_Ta,"C:\BH\SIM\WIND\TA160.001";

DSTATS:       NQ(LAM LABOUR_Q),# in LAM LABOUR_Q:
              MT(LAM LABOUR),LAM LABOUR Active:
              NT(LAM LABOUR),LAM LABOUR Busy:
              MR(CLEATER_R),CLEATER_R Available:
              NR(CLEATER_R),CLEATER_R Busy:
              NQ(CLEATER_R_Q),# in CLEATER_R_Q:
              MR(LAM INSPECT_R),LAM INSPECT_R Available:
              NR(LAM INSPECT_R),LAM INSPECT_R Busy:
              NQ(LAM INSPECT_R_Q),# in LAM INSPECT_R_Q:
              MR(INSULATOR_R),INSULATOR_R Available:
              NR(INSULATOR_R),INSULATOR_R Busy:
```

```
                NQ(INSULATOR_R_Q),# in INSULATOR_R_Q:
                MR(WINDER1_R),WINDER1_R Available:
                NR(WINDER1_R),WINDER1_R Busy:
                NQ(WINDER1_R_Q),# in WINDER1_R_Q:
                MR(WINDER2_R),WINDER2_R Available:
                NR(WINDER2_R),WINDER2_R Busy:
                NQ(WINDER2_R_Q),# in WINDER2_R_Q:
                MR(WINDER3_R),WINDER3_R Available:
                NR(WINDER3_R),WINDER3_R Busy:
                NQ(WINDER3_R_Q),# in WINDER3_R_Q:
                MR(WINDER4_R),WINDER4_R Available:
                NR(WINDER4_R),WINDER4_R Busy:
                NQ(WINDER4_R_Q),# in WINDER4_R_Q:
                NQ(FINISHING OFF OP_Q),# in FINISHING OFF OP_Q:
                MR(F_OFF_OPERATOR1),F_OFF_OPERATOR1 Available:
                NR(F_OFF_OPERATOR1),F_OFF_OPERATOR1 Busy:
                NQ(F_OFF_OPERATOR1_Q),# in F_OFF_OPERATOR1_Q:
                MR(F_OFF_OPERATOR2),F_OFF_OPERATOR2 Available:
                NR(F_OFF_OPERATOR2),F_OFF_OPERATOR2 Busy:
                NQ(F_OFF_OPERATOR2_Q),# in F_OFF_OPERATOR2_Q:
                MR(F_OFF_OPERATOR3),F_OFF_OPERATOR3 Available:
                NR(F_OFF_OPERATOR3),F_OFF_OPERATOR3 Busy:
                NQ(F_OFF_OPERATOR3_Q),# in F_OFF_OPERATOR3_Q:
                MR(F_OFF_OPERATOR5),F_OFF_OPERATOR5 Available:
                NR(F_OFF_OPERATOR5),F_OFF_OPERATOR5 Busy:
                NQ(F_OFF_OPERATOR5_Q),# in F_OFF_OPERATOR5_Q:
                MR(F_OFF_OPERATOR6),F_OFF_OPERATOR6 Available:
                NR(F_OFF_OPERATOR6),F_OFF_OPERATOR6 Busy:
                NQ(F_OFF_OPERATOR6_Q),# in F_OFF_OPERATOR6_Q:
                MR(F_OFF_OPERATOR7),F_OFF_OPERATOR7 Available:
                NR(F_OFF_OPERATOR7),F_OFF_OPERATOR7 Busy:
                NQ(F_OFF_OPERATOR7_Q),# in F_OFF_OPERATOR7_Q:
                MR(F_OFF_OPERATOR8),F_OFF_OPERATOR8 Available:
                NR(F_OFF_OPERATOR8),F_OFF_OPERATOR8 Busy:
                NQ(F_OFF_OPERATOR8_Q),# in F_OFF_OPERATOR8_Q:
                NQ(CONN_OPER_Q),# in CONN_OPER_Q:
                MR(CON_OPERATOR1),CON_OPERATOR1 Available:
                NR(CON_OPERATOR1),CON_OPERATOR1 Busy:
                NQ(CON_OPERATOR1_Q),# in CON_OPERATOR1_Q:
                MR(CON_OPERATOR2),CON_OPERATOR2 Available:
                NR(CON_OPERATOR2),CON_OPERATOR2 Busy:
                NQ(CON_OPERATOR2_Q),# in CON_OPERATOR2_Q:
                MR(CON_OPERATOR3),CON_OPERATOR3 Available:
                NR(CON_OPERATOR3),CON_OPERATOR3 Busy:
                NQ(CON_OPERATOR3_Q),# in CON_OPERATOR3_Q:
                MR(CON_OPERATOR4),CON_OPERATOR4 Available:
                NR(CON_OPERATOR4),CON_OPERATOR4 Busy:
                NQ(CON_OPERATOR4_Q),# in CON_OPERATOR4_Q:
                MR(CON_OPERATOR5),CON_OPERATOR5 Available:
                NR(CON_OPERATOR5),CON_OPERATOR5 Busy:
                NQ(CON_OPERATOR5_Q),# in CON_OPERATOR5_Q:
                MR(CON_OPERATOR6),CON_OPERATOR6 Available:
                NR(CON_OPERATOR6),CON_OPERATOR6 Busy:
                NQ(CON_OPERATOR6_Q),# in CON_OPERATOR6_Q:
                NQ(PRESSERS_M_Q),# in PRESSERS_M_Q:
                MR(PRESSER_1),PRESSER_1 Available:
                NR(PRESSER_1),PRESSER_1 Busy:
                NQ(PRESSER_1_Q),# in PRESSER_1_Q:
                MR(PRESSER_2),PRESSER_2 Available:
                NR(PRESSER_2),PRESSER_2 Busy:
                NQ(PRESSER_2_Q),# in PRESSER_2_Q:
                MR(TERMING_R),TERMING_R Available:
                NR(TERMING_R),TERMING_R Busy:
                NQ(TERMING_R_Q),# in TERMING_R_Q:
                MR(ELEC_TEST_R),ELEC_TEST_R Available:
                NR(ELEC_TEST_R),ELEC_TEST_R Busy:
                NQ(ELEC_TEST_R_Q),# in ELEC_TEST_R_Q:
                NQ(WYVERTIC MACHINES_Q),# in WYVERTIC MACHINES_Q:
                MR(WYVERTIC_1),WYVERTIC_1 Available:
                NR(WYVERTIC_1),WYVERTIC_1 Busy:
                NQ(WYVERTIC_1_Q),# in WYVERTIC_1_Q:
                MR(WYVERTIC_2),WYVERTIC_2 Available:
                NR(WYVERTIC_2),WYVERTIC_2 Busy:
                NQ(WYVERTIC_2_Q),# in WYVERTIC_2_Q;

REPLICATE,      3,0.0,3420,Yes,Yes;

SETS:           FINISHING OFF
OP,F_OFF_OPERATOR1,F_OFF_OPERATOR2,F_OFF_OPERATOR3,F_OFF_OPERATOR4,F_OFF_OPERATOR5,F_OFF
_OPERATOR6,
```

```
               F_OFF_OPERATOR7,F_OFF_OPERATOR8:

CONN_OPER,CON_OPERATOR1,CON_OPERATOR2,CON_OPERATOR3,CON_OPERATOR4,CON_OPERATOR5,CON_OPER
ATOR6:
               PRESSERS_M,PRESSER_1,PRESSER_2:
               ENT,Default,OVEN:
               WYVERTIC MACHINES,WYVERTIC_1,WYVERTIC_2;
```

E.15. Model Listing for Windings Assembly (Cell-100)

```
0$             STATION,       FINI_WOUND_P;
37$            TRACE,         -1,"-Arrived to station FINI_WOUND_P\n";
67$            DELAY:         0.;
45$            COUNT:         FINI_WOUND_P_C,1;
40$            TALLY:         FINI_WOUND_P_Ta,Interval(TIME IN SYSTEM),1;
30$            TRACE,         -1,"-Disposing entity\n";
38$            DISPOSE;
3$             STATION,       E_FINI_WOUND_P;
74$            TRACE,         -1,"-Arrived to station E_FINI_WOUND_P\n";
94$            STORE:         E_FINI_WOUND_P_S1;
95$            DELAY:         0.;
80$            UNSTORE:       ;
75$            DELAY:         0.000:NEXT(2$);

2$             TRACE,         -1,"-Splitting group representative\n";
96$            SPLIT:         A(*),M,NS,IS:NEXT(1$);

1$             DELAY:         0.00;
111$           ASSIGN:        M=E_FINI_WOUND_P;
127$           DELAY:         0.;
100$           TRACE,         -1,"-Transferred to station FINI_WOUND_P\n";
125$           ROUTE:         1.,FINI_WOUND_P;
4$             STATION,       DIP_WYVERT;
168$           TRACE,         -1,"-Arrived to station DIP_WYVERT\n";
288$           STORE:         DIP_WYVERT_S1;
289$           DELAY:         0.;
273$           UNSTORE:       ;
161$           TRACE,         -1,"-Waiting for resource DIP_WYVERT_R\n";
270$           QUEUE,         DIP_WYVERT_R_Q;
267$           SEIZE,         1:DIP_WYVERT_R,1;
195$           STORE:         DIP_WYVERT_S3;
               TRACE,         -1,"-Delay for processing time TRIA(95,100,105)\n";
254$           DELAY:         TRIA(95,100,105);
231$           UNSTORE:       ;
224$           STORE:         DIP_WYVERT_S4;
158$           TRACE,         -1,"-Releasing resource\n";
247$           RELEASE:       DIP_WYVERT_R,1;
211$           DELAY:         0.;
208$           UNSTORE:       ;
152$           TRACE,         -1,"-Transferred to station E_FINI_WOUND_P\n";
209$           ROUTE:         5,E_FINI_WOUND_P;
9$             STATION,       BATCHING_WIND;
296$           TRACE,         -1,"-Arrived to station BATCHING_WIND\n";
316$           STORE:         BATCHING_WIND_S1;
317$           DELAY:         0.;
302$           UNSTORE:       ;
297$           DELAY:         0.000:NEXT(8$);

8$             DELAY:         0.00;
320$           TRACE,         -1,"-Waiting for batch size of 20\n";
328$           GROUP,         :20,Last;
322$           TRACE,         -1,"-Entity created to represent temporary batch of size
20\n";
326$           DELAY:         0.000:NEXT(6$);

6$             DELAY:         0.00;
343$           ASSIGN:        M=BATCHING_WIND;
359$           DELAY:         0.;
332$           TRACE,         -1,"-Transferred to station DIP_WYVERT\n";
357$           ROUTE:         4,DIP_WYVERT;
10$            STATION,       REJECT_WIND;
378$           TRACE,         -1,"-Arrived to station REJECT_WIND\n";
408$           DELAY:         0.;
386$           COUNT:         REJECT_WIND_C,1;
371$           TRACE,         -1,"-Disposing entity\n";
379$           DISPOSE;
```

```
11$            STATION,      ELCT_TEST;
449$           TRACE,        -1,"-Arrived to Inspect station ELCT_TEST\n";
508$           DELAY:        0.;
564$           QUEUE,        ELCT_TEST_R_Q;
563$           SEIZE,        1:ELCT_TEST_R,1;
               TRACE,        -1,"-Delay for processing time TRIA(2,2.5,3) with failure
probability .05\n";
562$           DELAY:        TRIA(2,2.5,3);
552$           BRANCH,       1:With,.05,441$,Yes:
                             Else,442$,Yes;
441$           TRACE,        -1,"-Entity failed inspection\n";
456$           DELAY:        0.0;
497$           RELEASE:      ELCT_TEST_R,1;
472$           DELAY:        0.;
437$           TRACE,        -1,"-Transferred to station REJECT_WIND\n";
554$           ROUTE:        2,REJECT_WIND;
442$           TRACE,        -1,"-Entity passed inspection\n";
457$           DELAY:        0.0;
561$           RELEASE:      ELCT_TEST_R,1;
473$           DELAY:        0.;
431$           TRACE,        -1,"-Transferring to station BATCHING_WIND\n";
556$           ROUTE:        3,BATCHING_WIND;
13$            STATION,      LACING;
590$           TRACE,        -1,"-Arrived to station LACING\n";
627$           DELAY:        0.;
583$           TRACE,        -1,"-Waiting for resource LACING_R\n";
668$           QUEUE,        LACING_R_Q;
667$           SEIZE,        1:LACING_R,1;
               TRACE,        -1,"-Delay for processing time TRIA(.5,.55,.7)\n";
666$           DELAY:        TRIA(.5,.55,.7);
582$           TRACE,        -1,"-Releasing resource\n";
665$           RELEASE:      LACING_R,1;
599$           DELAY:        0.;
577$           TRACE,        -1,"-Transferred to station ELCT_TEST\n";
660$           ROUTE:        1,ELCT_TEST;
15$            STATION,      CONNECTING;
733$           TRACE,        -1,"-Arrived to station CONNECTING\n";
853$           STORE:        CONNECTING_S1;
854$           DELAY:        0.;
838$           UNSTORE:      ;
726$           TRACE,        -1,"-Waiting for resource CONNECTORS\n";
835$           QUEUE,        CONNECTORS_Q;
831$           SEIZE,        1:SELECT(CONNECTORS,CYC,),1;
760$           STORE:        CONNECTING_S3;
               TRACE,        -1,"-Delay for processing time TRIA(10,12,15)\n";
819$           DELAY:        TRIA(10,12,15);
796$           UNSTORE:      ;
789$           STORE:        CONNECTING_S4;
723$           TRACE,        -1,"-Releasing resource\n";
811$           RELEASE:      SELECT(CONNECTORS,,),1;
776$           DELAY:        0.;
773$           UNSTORE:      ;
717$           TRACE,        -1,"-Transferred to station PRESSING\n";
774$           ROUTE:        5,PRESSING;
17$            STATION,      PRESSING;
884$           TRACE,        -1,"-Arrived to station PRESSING\n";
1004$          STORE:        PRESSING_S1;
1005$          DELAY:        0.;
989$           UNSTORE:      ;
877$           TRACE,        -1,"-Waiting for resource PRESSING_R\n";
986$           QUEUE,        PRESSING_R_Q;
983$           SEIZE,        1:PRESSING_R,1;
911$           STORE:        PRESSING_S3;
               TRACE,        -1,"-Delay for processing time TRIA(.7,.8,1)\n";
970$           DELAY:        TRIA(.7,.8,1);
947$           UNSTORE:      ;
940$           STORE:        PRESSING_S4;
874$           TRACE,        -1,"-Releasing resource\n";
963$           RELEASE:      PRESSING_R,1;
927$           DELAY:        0.;
924$           UNSTORE:      ;
868$           TRACE,        -1,"-Transferred to station LACING\n";
925$           ROUTE:        .3,LACING;
19$            STATION,      FINISHING OFF;
1070$          TRACE,        -1,"-Arrived to station FINISHING OFF\n";
1190$          STORE:        FINISHING OFF_S1;
1191$          DELAY:        0.;
1175$          UNSTORE:      ;
1063$          TRACE,        -1,"-Waiting for resource FINISH OPERATORS\n";
1172$          QUEUE,        FINISH OPERATORS_Q;
```

```
1168$          SEIZE,        1:SELECT(FINISH OPERATORS,CYC,),1;
1097$          STORE:        FINISHING OFF_S3;
               TRACE,        -1,"-Delay for processing time TRIA(15,20,22)\n";
1156$          DELAY:        TRIA(15,20,22);
1133$          UNSTORE:      ;
1126$          STORE:        FINISHING OFF_S4;
1060$          TRACE,        -1,"-Releasing resource\n";
1148$          RELEASE:      SELECT(FINISH OPERATORS,,),1;
1113$          DELAY:        0.;
1110$          UNSTORE:      ;
1054$          TRACE,        -1,"-Transferred to station CONNECTING\n";
1111$          ROUTE:        5,CONNECTING;
21$            STATION,      WINDER_2;
1221$          TRACE,        -1,"-Arrived to station WINDER_2\n";
1341$          STORE:        WINDER_2_S1;
1342$          DELAY:        0.;
1326$          UNSTORE:      ;
1214$          TRACE,        -1,"-Waiting for resource WINDER_2_R\n";
1323$          QUEUE,        WINDER_2_R_Q;
1320$          SEIZE,        1:WINDER_2_R,1;
1248$          STORE:        WINDER_2_S3;
               TRACE,        -1,"-Delay for processing time TRIA(14,20,25)\n";
1307$          DELAY:        TRIA(14,20,25);
1284$          UNSTORE:      ;
1277$          STORE:        WINDER_2_S4;
1211$          TRACE,        -1,"-Releasing resource\n";
1300$          RELEASE:      WINDER_2_R,1;
1264$          DELAY:        0.;
1261$          UNSTORE:      ;
1205$          TRACE,        -1,"-Transferred to station FINISHING OFF\n";
1262$          ROUTE:        4,FINISHING OFF;
23$            STATION,      WINDER_1;
1372$          TRACE,        -1,"-Arrived to station WINDER_1\n";
1492$          STORE:        WINDER_1_S1;
1493$          DELAY:        0.;
1477$          UNSTORE:      ;
1365$          TRACE,        -1,"-Waiting for resource WINDER_1_R\n";
1474$          QUEUE,        WINDER_1_R_Q;
1471$          SEIZE,        1:WINDER_1_R,1;
1399$          STORE:        WINDER_1_S3;
               TRACE,        -1,"-Delay for processing time TRIA(2,3,5)\n";
1458$          DELAY:        TRIA(2,3,5);
1435$          UNSTORE:      ;
1428$          STORE:        WINDER_1_S4;
1362$          TRACE,        -1,"-Releasing resource\n";
1451$          RELEASE:      WINDER_1_R,1;
1415$          DELAY:        0.;
1412$          UNSTORE:      ;
1356$          TRACE,        -1,"-Transferred to station FINISHING OFF\n";
1413$          ROUTE:        5,FINISHING OFF;
28$            STATION,      CLEATER;
1589$          TRACE,        -1,"-Arrived to station CLEATER\n";
1710$          DELAY:        0.;
1584$          TRACE,        -1,"-Freeing transporter\n";
1704$          FREE:         ;
1582$          TRACE,        -1,"-Waiting for resource CLEATER_R\n";
1691$          QUEUE,        CLEATER_R_Q,10,1621$;
1688$          SEIZE,        1:CLEATER_R,1;
1616$          STORE:        CLEATER_S3;
               TRACE,        -1,"-Delay for processing time TRIA(.33,.5,.8)\n";
1675$          DELAY:        TRIA(.33,.5,.8);
1652$          UNSTORE:      ;
1645$          STORE:        CLEATER_S4;
1579$          TRACE,        -1,"-Releasing resource\n";
1668$          RELEASE:      CLEATER_R,1;
1632$          DELAY:        0.;
1629$          UNSTORE:      ;
1572$          TRACE,        -1,"-Transferred to next module\n";
1625$          DELAY:        0.00:NEXT(27$);

1621$          DISPOSE;
27$            TRACE,        -1,"-Choosing from 2 options\n";
1711$          BRANCH,       1:With,.80,26$,Yes:
                             With,.20,25$,Yes;
26$            DELAY:        0.00;
1726$          ASSIGN:       M=CLEATER;
1742$          DELAY:        0.;
1715$          TRACE,        -1,"-Transferred to station WINDER_1\n";
1740$          ROUTE:        3,WINDER_1;
25$            DELAY:        0.00;
```

```
1768$          ASSIGN:        M=CLEATER;
1784$          DELAY:         0.;
1757$          TRACE,         -1,"-Transferred to station WINDER_2\n";
1782$          ROUTE:         5,WINDER_2;
1820$          CREATE,        1:1:MARK(TIME IN SYSTEM);
1860$          STATION,       ARR_LAM;
1811$          TRACE,         -1,"-Arrived to system at station ARR_LAM\n";
1857$          ASSIGN:        Picture=Default;
1802$          TRACE,         -1,"-Waiting for transporter LAM_LABOUR\n";
1839$          REQUEST,       1:LAM_LABOUR(CYC);
1836$          DELAY:         0.;
1805$          TRACE,         -1,"-Transferred to station CLEATER\n";
1832$          TRANSPORT,     :,CLEATER;
```

E.16. Experiment Listing for Windings Assembly (Cell-100)

```
PROJECT,        ;

ATTRIBUTES:     TIME IN SYSTEM;

STORAGES:       CLEATER_S4:
                CLEATER_S3:
                WINDER_1_S4:
                WINDER_1_S3:
                WINDER_1_S1:
                WINDER_2_S4:
                WINDER_2_S3:
                WINDER_2_S1:
                FINISHING OFF_S4:
                FINISHING OFF_S3:
                FINISHING OFF_S1:
                PRESSING_S4:
                PRESSING_S3:
                PRESSING_S1:
                CONNECTING_S4:
                CONNECTING_S3:
                CONNECTING_S1:
                BATCHING_WIND_S1:
                DIP_WYVERT_S4:
                DIP_WYVERT_S3:
                DIP_WYVERT_S1:
                E_FINI_WOUND_P_S1;

QUEUES:         CLEATER_R_Q,FIFO:
                WINDER_1_R_Q,FIFO:
                WINDER_2_R_Q,FIFO:
                FINISH OPERATORS_Q,FIFO:
                F_OFF_OPERA._1_Q,FIFO:
                F_OFF_OPERA._2_Q,FIFO:
                F_OFF_OPERA._3_Q,FIFO:
                F_OFF_OPERA._4_Q,FIFO:
                F_OFF_OPERA._5_Q,FIFO:
                PRESSING_R_Q,FIFO:
                CONNECTORS_Q,FIFO:
                CONNECTING_1_Q,FIFO:
                CONNECTING_2_Q,FIFO:
                CONNECTING_4_Q,FIFO:
                CONNECTING_3_Q,FIFO:
                CONNECTING_5_Q,FIFO:
                LACING_R_Q,FIFO:
                ELCT_TEST_R_Q,FIFO:
                DIP_WYVERT_R_Q,FIFO;

PICTURES:       Default;

RESOURCES:      CLEATER_R,Capacity(1,):
                WINDER_1_R,Capacity(1,):
                WINDER_2_R,Capacity(1,):
                F_OFF_OPERA._1,Capacity(1,):
                F_OFF_OPERA._2,Capacity(1,):
                F_OFF_OPERA._3,Capacity(1,):
                F_OFF_OPERA._4,Capacity(1,):
                F_OFF_OPERA._5,Capacity(1,):
                PRESSING_R,Capacity(1,):
                CONNECTING_1,Capacity(1,):
                CONNECTING_2,Capacity(1,):
```

```
               CONNECTING_3,Capacity(1,):
               CONNECTING_4,Capacity(1,):
               CONNECTING_5,Capacity(1,):
               LACING_R,Capacity(1,):
               ELCT_TEST_R,Capacity(1,):
               DIP_WYVERT_R,Capacity(1,);

STATIONS:      ARR_LAM:
               CLEATER:
               WINDER_1:
               WINDER_2:
               FINISHING OFF:
               PRESSING:
               LACING:
               CONNECTING:
               ELCT_TEST:
               BATCHING_WIND:
               REJECT_WIND:
               DIP_WYVERT:
               E_FINI_WOUND_P:
               FINI_WOUND_P;

DISTANCES:     LAM_LABOUR_Dst,ARR_LAM-CLEATER-2;

TRANSPORTERS: LAM_LABOUR,1,DISTANCE(LAM_LABOUR_Dst),1.;

COUNTERS:      REJECT_WIND_C:
               FINI_WOUND_P_C,,Replicate,"C:\BH\SIM\WIND\CO100.001";

TALLIES:       FINI_WOUND_P_Ta,"C:\BH\SIM\WIND\TA100.001";

DSTATS:        MT(LAM_LABOUR),LAM_LABOUR Active:
               NT(LAM_LABOUR),LAM_LABOUR Busy:
               MR(CLEATER_R),CLEATER_R Available:
               NR(CLEATER_R),CLEATER_R Busy:
               NQ(CLEATER_R_Q),# in CLEATER_R_Q:
               MR(WINDER_1_R),WINDER_1_R Available:
               NR(WINDER_1_R),WINDER_1_R Busy:
               NQ(WINDER_1_R_Q),# in WINDER_1_R_Q:
               MR(WINDER_2_R),WINDER_2_R Available:
               NR(WINDER_2_R),WINDER_2_R Busy:
               NQ(WINDER_2_R_Q),# in WINDER_2_R_Q:
               NQ(FINISH OPERATORS_Q),# in FINISH OPERATORS_Q:
               MR(F_OFF_OPERA._1),F_OFF_OPERA._1 Available:
               NR(F_OFF_OPERA._1),F_OFF_OPERA._1 Busy:
               NQ(F_OFF_OPERA._1_Q),# in F_OFF_OPERA._1_Q:
               MR(F_OFF_OPERA._2),F_OFF_OPERA._2 Available:
               NR(F_OFF_OPERA._2),F_OFF_OPERA._2 Busy:
               NQ(F_OFF_OPERA._2_Q),# in F_OFF_OPERA._2_Q:
               MR(F_OFF_OPERA._3),F_OFF_OPERA._3 Available:
               NR(F_OFF_OPERA._3),F_OFF_OPERA._3 Busy:
               NQ(F_OFF_OPERA._3_Q),# in F_OFF_OPERA._3_Q:
               MR(F_OFF_OPERA._4),F_OFF_OPERA._4 Available:
               NR(F_OFF_OPERA._4),F_OFF_OPERA._4 Busy:
               NQ(F_OFF_OPERA._4_Q),# in F_OFF_OPERA._4_Q:
               MR(F_OFF_OPERA._5),F_OFF_OPERA._5 Available:
               NR(F_OFF_OPERA._5),F_OFF_OPERA._5 Busy:
               NQ(F_OFF_OPERA._5_Q),# in F_OFF_OPERA._5_Q:
               MR(PRESSING_R),PRESSING_R Available:
               NR(PRESSING_R),PRESSING_R Busy:
               NQ(PRESSING_R_Q),# in PRESSING_R_Q:
               NQ(CONNECTORS_Q),# in CONNECTORS_Q:
               MR(CONNECTING_1),CONNECTING_1 Available:
               NR(CONNECTING_1),CONNECTING_1 Busy:
               NQ(CONNECTING_1_Q),# in CONNECTING_1_Q:
               MR(CONNECTING_2),CONNECTING_2 Available:
               NR(CONNECTING_2),CONNECTING_2 Busy:
               NQ(CONNECTING_2_Q),# in CONNECTING_2_Q:
               MR(CONNECTING_4),CONNECTING_4 Available:
               NR(CONNECTING_4),CONNECTING_4 Busy:
               NQ(CONNECTING_4_Q),# in CONNECTING_4_Q:
               MR(CONNECTING_3),CONNECTING_3 Available:
               NR(CONNECTING_3),CONNECTING_3 Busy:
               NQ(CONNECTING_3_Q),# in CONNECTING_3_Q:
               MR(CONNECTING_5),CONNECTING_5 Available:
               NR(CONNECTING_5),CONNECTING_5 Busy:
               NQ(CONNECTING_5_Q),# in CONNECTING_5_Q:
               MR(LACING_R),LACING_R Available:
               NR(LACING_R),LACING_R Busy:
               NQ(LACING_R_Q),# in LACING_R_Q:
```

```
                MR(ELCT_TEST_R),ELCT_TEST_R Available:
                NR(ELCT_TEST_R),ELCT_TEST_R Busy:
                NQ(ELCT_TEST_R_Q),# in ELCT_TEST_R_Q:
                MR(DIP_WYVERT_R),DIP_WYVERT_R Available:
                NR(DIP_WYVERT_R),DIP_WYVERT_R Busy:
                NQ(DIP_WYVERT_R_Q),# in DIP_WYVERT_R_Q;

REPLICATE,      5,0.0,2865,Yes,Yes;

SETS:           FINISH
OPERATORS,F_OFF_OPERA._1,F_OFF_OPERA._2,F_OFF_OPERA._3,F_OFF_OPERA._4,F_OFF_OPERA._5:

CONNECTORS,CONNECTING_1,CONNECTING_2,CONNECTING_3,CONNECTING_4,CONNECTING_5;
```

E.17. Model Listing for Windings Assembly (Cell-112)

```
2$              STATION,        ENT_FINI_PACK;
30$             TRACE,          -1,"-Arrived to station ENT_FINI_PACK\n";
50$             STORE:          ENT_FINI_PACK_S1;
51$             DELAY:          0.;
36$             UNSTORE:        ;
31$             DELAY:          0.000:NEXT(1$);

1$              TRACE,          -1,"-Splitting group representative\n";
52$             SPLIT:          A(*),M,NS,IS:NEXT(0$);

0$              STATION,        FINI_WOUND_PACK;
60$             TRACE,          -1,"-Arrived to station FINI_WOUND_PACK\n";
90$             DELAY:          0.;
68$             COUNT:          FINI_WOUND_PACK_C,1;
63$             TALLY:          FINI_WOUND_PACK_Ta,Interval(TIME IN),1;
53$             TRACE,          -1,"-Disposing entity\n";
61$             DISPOSE;
3$              STATION,        DIP_PAKE_WYVER;
116$            TRACE,          -1,"-Arrived to station DIP_PAKE_WYVER\n";
153$            DELAY:          0.;
109$            TRACE,          -1,"-Waiting for resource DIP_PAKE_WYVER_R\n";
194$            QUEUE,          DIP_PAKE_WYVER_R_Q;
193$            SEIZE,          1:DIP_PAKE_WYVER_R,1;
                TRACE,          -1,"-Delay for processing time TRIA(90,100,110)\n";
192$            DELAY:          TRIA(90,100,110);
108$            TRACE,          -1,"-Releasing resource\n";
191$            RELEASE:        DIP_PAKE_WYVER_R,1;
125$            DELAY:          0.;
103$            TRACE,          -1,"-Transferred to station ENT_FINI_PACK\n";
186$            ROUTE:          4,ENT_FINI_PACK;
8$              STATION,        BATCHING;
201$            TRACE,          -1,"-Arrived to station BATCHING\n";
222$            DELAY:          0.;
195$            TRACE,          -1,"-Exiting conveyor\n";
211$            EXIT:           ;
202$            DELAY:          0.000:NEXT(7$);

7$              DELAY:          0.00;
225$            TRACE,          -1,"-Waiting for batch size of 20\n";
233$            GROUP,          :20,Last;
227$            TRACE,          -1,"-Entity created to represent temporary batch of size
20\n";
231$            DELAY:          0.000:NEXT(5$);

5$              DELAY:          0.00;
248$            ASSIGN:         M=BATCHING;
264$            DELAY:          0.;
237$            TRACE,          -1,"-Transferred to station DIP_PAKE_WYVER\n";
262$            ROUTE:          4,DIP_PAKE_WYVER;
9$              STATION,        REJ_WOUND_P;
283$            TRACE,          -1,"-Arrived to station REJ_WOUND_P\n";
313$            DELAY:          0.;
291$            COUNT:          REJ_WOUND_P_C,1;
276$            TRACE,          -1,"-Disposing entity\n";
284$            DISPOSE;
10$             STATION,        ELECT_TEST;
354$            TRACE,          -1,"-Arrived to Inspect station ELECT_TEST\n";
413$            DELAY:          0.;
348$            TRACE,          -1,"-Freeing transporter\n";
414$            EXIT:           ;
```

```
469$           QUEUE,         ELECT_TEST_R_Q;
468$           SEIZE,         1:ELECT_TEST_R,1;
               TRACE,         -1,"-Delay for processing time TRIA(3,3.5,4.2) with failure
probability .05\n";
467$           DELAY:         TRIA(3,3.5,4.2);
457$           BRANCH,        1:With,.05,346$,Yes:
                              Else,347$,Yes;
346$           TRACE,         -1,"-Entity failed inspection\n";
361$           DELAY:         0.0;
402$           RELEASE:       ELECT_TEST_R,1;
377$           DELAY:         0.;
342$           TRACE,         -1,"-Transferred to station REJ_WOUND_P\n";
459$           ROUTE:         1,REJ_WOUND_P;
347$           TRACE,         -1,"-Entity passed inspection\n";
362$           DELAY:         0.0;
466$           RELEASE:       ELECT_TEST_R,1;
332$           TRACE,         -1,"-Waiting for conveyor CONV2\n";
382$           ACCESS:        CONV2,1;
378$           DELAY:         0.;
336$           TRACE,         -1,"-Transferring to station BATCHING\n";
368$           CONVEY:        ,BATCHING;
12$            STATION,       LACING;
495$           TRACE,         -1,"-Arrived to station LACING\n";
532$           DELAY:         0.;
488$           TRACE,         -1,"-Waiting for resource LACING_R\n";
573$           QUEUE,         LACING_R_Q;
572$           SEIZE,         1:LACING_R,1;
               TRACE,         -1,"-Delay for processing time TRIA(.6,.7,.8)\n";
571$           DELAY:         TRIA(.6,.7,.8);
487$           TRACE,         -1,"-Releasing resource\n";
570$           RELEASE:       LACING_R,1;
478$           TRACE,         -1,"-Waiting for conveyor CONV2\n";
512$           ACCESS:        CONV2,1;
504$           DELAY:         0.;
482$           TRACE,         -1,"-Transferred to station ELECT_TEST\n";
521$           CONVEY:        ,ELECT_TEST;
14$            STATION,       PRESSING;
603$           TRACE,         -1,"-Arrived to station PRESSING\n";
724$           DELAY:         0.;
597$           TRACE,         -1,"-Exiting conveyor\n";
712$           EXIT:          ;
596$           TRACE,         -1,"-Waiting for resource PRESSING_R\n";
705$           QUEUE,         PRESSING_R_Q;
702$           SEIZE,         1:PRESSING_R,1;
630$           STORE:         PRESSING_S3;
               TRACE,         -1,"-Delay for processing time TRIA(.5,.8,1.3)\n";
689$           DELAY:         TRIA(.5,.8,1.3);
666$           UNSTORE:       ;
659$           STORE:         PRESSING_S4;
593$           TRACE,         -1,"-Releasing resource\n";
682$           RELEASE:       PRESSING_R,1;
646$           DELAY:         0.;
643$           UNSTORE:       ;
587$           TRACE,         -1,"-Transferred to station LACING\n";
644$           ROUTE:         .2,LACING;
16$            STATION,       CONNECTING;
754$           TRACE,         -1,"-Arrived to station CONNECTING\n";
875$           DELAY:         0.;
748$           TRACE,         -1,"-Exiting conveyor\n";
863$           EXIT:          ;
747$           TRACE,         -1,"-Waiting for resource CONNECTING_R\n";
856$           QUEUE,         CONNECTING_R_Q;
853$           SEIZE,         1:CONNECTING_R,1;
781$           STORE:         CONNECTING_S3;
               TRACE,         -1,"-Delay for processing time TRIA(10,15,20)\n";
840$           DELAY:         TRIA(10,15,20);
817$           UNSTORE:       ;
810$           STORE:         CONNECTING_S4;
744$           TRACE,         -1,"-Releasing resource\n";
833$           RELEASE:       CONNECTING_R,1;
734$           TRACE,         -1,"-Waiting for conveyor CONV1\n";
799$           ACCESS:        CONV1,1;
797$           DELAY:         0.;
794$           UNSTORE:       ;
738$           TRACE,         -1,"-Transferred to station PRESSING\n";
789$           CONVEY:        ,PRESSING;
18$            STATION,       FINISHING OFF;
905$           TRACE,         -1,"-Arrived to station FINISHING OFF\n";
1026$          DELAY:         0.;
899$           TRACE,         -1,"-Exiting conveyor\n";
```

```
1014$     EXIT:       ;
898$      TRACE,      -1,"-Waiting for resource FINISHING OFF_R\n";
1007$     QUEUE,      FINISHING OFF_R_Q;
1004$     SEIZE,      1:FINISHING OFF_R,1;
932$      STORE:      FINISHING OFF_S3;
          TRACE,      -1,"-Delay for processing time TRIA(20,25,30)\n";
991$      DELAY:      TRIA(20,25,30);
968$      UNSTORE:    ;
961$      STORE:      FINISHING OFF_S4;
895$      TRACE,      -1,"-Releasing resource\n";
984$      RELEASE:    FINISHING OFF_R,1;
885$      TRACE,      -1,"-Waiting for conveyor CONV1\n";
950$      ACCESS:     CONV1,1;
948$      DELAY:      0.;
945$      UNSTORE:    ;
889$      TRACE,      -1,"-Transferred to station CONNECTING\n";
940$      CONVEY:     ,CONNECTING;
20$       STATION,    INSULATING;
1056$     TRACE,      -1,"-Arrived to station INSULATING\n";
1176$     STORE:      INSULATING_S1;
1177$     DELAY:      0.;
1161$     UNSTORE:    ;
1049$     TRACE,      -1,"-Waiting for resource INSULATING_R\n";
1158$     QUEUE,      INSULATING_R_Q,50,1088$;
1155$     SEIZE,      1:INSULATING_R,1;
1083$     STORE:      INSULATING_S3;
          TRACE,      -1,"-Delay for processing time TRIA(.25,.33,.7)\n";
1142$     DELAY:      TRIA(.25,.33,.7);
1119$     UNSTORE:    ;
1112$     STORE:      INSULATING_S4;
1046$     TRACE,      -1,"-Releasing resource\n";
1135$     RELEASE:    INSULATING_R,1;
1099$     DELAY:      0.;
1096$     UNSTORE:    ;
1040$     TRACE,      -1,"-Transferred to station WINDING\n";
1097$     ROUTE:      2,WINDING;
1088$     DISPOSE;
22$       STATION,    WINDING;
1207$     TRACE,      -1,"-Arrived to station WINDING\n";
1327$     STORE:      WINDING_S1;
1328$     DELAY:      0.;
1312$     UNSTORE:    ;
1200$     TRACE,      -1,"-Waiting for resource WINDING_R\n";
1309$     QUEUE,      WINDING_R_Q,40,1239$;
1306$     SEIZE,      1:WINDING_R,1;
1234$     STORE:      WINDING_S3;
          TRACE,      -1,"-Delay for processing time TRIA(4,6,8)\n";
1293$     DELAY:      TRIA(4,6,8);
1270$     UNSTORE:    ;
1263$     STORE:      WINDING_S4;
1197$     TRACE,      -1,"-Releasing resource\n";
1286$     RELEASE:    WINDING_R,1;
1187$     TRACE,      -1,"-Waiting for conveyor CONV1\n";
1252$     ACCESS:     CONV1,1;
1250$     DELAY:      0.;
1247$     UNSTORE:    ;
1191$     TRACE,      -1,"-Transferred to station FINISHING OFF\n";
1242$     CONVEY:     ,FINISHING OFF;
1239$     DISPOSE;
1353$     CREATE,     20:EXPO(40):MARK(TIME IN);
1393$     STATION,    ARR_CLEAT;
1344$     TRACE,      -1,"-Arrived to system at station ARR_CLEAT\n";
1390$     ASSIGN:     Picture=Default;
1369$     DELAY:      0.;
1338$     TRACE,      -1,"-Transferred to station INSULATING\n";
1367$     ROUTE:      2,INSULATING;
```

E.18. Experiment Listing for Windings Assembly (Cell-112)

```
PROJECT,      ;

ATTRIBUTES:   TIME IN;

SCHEDULES:    SCHED1,1*500,0*60;

STORAGES:     INSULATING_S4:
```

```
               INSULATING_S3:
               INSULATING_S1:
               WINDING_S4:
               WINDING_S3:
               WINDING_S1:
               FINISHING OFF_S4:
               FINISHING OFF_S3:
               CONNECTING_S4:
               CONNECTING_S3:
               PRESSING_S4:
               PRESSING_S3:
               ENT_FINI_PACK_S1;

QUEUES:        INSULATING_R_Q,FIFO:
               WINDING_R_Q,FIFO:
               CONNECTING_R_Q,FIFO:
               PRESSING_R_Q,FIFO:
               LACING_R_Q,FIFO:
               ELECT_TEST_R_Q,FIFO:
               DIP_PAKE_WYVER_R_Q,FIFO:
               FINISHING OFF_R_Q,FIFO;

PICTURES:      Default;

RESOURCES:     INSULATING_R,Capacity(1,):
               WINDING_R,Schedule(SCHED1,Ignore):
               FINISHING OFF_R,Capacity(5,):
               CONNECTING_R,Capacity(5,):
               PRESSING_R,Capacity(5,):
               LACING_R,Capacity(1,):
               ELECT_TEST_R,Capacity(1,):
               DIP_PAKE_WYVER_R,Capacity(1,);

STATIONS:      ARR_CLEAT:
               INSULATING:
               WINDING:
               FINISHING OFF:
               CONNECTING:
               PRESSING:
               LACING:
               ELECT_TEST:
               BATCHING:
               REJ_WOUND_P:
               DIP_PAKE_WYVER:
               ENT_FINI_PACK:
               FINI_WOUND_PACK;

SEGMENTS:      CONV1_Seg,WINDING,FINISHING OFF-5,
               CONNECTING-5,
               PRESSING-5:
               CONV2_Seg,LACING,ELECT_TEST-4,
               BATCHING-8;

CONVEYORS:     CONV1,CONV1_Seg,2,1,Active,1,Accumulating:
               CONV2,CONV2_Seg,1,1,Active,1,Accumulating;

COUNTERS:      REJ_WOUND_P_C:
               FINI_WOUND_PACK_C,,Replicate,"C:\BH\SIM\WIND\CO112.001";

TALLIES:       FINI_WOUND_PACK_Ta,"C:\BH\SIM\WIND\TA112.001";

DSTATS:        MR(INSULATING_R),INSULATING_R Available:
               NR(INSULATING_R),INSULATING_R Busy:
               NQ(INSULATING_R_Q),# in INSULATING_R_Q:
               MR(WINDING_R),WINDING_R Available:
               NR(WINDING_R),WINDING_R Busy:
               NQ(WINDING_R_Q),# in WINDING_R_Q:
               MR(CONNECTING_R),CONNECTING_R Available:
               NR(CONNECTING_R),CONNECTING_R Busy:
               NQ(CONNECTING_R_Q),# in CONNECTING_R_Q:
               MR(PRESSING_R),PRESSING_R Available:
               NR(PRESSING_R),PRESSING_R Busy:
               NQ(PRESSING_R_Q),# in PRESSING_R_Q:
               MR(LACING_R),LACING_R Available:
               NR(LACING_R),LACING_R Busy:
               NQ(LACING_R_Q),# in LACING_R_Q:
               MR(ELECT_TEST_R),ELECT_TEST_R Available:
               NR(ELECT_TEST_R),ELECT_TEST_R Busy:
               NQ(ELECT_TEST_R_Q),# in ELECT_TEST_R_Q:
               MR(DIP_PAKE_WYVER_R),DIP_PAKE_WYVER_R Available:
```

```
               NR(DIP_PAKE_WYVER_R),DIP_PAKE_WYVER_R Busy:
               NQ(DIP_PAKE_WYVER_R_Q),# in DIP_PAKE_WYVER_R_Q:
               MR(FINISHING OFF_R),FINISHING OFF_R Available:
               NR(FINISHING OFF_R),FINISHING OFF_R Busy:
               NQ(FINISHING OFF_R_Q),# in FINISHING OFF_R_Q;

REPLICATE,     5,0.0,3100,Yes,Yes;
```

E.19. Model Listing for Windings Assembly (Cell-132)

```
2$             STATION,       ENT_FINI_BAT;
34$            TRACE,         -1,"-Arrived to station ENT_FINI_BAT\n";
54$            STORE:         ENT_FINI_BAT_S1;
55$            DELAY:         0.;
40$            UNSTORE:       ;
35$            DELAY:         0.000:NEXT(1$);

1$             TRACE,         -1,"-Splitting group representative\n";
56$            SPLIT:         A(*),M,NS,IS:NEXT(0$);

0$             STATION,       FINISHED_W_PACK;
64$            TRACE,         -1,"-Arrived to station FINISHED_W_PACK\n";
94$            DELAY:         0.;
72$            COUNT:         FINISHED_W_PACK_C,1;
67$            TALLY:         FINISHED_W_PACK_Ta,Interval(TIME IN SYSTEM),1;
57$            TRACE,         -1,"-Disposing entity\n";
65$            DISPOSE;
3$             STATION,       DIP_BAKE_WYVERTIC;
120$           TRACE,         -1,"-Arrived to station DIP_BAKE_WYVERTIC\n";
157$           DELAY:         0.;
113$           TRACE,         -1,"-Waiting for resource DIP_BAKE_WYVERTIC_R\n";
198$           QUEUE,         DIP_BAKE_WYVERTIC_R_Q;
197$           SEIZE,         1:DIP_BAKE_WYVERTIC_R,1;
               TRACE,         -1,"-Delay for processing time TRIA(90,100,110)\n";
196$           DELAY:         TRIA(90,100,110);
112$           TRACE,         -1,"-Releasing resource\n";
195$           RELEASE:       DIP_BAKE_WYVERTIC_R,1;
129$           DELAY:         0.;
107$           TRACE,         -1,"-Transferred to station ENT_FINI_BAT\n";
190$           ROUTE:         5,ENT_FINI_BAT;
8$             STATION,       BATCHING;
205$           TRACE,         -1,"-Arrived to station BATCHING\n";
225$           STORE:         BATCHING_S1;
226$           DELAY:         0.;
211$           UNSTORE:       ;
206$           DELAY:         0.000:NEXT(7$);

7$             DELAY:         0.00;
229$           TRACE,         -1,"-Waiting for batch size of 20\n";
237$           GROUP,         :20,Last;
231$           TRACE,         -1,"-Entity created to represent temporary batch of size
20\n";
235$           DELAY:         0.000:NEXT(5$);

5$             DELAY:         0.00;
252$           ASSIGN:        M=BATCHING;
268$           DELAY:         0.;
241$           TRACE,         -1,"-Transferred to station DIP_BAKE_WYVERTIC\n";
266$           ROUTE:         5,DIP_BAKE_WYVERTIC;
9$             STATION,       REJ_WOUND_P;
287$           TRACE,         -1,"-Arrived to station REJ_WOUND_P\n";
317$           DELAY:         0.;
295$           COUNT:         REJ_WOUND_P_C,1;
280$           TRACE,         -1,"-Disposing entity\n";
288$           DISPOSE;
10$            STATION,       ELECT_TEST;
358$           TRACE,         -1,"-Arrived to Inspect station ELECT_TEST\n";
417$           DELAY:         0.;
473$           QUEUE,         ELECT_TEST_R_Q;
472$           SEIZE,         1:ELECT_TEST_R,1;
               TRACE,         -1,"-Delay for processing time TRIA(2,2.3,2.5) with failure
probability .05\n";
471$           DELAY:         TRIA(2,2.3,2.5);
461$           BRANCH,        1:With,.05,350$,Yes:
                              Else,351$,Yes;
350$           TRACE,         -1,"-Entity failed inspection\n";
```

```
365$        DELAY:       0.0;
406$        RELEASE:     ELECT_TEST_R,1;
381$        DELAY:       0.;
346$        TRACE,       -1,"-Transferred to station REJ_WOUND_P\n";
463$        ROUTE:       1,REJ_WOUND_P;
351$        TRACE,       -1,"-Entity passed inspection\n";
366$        DELAY:       0.0;
470$        RELEASE:     ELECT_TEST_R,1;
382$        DELAY:       0.;
340$        TRACE,       -1,"-Transferring to station BATCHING\n";
465$        ROUTE:       4,BATCHING;
12$         STATION,     LACING;
499$        TRACE,       -1,"-Arrived to station LACING\n";
536$        DELAY:       0.;
492$        TRACE,       -1,"-Waiting for resource LACING_R\n";
577$        QUEUE,       LACING_R_Q;
576$        SEIZE,       1:LACING_R,1;
            TRACE,       -1,"-Delay for processing time TRIA(1.2,1.23,1.25)\n";
575$        DELAY:       TRIA(1.2,1.23,1.25);
491$        TRACE,       -1,"-Releasing resource\n";
574$        RELEASE:     LACING_R,1;
508$        DELAY:       0.;
486$        TRACE,       -1,"-Transferred to station ELECT_TEST\n";
569$        ROUTE:       2,ELECT_TEST;
14$         STATION,     PRESSING;
603$        TRACE,       -1,"-Arrived to station PRESSING\n";
640$        DELAY:       0.;
596$        TRACE,       -1,"-Waiting for resource PRESSING_R\n";
681$        QUEUE,       PRESSING_R_Q;
680$        SEIZE,       1:PRESSING_R,1;
            TRACE,       -1,"-Delay for processing time TRIA(.5,.7,.8)\n";
679$        DELAY:       TRIA(.5,.7,.8);
595$        TRACE,       -1,"-Releasing resource\n";
678$        RELEASE:     PRESSING_R,1;
612$        DELAY:       0.;
590$        TRACE,       -1,"-Transferred to station LACING\n";
673$        ROUTE:       1,LACING;
16$         STATION,     CONNECTION;
707$        TRACE,       -1,"-Arrived to station CONNECTION\n";
744$        DELAY:       0.;
700$        TRACE,       -1,"-Waiting for resource CONNECTION_R\n";
785$        QUEUE,       CONNECTION_R_Q;
784$        SEIZE,       1:CONNECTION_R,1;
            TRACE,       -1,"-Delay for processing time TRIA(8,12,16)\n";
783$        DELAY:       TRIA(8,12,16);
699$        TRACE,       -1,"-Releasing resource\n";
782$        RELEASE:     CONNECTION_R,1;
716$        DELAY:       0.;
694$        TRACE,       -1,"-Transferred to station PRESSING\n";
777$        ROUTE:       2,PRESSING;
18$         STATION,     FINISHING OFF;
811$        TRACE,       -1,"-Arrived to station FINISHING OFF\n";
848$        DELAY:       0.;
804$        TRACE,       -1,"-Waiting for resource FINISHING OFF_R\n";
889$        QUEUE,       FINISHING OFF_R_Q;
888$        SEIZE,       1:FINISHING OFF_R,1;
            TRACE,       -1,"-Delay for processing time TRIA(20,25,30)\n";
887$        DELAY:       TRIA(20,25,30);
803$        TRACE,       -1,"-Releasing resource\n";
886$        RELEASE:     FINISHING OFF_R,1;
820$        DELAY:       0.;
798$        TRACE,       -1,"-Transferred to station CONNECTION\n";
881$        ROUTE:       1,CONNECTION;
20$         STATION,     WINDINGS;
915$        TRACE,       -1,"-Arrived to station WINDINGS\n";
952$        DELAY:       0.;
908$        TRACE,       -1,"-Waiting for resource WINDINGS_R\n";
993$        QUEUE,       WINDINGS_R_Q,150,973$;
992$        SEIZE,       1:WINDINGS_R,1;
            TRACE,       -1,"-Delay for processing time TRIA(12.5,15,20)\n";
991$        DELAY:       TRIA(12.5,15,20);
907$        TRACE,       -1,"-Releasing resource\n";
990$        RELEASE:     WINDINGS_R,1;
924$        DELAY:       0.;
902$        TRACE,       -1,"-Transferred to station FINISHING OFF\n";
985$        ROUTE:       3,FINISHING OFF;
973$        DISPOSE;
22$         STATION,     INSULATORS;
1019$       TRACE,       -1,"-Arrived to station INSULATORS\n";
1056$       DELAY:       0.;
```

```
1012$      TRACE,      -1,"-Waiting for resource INSULATORS_R\n";
1097$      QUEUE,      INSULATORS_R_Q;
1096$      SEIZE,      1:INSULATORS_R,1;
           TRACE,      -1,"-Delay for processing time TRIA(.25,.33,.6)\n";
1095$      DELAY:      TRIA(.25,.33,.6);
1011$      TRACE,      -1,"-Releasing resource\n";
1094$      RELEASE:    INSULATORS_R,1;
1028$      DELAY:      0.;
1006$      TRACE,      -1,"-Transferred to station WINDINGS\n";
1089$      ROUTE:      4,WINDINGS;
24$        STATION,    CLEATING;
1123$      TRACE,      -1,"-Arrived to station CLEATING\n";
1160$      DELAY:      0.;
1116$      TRACE,      -1,"-Waiting for resource CLEATING_R\n";
1201$      QUEUE,      CLEATING_R_Q,50,1181$;
1200$      SEIZE,      1:CLEATING_R,1;
           TRACE,      -1,"-Delay for processing time TRIA(1,1.2,1.3)\n";
1199$      DELAY:      TRIA(1,1.2,1.3);
1115$      TRACE,      -1,"-Releasing resource\n";
1198$      RELEASE:    CLEATING_R,1;
1132$      DELAY:      0.;
1110$      TRACE,      -1,"-Transferred to station INSULATORS\n";
1193$      ROUTE:      1,INSULATORS;
1181$      DISPOSE;
26$        STATION,    SCALING;
1227$      TRACE,      -1,"-Arrived to station SCALING\n";
1264$      DELAY:      0.;
1220$      TRACE,      -1,"-Waiting for resource SCALING_R\n";
1305$      QUEUE,      SCALING_R_Q,20,1285$;
1304$      SEIZE,      1:SCALING_R,1;
           TRACE,      -1,"-Delay for processing time TRIA(.4,.5,.8)\n";
1303$      DELAY:      TRIA(.4,.5,.8);
1219$      TRACE,      -1,"-Releasing resource\n";
1302$      RELEASE:    SCALING_R,1;
1236$      DELAY:      0.;
1214$      TRACE,      -1,"-Transferred to station CLEATING\n";
1297$      ROUTE:      1,CLEATING;
1285$      DISPOSE;
1330$      CREATE,     1:EXPO(5):MARK(TIME IN SYSTEM);
1370$      STATION,    ARR_LAM;
1321$      TRACE,      -1,"-Arrived to system at station ARR_LAM\n";
1367$      ASSIGN:     Picture=Default;
1346$      DELAY:      0.;
1315$      TRACE,      -1,"-Transferred to station SCALING\n";
1344$      ROUTE:      1,SCALING;
```

E.20. Experiment Listing for Windings Assembly (Cell-132)

```
PROJECT,       ;

ATTRIBUTES:    TIME IN SYSTEM;

SCHEDULES:     SCH1,1*500,0*20;

STORAGES:      BATCHING_S1:
               ENT_FINI_BAT_S1;

QUEUES:        SCALING_R_Q,FIFO:
               CLEATING_R_Q,FIFO:
               INSULATORS_R_Q,FIFO:
               WINDINGS_R_Q,FIFO:
               FINISHING OFF_R_Q,FIFO:
               CONNECTION_R_Q,FIFO:
               PRESSING_R_Q,FIFO:
               LACING_R_Q,FIFO:
               ELECT_TEST_R_Q,FIFO:
               DIP_BAKE_WYVERTIC_R_Q,FIFO;

PICTURES:      Default;

RESOURCES:     SCALING_R,Capacity(1,):
               CLEATING_R,Schedule(SCH1,Ignore):
               INSULATORS_R,Capacity(2,):
               WINDINGS_R,Capacity(3,):
               FINISHING OFF_R,Capacity(7,):
               CONNECTION_R,Capacity(6,):
```

```
                PRESSING_R,Capacity(2,):
                LACING_R,Capacity(1,):
                ELECT_TEST_R,Capacity(1,):
                DIP_BAKE_WYVERTIC_R,Capacity(1,);

STATIONS:       ARR_LAM:
                SCALING:
                CLEATING:
                INSULATORS:
                WINDINGS:
                FINISHING OFF:
                CONNECTION:
                PRESSING:
                LACING:
                ELECT_TEST:
                BATCHING:
                REJ_WOUND_P:
                DIP_BAKE_WYVERTIC:
                ENT_FINI_BAT:
                FINISHED_W_PACK;

COUNTERS:       REJ_WOUND_P_C:
                FINISHED_W_PACK_C,,Replicate,"C:\BH\SIM\WIND\CO132.001";

TALLIES:        FINISHED_W_PACK_Ta,"C:\BH\SIM\WIND\TA132.001";

DSTATS:         MR(SCALING_R),SCALING_R Available:
                NR(SCALING_R),SCALING_R Busy:
                NQ(SCALING_R_Q),# in SCALING_R_Q:
                MR(CLEATING_R),CLEATING_R Available:
                NR(CLEATING_R),CLEATING_R Busy:
                NQ(CLEATING_R_Q),# in CLEATING_R_Q:
                MR(INSULATORS_R),INSULATORS_R Available:
                NR(INSULATORS_R),INSULATORS_R Busy:
                NQ(INSULATORS_R_Q),# in INSULATORS_R_Q:
                MR(WINDINGS_R),WINDINGS_R Available:
                NR(WINDINGS_R),WINDINGS_R Busy:
                NQ(WINDINGS_R_Q),# in WINDINGS_R_Q:
                MR(FINISHING OFF_R),FINISHING OFF_R Available:
                NR(FINISHING OFF_R),FINISHING OFF_R Busy:
                NQ(FINISHING OFF_R_Q),# in FINISHING OFF_R_Q:
                MR(CONNECTION_R),CONNECTION_R Available:
                NR(CONNECTION_R),CONNECTION_R Busy:
                NQ(CONNECTION_R_Q),# in CONNECTION_R_Q:
                MR(PRESSING_R),PRESSING_R Available:
                NR(PRESSING_R),PRESSING_R Busy:
                NQ(PRESSING_R_Q),# in PRESSING_R_Q:
                MR(LACING_R),LACING_R Available:
                NR(LACING_R),LACING_R Busy:
                NQ(LACING_R_Q),# in LACING_R_Q:
                MR(ELECT_TEST_R),ELECT_TEST_R Available:
                NR(ELECT_TEST_R),ELECT_TEST_R Busy:
                NQ(ELECT_TEST_R_Q),# in ELECT_TEST_R_Q:
                MR(DIP_BAKE_WYVERTIC_R),DIP_BAKE_WYVERTIC_R Available:
                NR(DIP_BAKE_WYVERTIC_R),DIP_BAKE_WYVERTIC_R Busy:
                NQ(DIP_BAKE_WYVERTIC_R_Q),# in DIP_BAKE_WYVERTIC_R_Q;

REPLICATE,      5,0.0,3100,Yes,Yes;
```

E.21. Model Listing for Assembly Line (Cell-100)

```
1$        STATION,     BATCHING;
42$       TRACE,       -1,"-Arrived to station BATCHING\n";
72$       DELAY:       0.;
36$       TRACE,       -1,"-Exiting conveyor\n";
62$       EXIT:        ;
50$       COUNT:       FINISH MOTORS_100,1;
45$       TALLY:       BATCHING_Ta,Interval(FLOW TIME),1;
35$       TRACE,       -1,"-Disposing entity\n";
43$       DISPOSE;
2$        STATION,     FAV_FITT;
102$      TRACE,       -1,"-Arrived to station FAV_FITT\n";
222$      STORE:       FAV_FITT_S1;
223$      DELAY:       0.;
207$      UNSTORE:     ;
95$       TRACE,       -1,"-Waiting for resource FAV_FITT_R\n";
```

```
204$            QUEUE,          FAV_FITT_R_Q;
201$            SEIZE,          1:FAV_FITT_R,1;
129$            STORE:          FAV_FITT_S3;
                TRACE,          -1,"-Delay for processing time TRIA(.5,1.2,1.3)\n";
188$            DELAY:          TRIA(.5,1.2,1.3);
165$            UNSTORE:        ;
158$            STORE:          FAV_FITT_S4;
92$             TRACE,          -1,"-Releasing resource\n";
181$            RELEASE:        FAV_FITT_R,1;
145$            DELAY:          0.;
142$            UNSTORE:        ;
86$             TRACE,          -1,"-Transferred to station BATCHING\n";
137$            CONVEY:         ,BATCHING;
4$              STATION,        PAINTING;
253$            TRACE,          -1,"-Arrived to station PAINTING\n";
373$            STORE:          PAINTING_S1;
374$            DELAY:          0.;
358$            UNSTORE:        ;
246$            TRACE,          -1,"-Waiting for resource PAINTING_R\n";
355$            QUEUE,          PAINTING_R_Q;
352$            SEIZE,          1:PAINTING_R,1;
280$            STORE:          PAINTING_S3;
                TRACE,          -1,"-Delay for processing time TRIA(1.5,2,4)\n";
339$            DELAY:          TRIA(1.5,2,4);
316$            UNSTORE:        ;
309$            STORE:          PAINTING_S4;
243$            TRACE,          -1,"-Releasing resource\n";
332$            RELEASE:        PAINTING_R,1;
296$            DELAY:          0.;
293$            UNSTORE:        ;
237$            TRACE,          -1,"-Transferred to station FAV_FITT\n";
288$            CONVEY:         ,FAV_FITT;
6$              STATION,        REJ_MOTORS;
382$            TRACE,          -1,"-Arrived to station REJ_MOTORS\n";
412$            DELAY:          0.;
376$            TRACE,          -1,"-Exiting conveyor\n";
402$            EXIT:           ;
390$            COUNT:          REJ_MOTORS_C,1;
375$            TRACE,          -1,"-Disposing entity\n";
383$            DISPOSE;
8$              STATION,        ELECT_TEST;
453$            TRACE,          -1,"-Arrived to Inspect station ELECT_TEST\n";
512$            DELAY:          0.;
568$            QUEUE,          ELECT_TEST_R_Q;
567$            SEIZE,          1:ELECT_TEST_R,1;
                TRACE,          -1,"-Delay for processing time TRIA(2.2,2.3,2.5) with
failure probability .1\n";
566$            DELAY:          TRIA(2.2,2.3,2.5);
556$            BRANCH,         1:With,.1,445$,Yes:
                                Else,446$,Yes;
445$            TRACE,          -1,"-Entity failed inspection\n";
460$            DELAY:          0.0;
501$            RELEASE:        ELECT_TEST_R,1;
476$            DELAY:          0.;
441$            TRACE,          -1,"-Transferred to station REJ_MOTORS\n";
558$            ROUTE:          2,REJ_MOTORS;
446$            TRACE,          -1,"-Entity passed inspection\n";
461$            DELAY:          0.0;
565$            RELEASE:        ELECT_TEST_R,1;
477$            DELAY:          0.;
434$            TRACE,          -1,"-Transferrinf to next module\n";
559$            DELAY:          0.000:NEXT(7$);

7$              DELAY:          0.00;
583$            ASSIGN:         M=ELECT_TEST;
599$            DELAY:          0.;
572$            TRACE,          -1,"-Transferred to station PAINTING\n";
591$            CONVEY:         ,PAINTING;
14$             STATION,        ENT_END_ST2;
617$            TRACE,          -1,"-Arrived to station ENT_END_ST2\n";
637$            STORE:          ENT_END_ST2_S1;
638$            DELAY:          0.;
623$            UNSTORE:        ;
618$            DELAY:          0.000:NEXT(ENDS_ASS L2);

ENDS_ASS L2     QUEUE,          ENDS_ASS Q2:DETACH;
ENDS_ASS L1     QUEUE,          ENDS_ASS Q1:DETACH;
                MATCH,          :ENDS_ASS L1,643$:
                                ENDS_ASS L2,642$;
643$            TRACE,          -1,"-Sending match to 12$\n"::NEXT(12$);
```

```
642$          TRACE,        -1,"-Sending match to 11$\n"::NEXT(11$);

12$           STATION,      FITTING_ENDS;
669$          TRACE,        -1,"-Arrived to station FITTING_ENDS\n";
706$          DELAY:        0.;
662$          TRACE,        -1,"-Waiting for resource FITTING_ENDS_R\n";
747$          QUEUE,        FITTING_ENDS_R_Q;
746$          SEIZE,        1:FITTING_ENDS_R,1;
              TRACE,        -1,"-Delay for processing time TRIA(1,1.3,1.5)\n";
745$          DELAY:        TRIA(1,1.3,1.5);
661$          TRACE,        -1,"-Releasing resource\n";
744$          RELEASE:      FITTING_ENDS_R,1;
678$          DELAY:        0.;
655$          TRACE,        -1,"-Transferred to next module\n";
738$          DELAY:        0.000:NEXT(10$);

10$           DELAY:        0.00;
762$          ASSIGN:       M=ENT_ENDS_ST;
778$          DELAY:        0.;
751$          TRACE,        -1,"-Transferred to station ELECT_TEST\n";
770$          CONVEY:       ,ELECT_TEST;
11$           TRACE,        -1,"-Disposing entity\n";
790$          DISPOSE;
815$          CREATE,       1:2;
855$          STATION,      ARR_ENDS;
806$          TRACE,        -1,"-Arrived to system at station ARR_ENDS\n";
852$          ASSIGN:       Picture=Default;
831$          DELAY:        0.;
800$          TRACE,        -1,"-Transferred to station ENT_END_ST2\n";
829$          ROUTE:        0,ENT_END_ST2;
15$           STATION,      ENT_ENDS_ST;
862$          TRACE,        -1,"-Arrived to station ENT_ENDS_ST\n";
882$          STORE:        ENT_ENDS_ST_S1;
883$          DELAY:        0.;
868$          UNSTORE:      ;
863$          DELAY:        0.000:NEXT(ENDS_ASS L1);

19$           STATION,      ENT_ROT_ASS2;
890$          TRACE,        -1,"-Arrived to station ENT_ROT_ASS2\n";
910$          STORE:        ENT_ROT_ASS2_S1;
911$          DELAY:        0.;
896$          UNSTORE:      ;
891$          DELAY:        0.000:NEXT(ROASS L2);

ROASS L2      QUEUE,        ROASS Q2:DETACH;
ROASS L1      QUEUE,        ROASS Q1:DETACH;
              MATCH,        :ROASS L1,916$:
                            ROASS L2,915$;
916$          TRACE,        -1,"-Sending match to 17$\n"::NEXT(17$);

915$          TRACE,        -1,"-Sending match to 0$\n"::NEXT(0$);

17$           STATION,      FIT_ROTORS;
942$          TRACE,        -1,"-Arrived to station FIT_ROTORS\n";
979$          DELAY:        0.;
935$          TRACE,        -1,"-Waiting for resource FIT_ROTORS_R\n";
1020$         QUEUE,        FIT_ROTORS_R_Q;
1019$         SEIZE,        1:FIT_ROTORS_R,1;
              TRACE,        -1,"-Delay for processing time TRIA(1.22,2.5,3.5)\n";
1018$         DELAY:        TRIA(1.22,2.5,3.5);
934$          TRACE,        -1,"-Releasing resource\n";
1017$         RELEASE:      FIT_ROTORS_R,1;
951$          DELAY:        0.;
928$          TRACE,        -1,"-Transferred to next module\n";
1011$         DELAY:        0.000:NEXT(16$);

16$           DELAY:        0.00;
1035$         ASSIGN:       M=ENT_ROT_ASS;
1051$         DELAY:        0.;
1024$         TRACE,        -1,"-Transferred to station ENT_ENDS_ST\n";
1043$         CONVEY:       ,ENT_ENDS_ST;
0$            TRACE,        -1,"-Disposing entity\n";
1063$         DISPOSE;
1088$         CREATE,       1:2;
1128$         STATION,      ARR_ROTORS;
1079$         TRACE,        -1,"-Arrived to system at station ARR_ROTORS\n";
1125$         ASSIGN:       Picture=Default;
1104$         DELAY:        0.;
1073$         TRACE,        -1,"-Transferred to station ENT_ROT_ASS2\n";
```

```
1102$           ROUTE:          0,ENT_ROT_ASS2;
20$             STATION,        ENT_ROT_ASS;
1135$           TRACE,          -1,"-Arrived to station ENT_ROT_ASS\n";
1155$           STORE:          ENT_ROT_ASS_S1;
1156$           DELAY:          0.;
1141$           UNSTORE:        ;
1136$           DELAY:          0.000:NEXT(ROASS L1);

25$             STATION,        ENT_TER_FT2;
1163$           TRACE,          -1,"-Arrived to station ENT_TER_FT2\n";
1183$           STORE:          ENT_TER_FT2_S1;
1184$           DELAY:          0.;
1169$           UNSTORE:        ;
1164$           DELAY:          0.000:NEXT(TBOX L2);

TBOX L2         QUEUE,          TBOX Q2:DETACH;
TBOX L1         QUEUE,          TBOX Q1:DETACH;
                MATCH,          :TBOX L1,1189$:
                                TBOX L2,1188$;
1189$           TRACE,          -1,"-Sending match to 23$\n"::NEXT(23$);

1188$           TRACE,          -1,"-Sending match to 22$\n"::NEXT(22$);

23$             STATION,        T_BOX_FITTING;
1215$           TRACE,          -1,"-Arrived to station T_BOX_FITTING\n";
1252$           DELAY:          0.;
1208$           TRACE,          -1,"-Waiting for resource T_BOX_FITTING_R\n";
1293$           QUEUE,          T_BOX_FITTING_R_Q;
1292$           SEIZE,          1:T_BOX_FITTING_R,1;
                TRACE,          -1,"-Delay for processing time TRIA(3,3.5,4)\n";
1291$           DELAY:          TRIA(3,3.5,4);
1207$           TRACE,          -1,"-Releasing resource\n";
1290$           RELEASE:        T_BOX_FITTING_R,1;
1224$           DELAY:          0.;
1201$           TRACE,          -1,"-Transferred to next module\n";
1284$           DELAY:          0.000:NEXT(21$);

21$             DELAY:          0.00;
1308$           ASSIGN:         M=ENT_TER_FT;
1324$           DELAY:          0.;
1297$           TRACE,          -1,"-Transferred to station ENT_ROT_ASS\n";
1316$           CONVEY:         ,ENT_ROT_ASS;
22$             TRACE,          -1,"-Disposing entity\n";
1336$           DISPOSE;
1361$           CREATE,         1:2;
1401$           STATION,        ARR_TERMINAL;
1352$           TRACE,          -1,"-Arrived to system at station ARR_TERMINAL\n";
1398$           ASSIGN:         Picture=Default;
1377$           DELAY:          0.;
1346$           TRACE,          -1,"-Transferred to station ENT_TER_FT2\n";
1375$           ROUTE:          0,ENT_TER_FT2;
26$             STATION,        ENT_TER_FT;
1408$           TRACE,          -1,"-Arrived to station ENT_TER_FT\n";
1428$           STORE:          ENT_TER_FT_S1;
1429$           DELAY:          0.;
1414$           UNSTORE:        ;
1409$           DELAY:          0.000:NEXT(TBOX L1);

27$             STATION,        MACHINING;
1459$           TRACE,          -1,"-Arrived to station MACHINING\n";
1579$           STORE:          MACHINING_S1;
1580$           DELAY:          0.;
1564$           UNSTORE:        ;
1452$           TRACE,          -1,"-Waiting for resource MACHINING_R\n";
1561$           QUEUE,          MACHINING_R_Q;
1558$           SEIZE,          1:MACHINING_R,1;
1486$           STORE:          MACHINING_S3;
                TRACE,          -1,"-Delay for processing time TRIA(2,2.4,3)\n";
1545$           DELAY:          TRIA(2,2.4,3);
1522$           UNSTORE:        ;
1515$           STORE:          MACHINING_S4;
1449$           TRACE,          -1,"-Releasing resource\n";
1538$           RELEASE:        MACHINING_R,1;
1502$           DELAY:          0.;
1499$           UNSTORE:        ;
1443$           TRACE,          -1,"-Transferred to station ENT_TER_FT\n";
1494$           CONVEY:         ,ENT_TER_FT;
33$             STATION,        ENT_ASS11;
1587$           TRACE,          -1,"-Arrived to station ENT_ASS11\n";
1607$           STORE:          ENT_ASS11_S1;
```

```
1608$           DELAY:          0.;
1593$           UNSTORE:        ;
1588$           DELAY:          0.000:NEXT(ASS1 L2);

ASS1 L2         QUEUE,          ASS1 Q2:DETACH;
ASS1 L1         QUEUE,          ASS1 Q1:DETACH;
                MATCH,          :ASS1 L1,1613$:
                                ASS1 L2,1612$;
1613$           TRACE,          -1,"-Sending match to 31$\n"::NEXT(31$);

1612$           TRACE,          -1,"-Sending match to 29$\n"::NEXT(29$);

31$             STATION,        FR_WIND_ASS;
1639$           TRACE,          -1,"-Arrived to station FR_WIND_ASS\n";
1676$           DELAY:          0.;
1632$           TRACE,          -1,"-Waiting for resource FR_WIND_ASS_R\n";
1717$           QUEUE,          FR_WIND_ASS_R_Q;
1716$           SEIZE,          1:FR_WIND_ASS_R,1;
                TRACE,          -1,"-Delay for processing time TRIA(2,2.5,3)\n";
1715$           DELAY:          TRIA(2,2.5,3);
1631$           TRACE,          -1,"-Releasing resource\n";
1714$           RELEASE:        FR_WIND_ASS_R,1;
1648$           DELAY:          0.;
1625$           TRACE,          -1,"-Transferred to next module\n";
1708$           DELAY:          0.000:NEXT(30$);

30$             DELAY:          0.00;
1732$           ASSIGN:         M=ENT_ASS1;
1748$           DELAY:          0.;
1721$           TRACE,          -1,"-Transferred to station MACHINING\n";
1740$           CONVEY:         ,MACHINING;
29$             TRACE,          -1,"-Disposing entity\n";
1760$           DISPOSE;
34$             STATION,        ENT_ASS1;
1767$           TRACE,          -1,"-Arrived to station ENT_ASS1\n";
1787$           STORE:          ENT_ASS1_S1;
1788$           DELAY:          0.;
1773$           UNSTORE:        ;
1768$           DELAY:          0.000:NEXT(ASS1 L1);

1813$           CREATE,         1:2;
1853$           STATION,        ARR_FRA;
1804$           TRACE,          -1,"-Arrived to system at station ARR_FRA\n";
1850$           ASSIGN:         Picture=Default;
1829$           DELAY:          0.;
1798$           TRACE,          -1,"-Transferred to station ENT_ASS11\n";
1827$           ROUTE:          0,ENT_ASS11;
1878$           CREATE,         1:1:MARK(FLOW TIME);
1918$           STATION,        ARR_WIND;
1869$           TRACE,          -1,"-Arrived to system at station ARR_WIND\n";
1915$           ASSIGN:         Picture=Default;
1859$           TRACE,          -1,"-Waiting for conveyor CONV1\n";
1896$           ACCESS:         CONV1,1;
1894$           DELAY:          0.;
1863$           TRACE,          -1,"-Transferred to station ENT_ASS1\n";
1887$           CONVEY:         ,ENT_ASS1;
```

E.22. Experiment Listing for Assembly Line (Cell-100)

```
PROJECT,        ;

ATTRIBUTES:     FLOW TIME;

STORAGES:       ENT_ASS1_S1:
                ENT_ASS11_S1:
                MACHINING_S4:
                MACHINING_S3:
                MACHINING_S1:
                ENT_TER_FT_S1:
                ENT_TER_FT2_S1:
                ENT_ROT_ASS_S1:
                ENT_ROT_ASS2_S1:
                ENT_ENDS_ST_S1:
                ENT_END_ST2_S1:
                PAINTING_S4:
                PAINTING_S3:
```

```
               PAINTING_S1:
               FAV_FITT_S4:
               FAV_FITT_S3:
               FAV_FITT_S1;

QUEUES:        ASS1 Q2,FIFO:
               ASS1 Q1:
               FR_WIND_ASS_R_Q,FIFO:
               MACHINING_R_Q,FIFO:
               TBOX Q2:
               TBOX Q1:
               T_BOX_FITTING_R_Q,FIFO:
               ROASS Q2:
               ROASS Q1:
               FIT_ROTORS_R_Q,FIFO:
               ENDS_ASS Q2:
               ENDS_ASS Q1:
               FITTING_ENDS_R_Q,FIFO:
               ELECT_TEST_R_Q,FIFO:
               PAINTING_R_Q,FIFO:
               FAV_FITT_R_Q,FIFO;

PICTURES:      Default;

RESOURCES:     FR_WIND_ASS_R,Capacity(1,):
               MACHINING_R,Capacity(1,):
               T_BOX_FITTING_R,Capacity(1,):
               FIT_ROTORS_R,Capacity(1,):
               FITTING_ENDS_R,Capacity(1,):
               ELECT_TEST_R,Capacity(1,):
               PAINTING_R,Capacity(1,):
               FAV_FITT_R,Capacity(1,);

STATIONS:      ARR_WIND:
               ENT_ASS1:
               ARR_FRA:
               ENT_ASS11:
               FR_WIND_ASS:
               MACHINING:
               ENT_TER_FT:
               ARR_TERMINAL:
               ENT_TER_FT2:
               T_BOX_FITTING:
               ENT_ENDS_ST:
               ENT_ROT_ASS:
               ARR_ROTORS:
               ENT_ROT_ASS2:
               FIT_ROTORS:
               ARR_ENDS:
               ENT_END_ST2:
               FITTING_ENDS:
               ELECT_TEST:
               REJ_MOTORS:
               PAINTING:
               FAV_FITT:
               BATCHING;

SEGMENTS:      CONV1_Seg,ARR_WIND,ENT_ASS1-10,
               MACHINING-10,
               ENT_TER_FT-10,
               ENT_ROT_ASS-10,
               ENT_ENDS_ST-8,
               ELECT_TEST-12,
               PAINTING-15,
               FAV_FITT-100,
               BATCHING-30;

CONVEYORS:     CONV1,CONV1_Seg,35,1,Active,1,Accumulating;

COUNTERS:      REJ_MOTORS_C:
               FINISH MOTORS_100,,Replicate,"C:\BH\SIM\ASSEMB\CASS100.001";

TALLIES:       BATCHING_Ta,"C:\BH\SIM\ASSEMB\TASS100.001";

DSTATS:        MR(FR_WIND_ASS_R),FR_WIND_ASS_R Available:
               NR(FR_WIND_ASS_R),FR_WIND_ASS_R Busy:
               NQ(FR_WIND_ASS_R_Q),# in FR_WIND_ASS_R_Q:
               MR(MACHINING_R),MACHINING_R Available:
               NR(MACHINING_R),MACHINING_R Busy:
               NQ(MACHINING_R_Q),# in MACHINING_R_Q:
```

```
                MR(T_BOX_FITTING_R),T_BOX_FITTING_R Available:
                NR(T_BOX_FITTING_R),T_BOX_FITTING_R Busy:
                NQ(T_BOX_FITTING_R_Q),# in T_BOX_FITTING_R_Q:
                MR(FIT_ROTORS_R),FIT_ROTORS_R Available:
                NR(FIT_ROTORS_R),FIT_ROTORS_R Busy:
                NQ(FIT_ROTORS_R_Q),# in FIT_ROTORS_R_Q:
                MR(FITTING_ENDS_R),FITTING_ENDS_R Available:
                NR(FITTING_ENDS_R),FITTING_ENDS_R Busy:
                NQ(FITTING_ENDS_R_Q),# in FITTING_ENDS_R_Q:
                MR(ELECT_TEST_R),ELECT_TEST_R Available:
                NR(ELECT_TEST_R),ELECT_TEST_R Busy:
                NQ(ELECT_TEST_R_Q),# in ELECT_TEST_R_Q:
                MR(PAINTING_R),PAINTING_R Available:
                NR(PAINTING_R),PAINTING_R Busy:
                NQ(PAINTING_R_Q),# in PAINTING_R_Q:
                MR(FAV_FITT_R),FAV_FITT_R Available:
                NR(FAV_FITT_R),FAV_FITT_R Busy:
                NQ(FAV_FITT_R_Q),# in FAV_FITT_R_Q:
                NEA(CONV1),,"C:\BH\SIM\ASSEMB\CONV100.001";

REPLICATE,      5,0.0,2865,No,Yes;
```

E.23. Model Listing for Assembly Line (Cell-112)

```
0$              STATION,        BATCHING;
26$             TRACE,          -1,"-Arrived to station BATCHING\n";
56$             DELAY:          0.;
20$             TRACE,          -1,"-Exiting conveyor\n";
46$             EXIT:           ;
34$             COUNT:          FINI_MOTORS112,1;
29$             TALLY:          BATCHING_Ta,Interval(FLOW TIME),1;
19$             TRACE,          -1,"-Disposing entity\n";
27$             DISPOSE;
1$              STATION,        FAN_ASS;
86$             TRACE,          -1,"-Arrived to station FAN_ASS\n";
206$            STORE:          FAN_ASS_S1;
207$            DELAY:          0.;
191$            UNSTORE:        ;
79$             TRACE,          -1,"-Waiting for resource FAN_ASS_R\n";
188$            QUEUE,          FAN_ASS_R_Q;
185$            SEIZE,          1:FAN_ASS_R,1;
113$            STORE:          FAN_ASS_S3;
                TRACE,          -1,"-Delay for processing time TRIA(1,1.2,2)\n";
172$            DELAY:          TRIA(1,1.2,2);
149$            UNSTORE:        ;
142$            STORE:          FAN_ASS_S4;
76$             TRACE,          -1,"-Releasing resource\n";
165$            RELEASE:        FAN_ASS_R,1;
129$            DELAY:          0.;
126$            UNSTORE:        ;
70$             TRACE,          -1,"-Transferred to station BATCHING\n";
121$            CONVEY:         ,BATCHING;
3$              STATION,        PAINTING;
237$            TRACE,          -1,"-Arrived to station PAINTING\n";
357$            STORE:          PAINTING_S1;
358$            DELAY:          0.;
342$            UNSTORE:        ;
230$            TRACE,          -1,"-Waiting for resource PAINTING_R\n";
339$            QUEUE,          PAINTING_R_Q;
336$            SEIZE,          1:PAINTING_R,1;
264$            STORE:          PAINTING_S3;
                TRACE,          -1,"-Delay for processing time TRIA(3,4,4.5)\n";
323$            DELAY:          TRIA(3,4,4.5);
300$            UNSTORE:        ;
293$            STORE:          PAINTING_S4;
227$            TRACE,          -1,"-Releasing resource\n";
316$            RELEASE:        PAINTING_R,1;
280$            DELAY:          0.;
277$            UNSTORE:        ;
221$            TRACE,          -1,"-Transferred to station FAN_ASS\n";
272$            CONVEY:         ,FAN_ASS;
5$              STATION,        REJ_MOTORS;
366$            TRACE,          -1,"-Arrived to station REJ_MOTORS\n";
396$            DELAY:          0.;
360$            TRACE,          -1,"-Exiting conveyor\n";
386$            EXIT:           ;
```

```
374$          COUNT:         REJ_MOTORS_C,1;
359$          TRACE,         -1,"-Disposing entity\n";
367$          DISPOSE;
7$            STATION,       ELECT_TEST;
437$          TRACE,         -1,"-Arrived to Inspect station ELECT_TEST\n";
496$          DELAY:         0.;
552$          QUEUE,         ELECT_TEST_R_Q;
551$          SEIZE,         1:ELECT_TEST_R,1;
              TRACE,         -1,"-Delay for processing time TRIA(2,2.3,2.8) with failure
probability .1\n";
550$          DELAY:         TRIA(2,2.3,2.8);
540$          BRANCH,        1:With,.1,429$,Yes:
                             Else,430$,Yes;
429$          TRACE,         -1,"-Entity failed inspection\n";
444$          DELAY:         0.0;
485$          RELEASE:       ELECT_TEST_R,1;
460$          DELAY:         0.;
425$          TRACE,         -1,"-Transferred to station REJ_MOTORS\n";
542$          ROUTE:         1,REJ_MOTORS;
430$          TRACE,         -1,"-Entity passed inspection\n";
445$          DELAY:         0.0;
549$          RELEASE:       ELECT_TEST_R,1;
461$          DELAY:         0.;
418$          TRACE,         -1,"-Transferrinf to next module\n";
543$          DELAY:         0.000:NEXT(6$);

6$            DELAY:         0.00;
567$          ASSIGN:        M=ELECT_TEST;
583$          DELAY:         0.;
556$          TRACE,         -1,"-Transferred to station PAINTING\n";
575$          CONVEY:        ,PAINTING;
9$            STATION,       ENDS_ASS;
624$          TRACE,         -1,"-Arrived to station ENDS_ASS\n";
744$          STORE:         ENDS_ASS_S1;
745$          DELAY:         0.;
729$          UNSTORE:       ;
617$          TRACE,         -1,"-Waiting for resource ENDS_ASS_R\n";
726$          QUEUE,         ENDS_ASS_R_Q;
723$          SEIZE,         1:ENDS_ASS_R,1;
651$          STORE:         ENDS_ASS_S3;
              TRACE,         -1,"-Delay for processing time TRIA(1,1.4,2)\n";
710$          DELAY:         TRIA(1,1.4,2);
687$          UNSTORE:       ;
680$          STORE:         ENDS_ASS_S4;
614$          TRACE,         -1,"-Releasing resource\n";
703$          RELEASE:       ENDS_ASS_R,1;
667$          DELAY:         0.;
664$          UNSTORE:       ;
608$          TRACE,         -1,"-Transferred to station ELECT_TEST\n";
659$          CONVEY:        ,ELECT_TEST;
11$           STATION,       ROTOR_ASS;
775$          TRACE,         -1,"-Arrived to station ROTOR_ASS\n";
895$          STORE:         ROTOR_ASS_S1;
896$          DELAY:         0.;
880$          UNSTORE:       ;
768$          TRACE,         -1,"-Waiting for resource ROTOR_ASS_R\n";
877$          QUEUE,         ROTOR_ASS_R_Q;
874$          SEIZE,         1:ROTOR_ASS_R,1;
802$          STORE:         ROTOR_ASS_S3;
              TRACE,         -1,"-Delay for processing time TRIA(1.22,1.6,2.2)\n";
861$          DELAY:         TRIA(1.22,1.6,2.2);
838$          UNSTORE:       ;
831$          STORE:         ROTOR_ASS_S4;
765$          TRACE,         -1,"-Releasing resource\n";
854$          RELEASE:       ROTOR_ASS_R,1;
818$          DELAY:         0.;
815$          UNSTORE:       ;
759$          TRACE,         -1,"-Transferred to station ENDS_ASS\n";
810$          CONVEY:        ,ENDS_ASS;
13$           STATION,       T_BOX_FITT;
926$          TRACE,         -1,"-Arrived to station T_BOX_FITT\n";
1046$         STORE:         T_BOX_FITT_S1;
1047$         DELAY:         0.;
1031$         UNSTORE:       ;
919$          TRACE,         -1,"-Waiting for resource T_BOX_FITT_R\n";
1028$         QUEUE,         T_BOX_FITT_R_Q;
1025$         SEIZE,         1:T_BOX_FITT_R,1;
953$          STORE:         T_BOX_FITT_S3;
              TRACE,         -1,"-Delay for processing time TRIA(3,3.5,4)\n";
1012$         DELAY:         TRIA(3,3.5,4);
```

```
989$        UNSTORE:      ;
982$        STORE:        T_BOX_FITT_S4;
916$        TRACE,        -1,"-Releasing resource\n";
1005$       RELEASE:      T_BOX_FITT_R,1;
969$        DELAY:        0.;
966$        UNSTORE:      ;
910$        TRACE,        -1,"-Transferred to station ROTOR_ASS\n";
961$        CONVEY:       ,ROTOR_ASS;
15$         STATION,      MACHINING;
1077$       TRACE,        -1,"-Arrived to station MACHINING\n";
1197$       STORE:        MACHINING_S1;
1198$       DELAY:        0.;
1182$       UNSTORE:      ;
1070$       TRACE,        -1,"-Waiting for resource MACHINING_R\n";
1179$       QUEUE,        MACHINING_R_Q;
1176$       SEIZE,        1:MACHINING_R,1;
1104$       STORE:        MACHINING_S3;
            TRACE,        -1,"-Delay for processing time TRIA(2,3,4)\n";
1163$       DELAY:        TRIA(2,3,4);
1140$       UNSTORE:      ;
1133$       STORE:        MACHINING_S4;
1067$       TRACE,        -1,"-Releasing resource\n";
1156$       RELEASE:      MACHINING_R,1;
1120$       DELAY:        0.;
1117$       UNSTORE:      ;
1061$       TRACE,        -1,"-Transferred to station T_BOX_FITT\n";
1112$       CONVEY:       ,T_BOX_FITT;
17$         STATION,      ASS_WIND_FRAM;
1228$       TRACE,        -1,"-Arrived to station ASS_WIND_FRAM\n";
1348$       STORE:        ASS_WIND_FRAM_S1;
1349$       DELAY:        0.;
1333$       UNSTORE:      ;
1221$       TRACE,        -1,"-Waiting for resource ASS_WIND_FRAM_R\n";
1330$       QUEUE,        ASS_WIND_FRAM_R_Q;
1327$       SEIZE,        1:ASS_WIND_FRAM_R,1;
1255$       STORE:        ASS_WIND_FRAM_S3;
            TRACE,        -1,"-Delay for processing time TRIA(2,2.3,3)\n";
1314$       DELAY:        TRIA(2,2.3,3);
1291$       UNSTORE:      ;
1284$       STORE:        ASS_WIND_FRAM_S4;
1218$       TRACE,        -1,"-Releasing resource\n";
1307$       RELEASE:      ASS_WIND_FRAM_R,1;
1271$       DELAY:        0.;
1268$       UNSTORE:      ;
1212$       TRACE,        -1,"-Transferred to station MACHINING\n";
1263$       CONVEY:       ,MACHINING;
1374$       CREATE,       1:1:MARK(FLOW TIME);
1414$       STATION,      ARR_WIND;
1365$       TRACE,        -1,"-Arrived to system at station ARR_WIND\n";
1411$       ASSIGN:       Picture=Default;
1355$       TRACE,        -1,"-Waiting for conveyor CONV1\n";
1392$       ACCESS:       CONV1,1;
1390$       DELAY:        0.;
1359$       TRACE,        -1,"-Transferred to station ASS_WIND_FRAM\n";
1383$       CONVEY:       ,ASS_WIND_FRAM;
```

E.24. Experiment Listing for Assembly Line (Cell-112)

```
PROJECT,      ;

ATTRIBUTES:   FLOW TIME;

STORAGES:     ASS_WIND_FRAM_S4:
              ASS_WIND_FRAM_S3:
              ASS_WIND_FRAM_S1:
              MACHINING_S4:
              MACHINING_S3:
              MACHINING_S1:
              T_BOX_FITT_S4:
              T_BOX_FITT_S3:
              T_BOX_FITT_S1:
              ROTOR_ASS_S4:
              ROTOR_ASS_S3:
              ROTOR_ASS_S1:
              ENDS_ASS_S4:
              ENDS_ASS_S3:
              ENDS_ASS_S1:
```

```
                PAINTING_S4:
                PAINTING_S3:
                PAINTING_S1:
                FAN_ASS_S4:
                FAN_ASS_S3:
                FAN_ASS_S1;

QUEUES:         ASS_WIND_FRAM_R_Q,FIFO:
                MACHINING_R_Q,FIFO:
                T_BOX_FITT_R_Q,FIFO:
                ROTOR_ASS_R_Q,FIFO:
                ENDS_ASS_R_Q,FIFO:
                ELECT_TEST_R_Q,FIFO:
                PAINTING_R_Q,FIFO:
                FAN_ASS_R_Q,FIFO;

PICTURES:       Default;

RESOURCES:      ASS_WIND_FRAM_R,Capacity(1,):
                MACHINING_R,Capacity(1,):
                T_BOX_FITT_R,Capacity(1,):
                ROTOR_ASS_R,Capacity(1,):
                ENDS_ASS_R,Capacity(1,):
                ELECT_TEST_R,Capacity(1,):
                PAINTING_R,Capacity(1,):
                FAN_ASS_R,Capacity(1,);

STATIONS:       ARR_WIND:
                ASS_WIND_FRAM:
                MACHINING:
                T_BOX_FITT:
                ROTOR_ASS:
                ENDS_ASS:
                ELECT_TEST:
                REJ_MOTORS:
                PAINTING:
                FAN_ASS:
                BATCHING;

SEGMENTS:       CONV1_Seg,ARR_WIND,ASS_WIND_FRAM-10,
                MACHINING-10,
                T_BOX_FITT-8,
                ROTOR_ASS-8,
                ENDS_ASS-8,
                ELECT_TEST-10,
                PAINTING-10,
                FAN_ASS-50,
                BATCHING-10;

CONVEYORS:      CONV1,CONV1_Seg,10,1,Active,1,Accumulating;

COUNTERS:       REJ_MOTORS_C:
                FINI_MOTORS112,,Replicate,"C:\BH\SIM\ASSEMB\CASS112.001";

TALLIES:        BATCHING_Ta,"C:\BH\SIM\ASSEMB\TASS112.001";

DSTATS:         MR(ASS_WIND_FRAM_R),ASS_WIND_FRAM_R Available:
                NR(ASS_WIND_FRAM_R),ASS_WIND_FRAM_R Busy:
                NQ(ASS_WIND_FRAM_R_Q),# in ASS_WIND_FRAM_R_Q:
                MR(MACHINING_R),MACHINING_R Available:
                NR(MACHINING_R),MACHINING_R Busy:
                NQ(MACHINING_R_Q),# in MACHINING_R_Q:
                MR(T_BOX_FITT_R),T_BOX_FITT_R Available:
                NR(T_BOX_FITT_R),T_BOX_FITT_R Busy:
                NQ(T_BOX_FITT_R_Q),# in T_BOX_FITT_R_Q:
                MR(ROTOR_ASS_R),ROTOR_ASS_R Available:
                NR(ROTOR_ASS_R),ROTOR_ASS_R Busy:
                NQ(ROTOR_ASS_R_Q),# in ROTOR_ASS_R_Q:
                MR(ENDS_ASS_R),ENDS_ASS_R Available:
                NR(ENDS_ASS_R),ENDS_ASS_R Busy:
                NQ(ENDS_ASS_R_Q),# in ENDS_ASS_R_Q:
                MR(ELECT_TEST_R),ELECT_TEST_R Available:
                NR(ELECT_TEST_R),ELECT_TEST_R Busy:
                NQ(ELECT_TEST_R_Q),# in ELECT_TEST_R_Q:
                MR(PAINTING_R),PAINTING_R Available:
                NR(PAINTING_R),PAINTING_R Busy:
                NQ(PAINTING_R_Q),# in PAINTING_R_Q:
                MR(FAN_ASS_R),FAN_ASS_R Available:
                NR(FAN_ASS_R),FAN_ASS_R Busy:
                NQ(FAN_ASS_R_Q),# in FAN_ASS_R_Q:
```

```
              NEA(CONV1),,"C:\BH\SIM\ASSEMB\CONV112.001";

REPLICATE,    5,0.0,2865,No,Yes;
```

E.25. Model Listing for Assembly Line (Cell-132)

```
0$            STATION,      BATCHING;
26$           TRACE,        -1,"-Arrived to station BATCHING\n";
56$           DELAY:        0.;
20$           TRACE,        -1,"-Exiting conveyor\n";
46$           EXIT:         ;
34$           COUNT:        FINI_MOTORS132,1;
29$           TALLY:        BATCHING_Ta,Interval(FLOW TIME),1;
19$           TRACE,        -1,"-Disposing entity\n";
27$           DISPOSE;
1$            STATION,      FAN_ASS;
86$           TRACE,        -1,"-Arrived to station FAN_ASS\n";
206$          STORE:        FAN_ASS_S1;
207$          DELAY:        0.;
191$          UNSTORE:      ;
79$           TRACE,        -1,"-Waiting for resource FAN_ASS_R\n";
188$          QUEUE,        FAN_ASS_R_Q;
185$          SEIZE,        1:FAN_ASS_R,1;
113$          STORE:        FAN_ASS_S3;
              TRACE,        -1,"-Delay for processing time TRIA(1,1.2,2)\n";
172$          DELAY:        TRIA(1,1.2,2);
149$          UNSTORE:      ;
142$          STORE:        FAN_ASS_S4;
76$           TRACE,        -1,"-Releasing resource\n";
165$          RELEASE:      FAN_ASS_R,1;
129$          DELAY:        0.;
126$          UNSTORE:      ;
70$           TRACE,        -1,"-Transferred to station BATCHING\n";
121$          CONVEY:       ,BATCHING;
3$            STATION,      PAINTING;
237$          TRACE,        -1,"-Arrived to station PAINTING\n";
357$          STORE:        PAINTING_S1;
358$          DELAY:        0.;
342$          UNSTORE:      ;
230$          TRACE,        -1,"-Waiting for resource PAINTING_R\n";
339$          QUEUE,        PAINTING_R_Q;
336$          SEIZE,        1:PAINTING_R,1;
264$          STORE:        PAINTING_S3;
              TRACE,        -1,"-Delay for processing time TRIA(2,2.5,4)\n";
323$          DELAY:        TRIA(2,2.5,4);
300$          UNSTORE:      ;
293$          STORE:        PAINTING_S4;
227$          TRACE,        -1,"-Releasing resource\n";
316$          RELEASE:      PAINTING_R,1;
280$          DELAY:        0.;
277$          UNSTORE:      ;
221$          TRACE,        -1,"-Transferred to station FAN_ASS\n";
272$          CONVEY:       ,FAN_ASS;
5$            STATION,      REJ_MOTORS;
366$          TRACE,        -1,"-Arrived to station REJ_MOTORS\n";
396$          DELAY:        0.;
360$          TRACE,        -1,"-Exiting conveyor\n";
386$          EXIT:         ;
374$          COUNT:        REJ_MOTORS_C,1;
359$          TRACE,        -1,"-Disposing entity\n";
367$          DISPOSE;
7$            STATION,      ELECT_TEST;
437$          TRACE,        -1,"-Arrived to Inspect station ELECT_TEST\n";
496$          DELAY:        0.;
552$          QUEUE,        ELECT_TEST_R_Q;
551$          SEIZE,        1:ELECT_TEST_R,1;
              TRACE,        -1,"-Delay for processing time TRIA(2,2.3,2.8) with failure
probability .1\n";
550$          DELAY:        TRIA(2,2.3,2.8);
540$          BRANCH,       1:With,.1,429$,Yes:
                            Else,430$,Yes;
429$          TRACE,        -1,"-Entity failed inspection\n";
444$          DELAY:        0.0;
485$          RELEASE:      ELECT_TEST_R,1;
460$          DELAY:        0.;
425$          TRACE,        -1,"-Transferred to station REJ_MOTORS\n";
```

```
542$          ROUTE:          1,REJ_MOTORS;
430$          TRACE,          -1,"-Entity passed inspection\n";
445$          DELAY:          0.0;
549$          RELEASE:        ELECT_TEST_R,1;
461$          DELAY:          0.;
418$          TRACE,          -1,"-Transferrinf to next module\n";
543$          DELAY:          0.000:NEXT(6$);

6$            DELAY:          0.00;
567$          ASSIGN:         M=ELECT_TEST;
583$          DELAY:          0.;
556$          TRACE,          -1,"-Transferred to station PAINTING\n";
575$          CONVEY:         ,PAINTING;
9$            STATION,        ENDS_ASS;
624$          TRACE,          -1,"-Arrived to station ENDS_ASS\n";
744$          STORE:          ENDS_ASS_S1;
745$          DELAY:          0.;
729$          UNSTORE:        ;
617$          TRACE,          -1,"-Waiting for resource ENDS_ASS_R\n";
726$          QUEUE,          ENDS_ASS_R_Q;
723$          SEIZE,          1:ENDS_ASS_R,1;
651$          STORE:          ENDS_ASS_S3;
              TRACE,          -1,"-Delay for processing time TRIA(1.5,2,2.3)\n";
710$          DELAY:          TRIA(1.5,2,2.3);
687$          UNSTORE:        ;
680$          STORE:          ENDS_ASS_S4;
614$          TRACE,          -1,"-Releasing resource\n";
703$          RELEASE:        ENDS_ASS_R,1;
667$          DELAY:          0.;
664$          UNSTORE:        ;
608$          TRACE,          -1,"-Transferred to station ELECT_TEST\n";
659$          CONVEY:         ,ELECT_TEST;
11$           STATION,        ROTOR_ASS;
775$          TRACE,          -1,"-Arrived to station ROTOR_ASS\n";
895$          STORE:          ROTOR_ASS_S1;
896$          DELAY:          0.;
880$          UNSTORE:        ;
768$          TRACE,          -1,"-Waiting for resource ROTOR_ASS_R\n";
877$          QUEUE,          ROTOR_ASS_R_Q;
874$          SEIZE,          1:ROTOR_ASS_R,1;
802$          STORE:          ROTOR_ASS_S3;
              TRACE,          -1,"-Delay for processing time TRIA(2,2.3,3)\n";
861$          DELAY:          TRIA(2,2.3,3);
838$          UNSTORE:        ;
831$          STORE:          ROTOR_ASS_S4;
765$          TRACE,          -1,"-Releasing resource\n";
854$          RELEASE:        ROTOR_ASS_R,1;
818$          DELAY:          0.;
815$          UNSTORE:        ;
759$          TRACE,          -1,"-Transferred to station ENDS_ASS\n";
810$          CONVEY:         ,ENDS_ASS;
13$           STATION,        T_BOX_FITT;
926$          TRACE,          -1,"-Arrived to station T_BOX_FITT\n";
1046$         STORE:          T_BOX_FITT_S1;
1047$         DELAY:          0.;
1031$         UNSTORE:        ;
919$          TRACE,          -1,"-Waiting for resource T_BOX_FITT_R\n";
1028$         QUEUE,          T_BOX_FITT_R_Q;
1025$         SEIZE,          1:T_BOX_FITT_R,1;
953$          STORE:          T_BOX_FITT_S3;
              TRACE,          -1,"-Delay for processing time TRIA(3,3.5,4)\n";
1012$         DELAY:          TRIA(3,3.5,4);
989$          UNSTORE:        ;
982$          STORE:          T_BOX_FITT_S4;
916$          TRACE,          -1,"-Releasing resource\n";
1005$         RELEASE:        T_BOX_FITT_R,1;
969$          DELAY:          0.;
966$          UNSTORE:        ;
910$          TRACE,          -1,"-Transferred to station ROTOR_ASS\n";
961$          CONVEY:         ,ROTOR_ASS;
15$           STATION,        MACHINING;
1077$         TRACE,          -1,"-Arrived to station MACHINING\n";
1197$         STORE:          MACHINING_S1;
1198$         DELAY:          0.;
1182$         UNSTORE:        ;
1070$         TRACE,          -1,"-Waiting for resource MACHINING_R\n";
1179$         QUEUE,          MACHINING_R_Q;
1176$         SEIZE,          1:MACHINING_R,1;
1104$         STORE:          MACHINING_S3;
              TRACE,          -1,"-Delay for processing time TRIA(3,4,5)\n";
```

```
1163$          DELAY:          TRIA(3,4,5);
1140$          UNSTORE:        ;
1133$          STORE:          MACHINING_S4;
1067$          TRACE,          -1,"-Releasing resource\n";
1156$          RELEASE:        MACHINING_R,1;
1120$          DELAY:          0.;
1117$          UNSTORE:        ;
1061$          TRACE,          -1,"-Transferred to station T_BOX_FITT\n";
1112$          CONVEY:         ,T_BOX_FITT;
17$            STATION,        ASS_WIND_FRAM;
1228$          TRACE,          -1,"-Arrived to station ASS_WIND_FRAM\n";
1348$          STORE:          ASS_WIND_FRAM_S1;
1349$          DELAY:          0.;
1333$          UNSTORE:        ;
1221$          TRACE,          -1,"-Waiting for resource ASS_WIND_FRAM_R\n";
1330$          QUEUE,          ASS_WIND_FRAM_R_Q;
1327$          SEIZE,          1:ASS_WIND_FRAM_R,1;
1255$          STORE:          ASS_WIND_FRAM_S3;
               TRACE,          -1,"-Delay for processing time TRIA(3,3.3,3.5)\n";
1314$          DELAY:          TRIA(3,3.3,3.5);
1291$          UNSTORE:        ;
1284$          STORE:          ASS_WIND_FRAM_S4;
1218$          TRACE,          -1,"-Releasing resource\n";
1307$          RELEASE:        ASS_WIND_FRAM_R,1;
1271$          DELAY:          0.;
1268$          UNSTORE:        ;
1212$          TRACE,          -1,"-Transferred to station MACHINING\n";
1263$          CONVEY:         ,MACHINING;
1374$          CREATE,         1:1:MARK(FLOW TIME);
1414$          STATION,        ARR_WIND;
1365$          TRACE,          -1,"-Arrived to system at station ARR_WIND\n";
1411$          ASSIGN:         Picture=Default;
1355$          TRACE,          -1,"-Waiting for conveyor CONV1\n";
1392$          ACCESS:         CONV1,1;
1390$          DELAY:          0.;
1359$          TRACE,          -1,"-Transferred to station ASS_WIND_FRAM\n";
1383$          CONVEY:         ,ASS_WIND_FRAM;
```

E.26. Experiment Listing for Assembly Line (Cell-132)

```
PROJECT,        ;

ATTRIBUTES:     FLOW TIME;

STORAGES:       ASS_WIND_FRAM_S4:
                ASS_WIND_FRAM_S3:
                ASS_WIND_FRAM_S1:
                MACHINING_S4:
                MACHINING_S3:
                MACHINING_S1:
                T_BOX_FITT_S4:
                T_BOX_FITT_S3:
                T_BOX_FITT_S1:
                ROTOR_ASS_S4:
                ROTOR_ASS_S3:
                ROTOR_ASS_S1:
                ENDS_ASS_S4:
                ENDS_ASS_S3:
                ENDS_ASS_S1:
                PAINTING_S4:
                PAINTING_S3:
                PAINTING_S1:
                FAN_ASS_S4:
                FAN_ASS_S3:
                FAN_ASS_S1;

QUEUES:         ASS_WIND_FRAM_R_Q,FIFO:
                MACHINING_R_Q,FIFO:
                T_BOX_FITT_R_Q,FIFO:
                ROTOR_ASS_R_Q,FIFO:
                ENDS_ASS_R_Q,FIFO:
                ELECT_TEST_R_Q,FIFO:
                PAINTING_R_Q,FIFO:
                FAN_ASS_R_Q,FIFO;

PICTURES:       Default;
```

```
RESOURCES:    ASS_WIND_FRAM_R,Capacity(1,):
              MACHINING_R,Capacity(1,):
              T_BOX_FITT_R,Capacity(1,):
              ROTOR_ASS_R,Capacity(1,):
              ENDS_ASS_R,Capacity(1,):
              ELECT_TEST_R,Capacity(1,):
              PAINTING_R,Capacity(1,):
              FAN_ASS_R,Capacity(1,);

STATIONS:     ARR_WIND:
              ASS_WIND_FRAM:
              MACHINING:
              T_BOX_FITT:
              ROTOR_ASS:
              ENDS_ASS:
              ELECT_TEST:
              REJ_MOTORS:
              PAINTING:
              FAN_ASS:
              BATCHING;

SEGMENTS:     CONV1_Seg,ARR_WIND,ASS_WIND_FRAM-10,
              MACHINING-10,
              T_BOX_FITT-8,
              ROTOR_ASS-8,
              ENDS_ASS-8,
              ELECT_TEST-10,
              PAINTING-10,
              FAN_ASS-50,
              BATCHING-10;

CONVEYORS:    CONV1,CONV1_Seg,10,1,Active,1,Accumulating;

COUNTERS:     REJ_MOTORS_C:
              FINI_MOTORS132,,Replicate,"C:\BH\SIM\ASSEMB\CASS132.001";

TALLIES:      BATCHING_Ta,"C:\BH\SIM\ASSEMB\TASS132.001";

DSTATS:       MR(ASS_WIND_FRAM_R),ASS_WIND_FRAM_R Available:
              NR(ASS_WIND_FRAM_R),ASS_WIND_FRAM_R Busy:
              NQ(ASS_WIND_FRAM_R_Q),# in ASS_WIND_FRAM_R_Q:
              MR(MACHINING_R),MACHINING_R Available:
              NR(MACHINING_R),MACHINING_R Busy:
              NQ(MACHINING_R_Q),# in MACHINING_R_Q:
              MR(T_BOX_FITT_R),T_BOX_FITT_R Available:
              NR(T_BOX_FITT_R),T_BOX_FITT_R Busy:
              NQ(T_BOX_FITT_R_Q),# in T_BOX_FITT_R_Q:
              MR(ROTOR_ASS_R),ROTOR_ASS_R Available:
              NR(ROTOR_ASS_R),ROTOR_ASS_R Busy:
              NQ(ROTOR_ASS_R_Q),# in ROTOR_ASS_R_Q:
              MR(ENDS_ASS_R),ENDS_ASS_R Available:
              NR(ENDS_ASS_R),ENDS_ASS_R Busy:
              NQ(ENDS_ASS_R_Q),# in ENDS_ASS_R_Q:
              MR(ELECT_TEST_R),ELECT_TEST_R Available:
              NR(ELECT_TEST_R),ELECT_TEST_R Busy:
              NQ(ELECT_TEST_R_Q),# in ELECT_TEST_R_Q:
              MR(PAINTING_R),PAINTING_R Available:
              NR(PAINTING_R),PAINTING_R Busy:
              NQ(PAINTING_R_Q),# in PAINTING_R_Q:
              MR(FAN_ASS_R),FAN_ASS_R Available:
              NR(FAN_ASS_R),FAN_ASS_R Busy:
              NQ(FAN_ASS_R_Q),# in FAN_ASS_R_Q:
              NEA(CONV1),,"C:\BH\SIM\ASSEMB\CONV132.001";

REPLICATE,    5,0.0,2865,No,Yes;
```

E.27. Model Listing for Assembly Line (Cell-160/180)

```
0$          STATION,     BATCHING;
30$         TRACE,       -1,"-Arrived to station BATCHING\n";
60$         DELAY:       0.;
24$         TRACE,       -1,"-Exiting conveyor\n";
50$         EXIT:        ;
37$         COUNT:       FINISHED MOTORS(MOTOR_T),1;
32$         TALLY:       TIME IN SYS(MOTOR_T),Interval(ARR TIME1),1;
```

```
23$             TRACE,          -1,"-Disposing entity\n";
31$             DISPOSE;
1$              STATION,        FAN_ASS;
90$             TRACE,          -1,"-Arrived to station FAN_ASS\n";
210$            STORE:          FAN_ASS_S1;
211$            DELAY:          0.;
195$            UNSTORE:        ;
83$             TRACE,          -1,"-Waiting for resource FAN_ASS_R\n";
192$            QUEUE,          FAN_ASS_R_Q;
189$            SEIZE,          1:FAN_ASS_R,1;
117$            STORE:          FAN_ASS_S3;
                TRACE,          -1,"-Delay for processing time TRIA(2,3,3.2)\n";
176$            DELAY:          TRIA(2,3,3.2);
153$            UNSTORE:        ;
146$            STORE:          FAN_ASS_S4;
80$             TRACE,          -1,"-Releasing resource\n";
169$            RELEASE:        FAN_ASS_R,1;
133$            DELAY:          0.;
130$            UNSTORE:        ;
74$             TRACE,          -1,"-Transferred to station BATCHING\n";
125$            CONVEY:         ,BATCHING;
3$              STATION,        REJ_MOTORS;
219$            TRACE,          -1,"-Arrived to station REJ_MOTORS\n";
249$            DELAY:          0.;
213$            TRACE,          -1,"-Exiting conveyor\n";
239$            EXIT:           ;
227$            COUNT:          REJ_MOTORS_C,1;
212$            TRACE,          -1,"-Disposing entity\n";
220$            DISPOSE;
4$              STATION,        PAINTING;
279$            TRACE,          -1,"-Arrived to station PAINTING\n";
399$            STORE:          PAINTING_S1;
400$            DELAY:          0.;
384$            UNSTORE:        ;
272$            TRACE,          -1,"-Waiting for resource PAINTING_R\n";
381$            QUEUE,          PAINTING_R_Q;
378$            SEIZE,          1:PAINTING_R,1;
306$            STORE:          PAINTING_S3;
                TRACE,          -1,"-Delay for processing time TRIA(3,4,5)\n";
365$            DELAY:          TRIA(3,4,5);
342$            UNSTORE:        ;
335$            STORE:          PAINTING_S4;
269$            TRACE,          -1,"-Releasing resource\n";
358$            RELEASE:        PAINTING_R,1;
322$            DELAY:          0.;
319$            UNSTORE:        ;
263$            TRACE,          -1,"-Transferred to station FAN_ASS\n";
314$            CONVEY:         ,FAN_ASS;
7$              STATION,        ELECT_TEST;
441$            TRACE,          -1,"-Arrived to Inspect station ELECT_TEST\n";
500$            DELAY:          0.;
556$            QUEUE,          ELECT_TEST_R_Q;
555$            SEIZE,          1:ELECT_TEST_R,1;
                TRACE,          -1,"-Delay for processing time TRIA(2,3,4) with failure
probability .1\n";
554$            DELAY:          TRIA(2,3,4);
544$            BRANCH,         1:With,.1,433$,Yes:
                                Else,434$,Yes;
433$            TRACE,          -1,"-Entity failed inspection\n";
448$            DELAY:          0.0;
489$            RELEASE:        ELECT_TEST_R,1;
464$            DELAY:          0.;
429$            TRACE,          -1,"-Transferred to station REJ_MOTORS\n";
546$            ROUTE:          3,REJ_MOTORS;
434$            TRACE,          -1,"-Entity passed inspection\n";
449$            DELAY:          0.0;
553$            RELEASE:        ELECT_TEST_R,1;
465$            DELAY:          0.;
422$            TRACE,          -1,"-Transferrinf to next module\n";
547$            DELAY:          0.000:NEXT(6$);

6$              DELAY:          0.00;
571$            ASSIGN:         M=ELECT_TEST;
587$            DELAY:          0.;
560$            TRACE,          -1,"-Transferred to station PAINTING\n";
579$            CONVEY:         ,PAINTING;
9$              STATION,        ENDS_FITTING;
642$            TRACE,          -1,"-Arrived to station ENDS_FITTING\n";
763$            DELAY:          0.;
636$            TRACE,          -1,"-Exiting conveyor\n";
```

```
751$          EXIT:          ;
635$          TRACE,         -1,"-Waiting for resource ENDS_FITTING_S\n";
744$          QUEUE,         ENDS_FITTING_S_Q;
740$          SEIZE,         1:SELECT(ENDS_FITTING_S,POR,),1;
669$          STORE:         ENDS_FITTING_S3;
              TRACE,         -1,"-Delay for processing time TRIA(3,4,5)\n";
728$          DELAY:         TRIA(3,4,5);
705$          UNSTORE:       ;
698$          STORE:         ENDS_FITTING_S4;
632$          TRACE,         -1,"-Releasing resource\n";
720$          RELEASE:       SELECT(ENDS_FITTING_S,,),1;
622$          TRACE,         -1,"-Waiting for conveyor CONV4\n";
687$          ACCESS:        CONV4,1;
685$          DELAY:         0.;
682$          UNSTORE:       ;
626$          TRACE,         -1,"-Transferred to station ELECT_TEST\n";
677$          CONVEY:        ,ELECT_TEST;
11$           STATION,       ROTOR_ASS;
793$          TRACE,         -1,"-Arrived to station ROTOR_ASS\n";
913$          STORE:         ROTOR_ASS_S1;
914$          DELAY:         0.;
898$          UNSTORE:       ;
786$          TRACE,         -1,"-Waiting for resource ROTOR_ASS_R\n";
895$          QUEUE,         ROTOR_ASS_R_Q;
892$          SEIZE,         1:ROTOR_ASS_R,1;
820$          STORE:         ROTOR_ASS_S3;
              TRACE,         -1,"-Delay for processing time TRIA(5,6,7)\n";
879$          DELAY:         TRIA(5,6,7);
856$          UNSTORE:       ;
849$          STORE:         ROTOR_ASS_S4;
783$          TRACE,         -1,"-Releasing resource\n";
872$          RELEASE:       ROTOR_ASS_R,1;
836$          DELAY:         0.;
833$          UNSTORE:       ;
777$          TRACE,         -1,"-Transferred to station ENDS_FITTING\n";
828$          CONVEY:        ,ENDS_FITTING;
13$           STATION,       T_BOX_FIT;
958$          TRACE,         -1,"-Arrived to station T_BOX_FIT\n";
1079$         DELAY:         0.;
952$          TRACE,         -1,"-Exiting conveyor\n";
1067$         EXIT:          ;
951$          TRACE,         -1,"-Waiting for resource T_BOX_FIT_S\n";
1060$         QUEUE,         T_BOX_FIT_S_Q;
1056$         SEIZE,         1:SELECT(T_BOX_FIT_S,POR,),1;
985$          STORE:         T_BOX_FIT_S3;
              TRACE,         -1,"-Delay for processing time TRIA(2.5,3,4)\n";
1044$         DELAY:         TRIA(2.5,3,4);
1021$         UNSTORE:       ;
1014$         STORE:         T_BOX_FIT_S4;
948$          TRACE,         -1,"-Releasing resource\n";
1036$         RELEASE:       SELECT(T_BOX_FIT_S,,),1;
938$          TRACE,         -1,"-Waiting for conveyor CONV3\n";
1003$         ACCESS:        CONV3,1;
1001$         DELAY:         0.;
998$          UNSTORE:       ;
942$          TRACE,         -1,"-Transferred to station ROTOR_ASS\n";
993$          CONVEY:        ,ROTOR_ASS;
15$           STATION,       MACHINING_CI;
1109$         TRACE,         -1,"-Arrived to station MACHINING_CI\n";
1229$         STORE:         MACHINING_CI_S1;
1230$         DELAY:         0.;
1214$         UNSTORE:       ;
1102$         TRACE,         -1,"-Waiting for resource MACHINING_CI_R\n";
1211$         QUEUE,         MACHINING_CI_R_Q;
1208$         SEIZE,         1:MACHINING_CI_R,1;
1136$         STORE:         MACHINING_CI_S3;
              TRACE,         -1,"-Delay for processing time TRIA(1,2,2.4)\n";
1195$         DELAY:         TRIA(1,2,2.4);
1172$         UNSTORE:       ;
1165$         STORE:         MACHINING_CI_S4;
1099$         TRACE,         -1,"-Releasing resource\n";
1188$         RELEASE:       MACHINING_CI_R,1;
1152$         DELAY:         0.;
1149$         UNSTORE:       ;
1093$         TRACE,         -1,"-Transferred to station T_BOX_FIT\n";
1144$         CONVEY:        ,T_BOX_FIT;
17$           STATION,       ASS_WIND_CI;
1260$         TRACE,         -1,"-Arrived to station ASS_WIND_CI\n";
1380$         STORE:         ASS_WIND_CI_S1;
1381$         DELAY:         0.;
```

```
1365$     UNSTORE:     ;
1253$     TRACE,       -1,"-Waiting for resource ASS_WIND_CI_R\n";
1362$     QUEUE,       ASS_WIND_CI_R_Q,20,1292$;
1359$     SEIZE,       1:ASS_WIND_CI_R,1;
1287$     STORE:       ASS_WIND_CI_S3;
          TRACE,       -1,"-Delay for processing time TRIA(1.5,3,5)\n";
1346$     DELAY:       TRIA(1.5,3,5);
1323$     UNSTORE:     ;
1316$     STORE:       ASS_WIND_CI_S4;
1250$     TRACE,       -1,"-Releasing resource\n";
1339$     RELEASE:     ASS_WIND_CI_R,1;
1240$     TRACE,       -1,"-Waiting for conveyor CONV2\n";
1305$     ACCESS:      CONV2,1;
1303$     DELAY:       0.;
1300$     UNSTORE:     ;
1244$     TRACE,       -1,"-Transferred to station MACHINING_CI\n";
1295$     CONVEY:      ,MACHINING_CI;
1292$     DISPOSE;
1406$     CREATE,      1:1:MARK(ARR TIME1);
1405$     ASSIGN:      MOTOR_T=2;
1446$     STATION,     AAR_CI;
1397$     TRACE,       -1,"-Arrived to system at station AAR_CI\n";
1443$     ASSIGN:      Picture=Default;
1422$     DELAY:       0.;
1391$     TRACE,       -1,"-Transferred to station ASS_WIND_CI\n";
1420$     ROUTE:       .2,ASS_WIND_CI;
19$       STATION,     MACHINING;
1476$     TRACE,       -1,"-Arrived to station MACHINING\n";
1596$     STORE:       MACHINING_S1;
1597$     DELAY:       0.;
1581$     UNSTORE:     ;
1469$     TRACE,       -1,"-Waiting for resource MACHINING_R\n";
1578$     QUEUE,       MACHINING_R_Q;
1575$     SEIZE,       1:MACHINING_R,1;
1503$     STORE:       MACHINING_S3;
          TRACE,       -1,"-Delay for processing time TRIA(2,3,4)\n";
1562$     DELAY:       TRIA(2,3,4);
1539$     UNSTORE:     ;
1532$     STORE:       MACHINING_S4;
1466$     TRACE,       -1,"-Releasing resource\n";
1555$     RELEASE:     MACHINING_R,1;
1519$     DELAY:       0.;
1516$     UNSTORE:     ;
1460$     TRACE,       -1,"-Transferred to station T_BOX_FIT\n";
1511$     CONVEY:      ,T_BOX_FIT;
21$       STATION,     ASS_WIND_FRA;
1627$     TRACE,       -1,"-Arrived to station ASS_WIND_FRA\n";
1747$     STORE:       ASS_WIND_FRA_S1;
1748$     DELAY:       0.;
1732$     UNSTORE:     ;
1620$     TRACE,       -1,"-Waiting for resource ASS_WIND_FRA_R\n";
1729$     QUEUE,       ASS_WIND_FRA_R_Q,20,1659$;
1726$     SEIZE,       1:ASS_WIND_FRA_R,1;
1654$     STORE:       ASS_WIND_FRA_S3;
          TRACE,       -1,"-Delay for processing time TRIA(2,2.3,3)\n";
1713$     DELAY:       TRIA(2,2.3,3);
1690$     UNSTORE:     ;
1683$     STORE:       ASS_WIND_FRA_S4;
1617$     TRACE,       -1,"-Releasing resource\n";
1706$     RELEASE:     ASS_WIND_FRA_R,1;
1607$     TRACE,       -1,"-Waiting for conveyor CONV1\n";
1672$     ACCESS:      CONV1,1;
1670$     DELAY:       0.;
1667$     UNSTORE:     ;
1611$     TRACE,       -1,"-Transferred to station MACHINING\n";
1662$     CONVEY:      ,MACHINING;
1659$     DISPOSE;
1773$     CREATE,      1:1:MARK(ARR TIME1);
1772$     ASSIGN:      MOTOR_T=1;
1813$     STATION,     ARR_WIND;
1764$     TRACE,       -1,"-Arrived to system at station ARR_WIND\n";
1810$     ASSIGN:      Picture=Default;
1789$     DELAY:       0.;
1758$     TRACE,       -1,"-Transferred to station ASS_WIND_FRA\n";
1787$     ROUTE:       .2,ASS_WIND_FRA;
```

E.28. Experiment Listing for Assembly Line (Cell-160/180)

```
PROJECT,        ;

ATTRIBUTES:     MOTOR_T:
                ARR TIME1;

STORAGES:       ASS_WIND_FRA_S4:
                ASS_WIND_FRA_S3:
                ASS_WIND_FRA_S1:
                MACHINING_S4:
                MACHINING_S3:
                MACHINING_S1:
                ASS_WIND_CI_S4:
                ASS_WIND_CI_S3:
                ASS_WIND_CI_S1:
                MACHINING_CI_S4:
                MACHINING_CI_S3:
                MACHINING_CI_S1:
                T_BOX_FIT_S4:
                T_BOX_FIT_S3:
                ROTOR_ASS_S4:
                ROTOR_ASS_S3:
                ROTOR_ASS_S1:
                ENDS_FITTING_S4:
                ENDS_FITTING_S3:
                PAINTING_S4:
                PAINTING_S3:
                PAINTING_S1:
                FAN_ASS_S4:
                FAN_ASS_S3:
                FAN_ASS_S1;

QUEUES:         ASS_WIND_FRA_R_Q,FIFO:
                MACHINING_R_Q,FIFO:
                ASS_WIND_CI_R_Q,FIFO:
                MACHINING_CI_R_Q,FIFO:
                T_BOX_FIT_S_Q,FIFO:
                T_BOX_ST1_Q,FIFO:
                T_BOX_ST2_Q,FIFO:
                ROTOR_ASS_R_Q,FIFO:
                ENDS_FITTING_S_Q,FIFO:
                ENDS_FIT_ST1_Q,FIFO:
                ENDS_FIT_ST2_Q,FIFO:
                ELECT_TEST_R_Q,FIFO:
                PAINTING_R_Q,FIFO:
                FAN_ASS_R_Q,FIFO;

PICTURES:       Default;

RESOURCES:      ASS_WIND_FRA_R,Capacity(1,):
                MACHINING_R,Capacity(1,):
                ASS_WIND_CI_R,Capacity(1,):
                MACHINING_CI_R,Capacity(1,):
                T_BOX_ST1,Capacity(1,):
                T_BOX_ST2,Capacity(1,):
                ROTOR_ASS_R,Capacity(1,):
                ENDS_FIT_ST1,Capacity(1,):
                ENDS_FIT_ST2,Capacity(1,):
                ELECT_TEST_R,Capacity(1,):
                PAINTING_R,Capacity(1,):
                FAN_ASS_R,Capacity(1,);

STATIONS:       ARR_WIND:
                ASS_WIND_FRA:
                MACHINING:
                T_BOX_FIT:
                AAR_CI:
                ASS_WIND_CI:
                MACHINING_CI:
                ROTOR_ASS:
                ENDS_FITTING:
                ELECT_TEST:
                REJ_MOTORS:
                PAINTING:
                FAN_ASS:
                BATCHING;
```

```
SEGMENTS:       CONV1_Seg,ASS_WIND_FRA,MACHINING-20,
                T_BOX_FIT-10:
                CONV2_Seg,ASS_WIND_CI,MACHINING_CI-5,
                T_BOX_FIT-15:
                CONV3_Seg,T_BOX_FIT,ROTOR_ASS-10,
                ENDS_FITTING-10:
                CONV4_Seg,ENDS_FITTING,ELECT_TEST-15,
                PAINTING-20,
                FAN_ASS-50,
                BATCHING-20;

CONVEYORS:      CONV1,CONV1_Seg,10,1,Active,1,Accumulating:
                CONV2,CONV2_Seg,1.,1,Active,1,Accumulating:
                CONV3,CONV3_Seg,1.,1,Active,1,Accumulating:
                CONV4,CONV4_Seg,1.,1,Active,1,Accumulating;

COUNTERS:       REJ_MOTORS_C:
                MOTORS_AL,,Replicate,"C:\BH\SIM\ASSEMB\CAL160.001":
                MOTORS_CI,,Replicate,"C:\BH\SIM\ASSEMB\CCI160.001";

TALLIES:        TIS_AL_M,"C:\BH\SIM\ASSEMB\TAAL160.001":
                TIS_CI_M,"C:\BH\SIM\ASSEMB\TACI160.001";

DSTATS:         MR(ASS_WIND_FRA_R),ASS_WIND_FRA_R Available:
                NR(ASS_WIND_FRA_R),ASS_WIND_FRA_R Busy:
                NQ(ASS_WIND_FRA_R_Q),# in ASS_WIND_FRA_R_Q:
                MR(MACHINING_R),MACHINING_R Available:
                NR(MACHINING_R),MACHINING_R Busy:
                NQ(MACHINING_R_Q),# in MACHINING_R_Q:
                MR(ASS_WIND_CI_R),ASS_WIND_CI_R Available:
                NR(ASS_WIND_CI_R),ASS_WIND_CI_R Busy:
                NQ(ASS_WIND_CI_R_Q),# in ASS_WIND_CI_R_Q:
                MR(MACHINING_CI_R),MACHINING_CI_R Available:
                NR(MACHINING_CI_R),MACHINING_CI_R Busy:
                NQ(MACHINING_CI_R_Q),# in MACHINING_CI_R_Q:
                NQ(T_BOX_FIT_S_Q),# in T_BOX_FIT_S_Q:
                MR(T_BOX_ST1),T_BOX_ST1 Available:
                NR(T_BOX_ST1),T_BOX_ST1 Busy:
                NQ(T_BOX_ST1_Q),# in T_BOX_ST1_Q:
                MR(T_BOX_ST2),T_BOX_ST2 Available:
                NR(T_BOX_ST2),T_BOX_ST2 Busy:
                NQ(T_BOX_ST2_Q),# in T_BOX_ST2_Q:
                MR(ROTOR_ASS_R),ROTOR_ASS_R Available:
                NR(ROTOR_ASS_R),ROTOR_ASS_R Busy:
                NQ(ROTOR_ASS_R_Q),# in ROTOR_ASS_R_Q:
                NQ(ENDS_FITTING_S_Q),# in ENDS_FITTING_S_Q:
                MR(ENDS_FIT_ST1),ENDS_FIT_ST1 Available:
                NR(ENDS_FIT_ST1),ENDS_FIT_ST1 Busy:
                NQ(ENDS_FIT_ST1_Q),# in ENDS_FIT_ST1_Q:
                MR(ENDS_FIT_ST2),ENDS_FIT_ST2 Available:
                NR(ENDS_FIT_ST2),ENDS_FIT_ST2 Busy:
                NQ(ENDS_FIT_ST2_Q),# in ENDS_FIT_ST2_Q:
                MR(ELECT_TEST_R),ELECT_TEST_R Available:
                NR(ELECT_TEST_R),ELECT_TEST_R Busy:
                NQ(ELECT_TEST_R_Q),# in ELECT_TEST_R_Q:
                MR(PAINTING_R),PAINTING_R Available:
                NR(PAINTING_R),PAINTING_R Busy:
                NQ(PAINTING_R_Q),# in PAINTING_R_Q:
                MR(FAN_ASS_R),FAN_ASS_R Available:
                NR(FAN_ASS_R),FAN_ASS_R Busy:
                NQ(FAN_ASS_R_Q),# in FAN_ASS_R_Q;

REPLICATE,      5,0.0,2865,No,Yes;

SETS:           T_BOX_FIT_S,T_BOX_ST1,T_BOX_ST2:
                ENDS_FITTING_S,ENDS_FIT_ST1,ENDS_FIT_ST2:
                FINISHED MOTORS,MOTORS_AL,MOTORS_CI:
                TIME IN SYS,TIS_AL_M,TIS_CI_M;
```

APPENDIX-F

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 1 of 1

Project: Run execution date : 2/13/1998
Analyst: Model revision date: 2/13/1998

Replication ended at time : 2265.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
T_R_100_AT_ST1	204.87	.51802	9.4687	392.99	588
T_R_112_AT_ST1	204.05	.50446	2.6129	400.09	533

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
STATION_1_R Available	1.0000	.00000	1.0000	1.0000	1.0000
STATION_1_R Busy	.99994	.00768	.00000	1.0000	1.0000
# in STATION_1_R_Q	115.28	.48977	.00000	212.00	182.00
BSIZE	34.345	.14902	16.000	55.000	31.000

COUNTERS

Identifier	Count	Limit
ROTORS_100 AT ST1	588	Infinite
ROTORS_112 AT ST1	533	Infinite

Figure F-1. Summary report for rotor casting station-1

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 1 of 1

Project: Run execution date : 2/13/1998
Analyst: Model revision date: 2/13/1998

Replication ended at time : 5040.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
T_R_132_AT_ST2	1054.0	.54051	41.423	1993.5	1495

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
BSIZE	29.486	.16857	11.000	48.000	38.000
STATION_2_R Available	1.0000	.00000	1.0000	1.0000	1.0000
STATION_2_R Busy	.99991	.00927	.00000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
ROTORS_132 AT ST2	1495	Infinite
R132 BAT_AT_ST2	67	Infinite

Figure F-2. Summary report for rotor casting station-2.

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 1 of 1

Project: Run execution date : 2/13/1998
Analyst: Model revision date: 2/13/1998

Replication ended at time : 5040.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
T_R_132_AT_ST3	1393.2	.58963	21.097	2768.8	599
T_R_160_AT_ST3	1376.8	.57088	5.8200	2780.2	541

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
BSIZE	19.360	.25260	1.0000	38.000	25.000
STATION_3_R Available	1.0000	.00000	1.0000	1.0000	1.0000
STATION_3_R Busy	.99997	.00515	.00000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
ROTORS_132 AT ST3	599	Infinite
ROTORS_160 AT ST3	541	Infinite

Figure F-3. Summary report for rotor casting station-3.

```
                         SIMAN V - License #9999999
                         Systems Modeling Corporation

                       Summary for Replication 1 of 1

Project:                                   Run execution date :    2/13/1998
Analyst:                                   Model revision date:    2/13/1998

Replication ended at time       : 5040.0

                              TALLY VARIABLES

Identifier                Average    Variation   Minimum    Maximum   Observations
_________________________________________________________________________________

T_R_160_AT_ST4             2342.9     .54251     222.68     4468.9        569

                         DISCRETE-CHANGE VARIABLES

Identifier                Average    Variation   Minimum    Maximum   Final Value
_________________________________________________________________________________

BSIZE                      29.300     .16421     11.000     49.000     34.000
STATION_4_R Available      1.0000     .00000     1.0000     1.0000     1.0000
STATION_4_R Busy           .99985     .01206     .00000     1.0000     1.0000

                                 COUNTERS

                    Identifier                 Count   Limit
                    __________________________________________

                    ROTORS_160 AT ST4            569  Infinite
                    R160 BAT_AT_ST4               26  Infinite
```

Figure F-4. Summary report for rotor casting station-4.

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 1 of 1

Project: Run execution date : 2/13/1998
Analyst: Model revision date: 2/13/1998

Replication ended at time : 2265.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
T_R_160_AT_ST5	1165.1	.49328	91.765	2009.7	52
T_R_180_AT_ST5	1010.1	.59832	20.579	2029.6	96
T_R_C225_AT_ST5	986.21	.57438	10.320	2020.3	78

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
BSIZE	19.339	.25782	3.0000	40.000	21.000
STATION_5_R Available	1.0000	.00000	1.0000	1.0000	1.0000
STATION_5_R Busy	.99981	.01384	.00000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
ROTORS_160 AT ST5	52	Infinite
ROTORS_180 AT ST5	96	Infinite
ROTORS_C225 AT ST5	78	Infinite

Figure F-5. Summary report for rotor casting staion-5

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 1 of 1

Project: Run execution date : 2/13/1998
Analyst: Model revision date: 2/13/1998

Replication ended at time : 2265.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
T_R_100_AT_ST6	817.01	.60067	16.846	1700.3	280
T_R_112_AT_ST6	908.53	.50898	30.251	1698.8	96
T_R_132_AT_ST6	862.38	.56079	3.8695	1715.3	567

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
BSIZE	79.426	.06237	62.000	100.00	83.000
STATION_6_R Available	1.0000	.00000	1.0000	1.0000	1.0000
STATION_6_R Busy	.99985	.01230	.00000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
ROTORS_100 AT ST6	280	Infinite
ROTORS_112 AT ST6	96	Infinite
ROTORS_132 AT ST6	567	Infinite

Figure F-6. Summary report for rotor casting station-6.

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 5 of 7

Project: Run execution date : 2/15/1998
Analyst: Model revision date: 2/15/1998

Replication ended at time : 6000.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
F_CELL1_Ta	632.24	.33945	180.17	1061.1	424
F_CELL2_Ta	631.01	.32084	199.74	1019.0	379
F_CELL3_Ta	951.71	.40662	304.98	1742.0	241

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
ASRST Active	1.0000	.00000	1.0000	1.0000	1.0000
ASRST Busy	.52520	.95081	.00000	1.0000	.00000
EMAG_R Available	1.0000	.00000	1.0000	1.0000	1.0000
EMAG_R Busy	.99330	.08211	.00000	1.0000	1.0000
# in EMAG_R_Q	.00000	--	.00000	.00000	.00000
EMAG OPERATOR_R Availa	1.0000	.00000	1.0000	1.0000	1.0000
EMAG_Q_SPACE Available	18.000	.00000	18.000	18.000	18.000
EMAG_Q_SPACE Busy	17.326	.10486	.00000	18.000	12.000
CELL_1_R Available	.74500	.58505	.00000	1.0000	1.0000
CELL_1_R Busy	.52974	.94218	.00000	1.0000	1.0000
CELL_2_R Available	.74500	.58505	.00000	1.0000	1.0000
CELL_2_R Busy	.47331	1.0548	.00000	1.0000	1.0000
# in CELL_2_R_Q	.24951	1.7343	.00000	1.0000	.00000
CELL_3_R Available	.74500	.58505	.00000	1.0000	1.0000
CELL_3_R Busy	.60303	.81135	.00000	1.0000	1.0000
# in CELL_3_R_Q	33.981	.68337	.00000	79.000	46.000
CELL_4_R Available	.74500	.58505	.00000	1.0000	1.0000
CELL_4_R Busy	.20270	1.9832	.00000	1.0000	1.0000
# in CELL_4_R_Q	7.0530	1.6854	.00000	40.000	31.000
# in TO_CELL1_Q	12.310	.87563	.00000	45.000	42.000
# in TO_CELL2_Q	11.061	.95433	.00000	51.000	11.000
# in CELL_1_R_Q	.25077	1.7285	.00000	1.0000	.00000

COUNTERS

Identifier	Count	Limit
STEEL BARS USED	155	Infinite
# BATCHES OF TYPE 1	1	Infinite
# BATCHES OF TYPE 2	3	Infinite
# BATCHES OF TYPE 3	4	Infinite
# BATCHES OF TYPE 4	4	Infinite
# BATCHES OF TYPE 5	3	Infinite
# BATCHES OF TYPE 6	1	Infinite
# BATCHES OF TYPE 7	4	Infinite
# BATCHES OF TYPE 8	3	Infinite
# BATCHES OF TYPE 9	3	Infinite
# BATCHES OF TYPE 10	2	Infinite
SAWED_SHAFTS_TYPE 1	9	Infinite
SAWED_SHAFTS_TYPE 2	188	Infinite
SAWED_SHAFTS_TYPE 3	93	Infinite
SAWED_SHAFTS_TYPE 4	241	Infinite
SAWED_SHAFTS_TYPE 5	201	Infinite
SAWED_SHAFTS_TYPE 6	7	Infinite
SAWED_SHAFTS_TYPE 7	123	Infinite
SAWED_SHAFTS_TYPE 8	171	Infinite
SAWED_SHAFTS_TYPE 9	177	Infinite
SAWED_SHAFTS_TYPE 10	54	Infinite
PRM_M_SHAFT TYPE 1	9	Infinite
PRM_M_SHAFT TYPE 2	188	Infinite
PRM_M_SHAFT TYPE 3	81	Infinite
PRM_M_SHAFT TYPE 4	241	Infinite
PRM_M_SHAFT TYPE 5	201	Infinite
PRM_M_SHAFT TYPE 6	7	Infinite
PRM_M_SHAFT TYPE 7	123	Infinite
PRM_M_SHAFT TYPE 8	171	Infinite
PRM_M_SHAFT TYPE 9	177	Infinite
PRM_M_SHAFT TYPE 10	54	Infinite
# SAW SETUPS	27	Infinite
# OF EMAG SETUPS	1253	Infinite
SHAFTS COMP_BY_EMAG	1252	Infinite
F_CELL1_C	424	Infinite
F_CELL2_C	379	Infinite
F_CELL3_C	241	Infinite

FREQUENCIES

Identifier	--Occurrences-- Category	Number	AvgTime	Standard Percent	Restricted Percent
STATE(SAWING_R)	BLOCKED	802	3.3448	44.71	44.71
	BUSY	819	1.6492	22.51	22.51
	IDLE	156	12.607	32.78	32.78

Figure F-7. Simulation report for shaft production (line-1)

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 2 of 3

Project: Run execution date : 2/15/1998
Analyst: Model revision date: 2/15/1998
Replication ended at time : 4458.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINISHED BOX_Ta	2227.0	.57650	8.5847	4453.6	870

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OKUMA_LT_15_M_R Availa	.96187	.19911	.00000	1.0000	1.0000
OKUMA_LT_15_M_R Busy	.97085	.17329	.00000	1.0000	1.0000
# in OKUMA_LT_15_M_R_Q	9.6223	.08794	.00000	10.000	10.000
GRINDING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
GRINDING_R Busy	.55123	.90230	.00000	1.0000	1.0000
# in GRINDING_R_Q	.00000	--	.00000	.00000	.00000
OPERATOR Active	1.0000	.00000	1.0000	1.0000	1.0000
OPERATOR Busy	.72693	.61290	.00000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
FINISHED BOX_C	870	Infinite

Figure F-8. Simulation report for shaft production line-2.

SIMAN V - License #9999999
Systems Modeling Corporation
Summary for Replication 3 of 3

Project: Run execution date : 2/15/1998
Analyst: Model revision date: 2/15/1998

Replication ended at time : 4485.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINISHED BOX_Ta	2242.2	.57829	10.816	4483.1	692
REJECTED_Ta	2454.5	.46275	96.670	3926.9	9

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OKUMA_LT_15_M_R Availa	.96210	.19849	.00000	1.0000	1.0000
OKUMA_LT_15_M_R Busy	.96868	.17982	.00000	1.0000	1.0000
# in OKUMA_LT_15_M_R_Q	9.6226	.08880	.00000	10.000	10.000
GRINDING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
GRINDING_R Busy	.62475	.77501	.00000	1.0000	1.0000
# in GRINDING_R_Q	.00000	--	.00000	.00000	.00000
OPERATOR Active	1.0000	.00000	1.0000	1.0000	1.0000
OPERATOR Busy	.99893	.03266	.00000	1.0000	1.0000
INSPECTION_R Available	1.0000	.00000	1.0000	1.0000	1.0000
INSPECTION_R Busy	.23347	1.8119	.00000	1.0000	.00000
# in INSPECTION_R_Q	.00000	--	.00000	.00000	.00000

COUNTERS

Identifier	Count	Limit
FINISHED BOX_C	692	Infinite
REJECTED_C	9	Infinite

Figure F-9. Simulation report for shaft production (cell-100).

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 2 of 3

Project: Run execution date : 2/15/1998
Analyst: Model revision date: 2/15/1998

Replication ended at time : 4485.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINISHED BOX_Ta	2248.0	.57506	10.993	4479.2	695
REJECTED_Ta	2131.2	.69384	224.35	4214.7	5

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OKUMA_LT_15_M_R Availa	.96210	.19849	.00000	1.0000	1.0000
OKUMA_LT_15_M_R Busy	.96336	.19503	.00000	1.0000	1.0000
# in OKUMA_LT_15_M_R_Q	7.5550	.32942	.00000	10.000	9.0000
GRINDING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
GRINDING_R Busy	.62480	.77493	.00000	1.0000	.00000
# in GRINDING_R_Q	.00000	--	.00000	.00000	.00000
OPERATOR Active	1.0000	.00000	1.0000	1.0000	1.0000
OPERATOR Busy	.99878	.03502	.00000	1.0000	1.0000
INSPECTION_R Available	1.0000	.00000	1.0000	1.0000	1.0000
INSPECTION_R Busy	.23342	1.8122	.00000	1.0000	1.0000
# in INSPECTION_R_Q	.00000	--	.00000	.00000	.00000

COUNTERS

Identifier	Count	Limit
FINISHED BOX_C	695	Infinite
REJECTED_C	5	Infinite

Figure F-10. Simulation results for shaft production (cell-112).

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 3 of 5

Project: Run execution date : 2/25/1998
Analyst: AL_AMARI Model revision date: 2/25/1998

Replication ended at time : 8550.0
Statistics were cleared at time: 5700.0
Statistics accumulated for time: 2850.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
LARGE ASS IN SYS	4421.9	.11146	3575.7	5269.5	1138
SMALL ASS IN SYS	3735.9	.11033	3023.9	4450.6	1428

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
SHAFT FITTING_1_R Avai	1.0000	.00000	1.0000	1.0000	1.0000
SHAFT FITTING_1_R Busy	.08777	3.2239	.00000	1.0000	.00000
# in SHAFT FITTING_1_R	.00000	--	.00000	.00000	.00000
SHAFT FITTING_2_R Avai	1.0000	.00000	1.0000	1.0000	1.0000
SHAFT FITTING_2_R Busy	.11003	2.8440	.00000	1.0000	.00000
# in SHAFT FITTING_2_R	.00000	--	.00000	.00000	.00000
COOLING STATION_1_R Av	1.0000	.00000	1.0000	1.0000	1.0000
COOLING STATION_1_R Bu	.10542	2.9131	.00000	1.0000	.00000
# in COOLING STATION_1	.00000	--	.00000	.00000	.00000
COOLING STATION_2_R Av	1.0000	.00000	1.0000	1.0000	1.0000
COOLING STATION_2_R Bu	.12172	2.6862	.00000	1.0000	.00000
# in COOLING STATION_2	.00000	--	.00000	.00000	.00000
# in TURNING STATION_1	4.9137	1.2530	.00000	18.000	.00000
TURNING_MACHINE_11 Ava	1.0000	.00000	1.0000	1.0000	1.0000
TURNING_MACHINE_11 Bus	.54658	.91081	.00000	1.0000	.00000
# in TURNING_MACHINE_1	.00000	--	.00000	.00000	.00000
TURNING_MACHINE_12 Ava	1.0000	.00000	1.0000	1.0000	1.0000
TURNING_MACHINE_12 Bus	.54797	.90825	.00000	1.0000	.00000
# in TURNING_MACHINE_1	.00000	--	.00000	.00000	.00000
# in TURNING STATION_2	4.8810	1.2646	.00000	18.000	14.000
TURNING_MACHINE_21 Ava	1.0000	.00000	1.0000	1.0000	1.0000
TURNING_MACHINE_21 Bus	.53883	.92513	.00000	1.0000	1.0000
# in TURNING_MACHINE_2	.00000	--	.00000	.00000	.00000
TURNING_MACHINE_22 Ava	1.0000	.00000	1.0000	1.0000	1.0000
TURNING_MACHINE_22 Bus	.53973	.92346	.00000	1.0000	1.0000
# in TURNING_MACHINE_2	.00000	--	.00000	.00000	.00000
# in SCALING STATION_1	3.2973	1.6839	.00000	18.000	.00000
SCALE_11 Available	1.0000	.00000	1.0000	1.0000	1.0000
SCALE_11 Busy	.36591	1.3164	.00000	1.0000	.00000
# in SCALE_11_Q	.00000	--	.00000	.00000	.00000
SCALE_12 Available	1.0000	.00000	1.0000	1.0000	1.0000
SCALE_12 Busy	.36639	1.3150	.00000	1.0000	.00000
# in SCALE_12_Q	.00000	--	.00000	.00000	.00000
SCALE_21 Available	1.0000	.00000	1.0000	1.0000	1.0000
SCALE_21 Busy	.75194	.57436	.00000	1.0000	1.0000
# in SCALE_21_Q	.00000	--	.00000	.00000	.00000
SCALE_22 Available	1.0000	.00000	1.0000	1.0000	1.0000
SCALE_22 Busy	.75168	.57476	.00000	1.0000	1.0000
# in SCALE_22_Q	.00000	--	.00000	.00000	.00000
# in SCALING STATION_2	6.7629	.93233	.00000	18.000	4.0000
# in WIP_Q1	9.6000	.60056	.00000	20.000	11.000
# in WIP_Q2	9.5035	.60579	.00000	20.000	13.000

COUNTERS

Identifier	Count	Limit
LARGE_ROTORS	1138	Infinite
SMALL_ROTORS	1428	Infinite

Figure F-11. Simulation report for rotor assembly line-1

SIMAN V - License #9999999
Systems Modeling Corporation
Summary for Replication 3 of 5

Project: Run execution date : 2/25/1998
Analyst: Model revision date: 2/25/1998

Replication ended at time : 2850.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINISHED ROTOR ASS_Ta	128.43	.08786	106.61	165.60	844

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR Active	1.0000	.00000	1.0000	1.0000	1.0000
OPERATOR Busy	.26887	1.6490	.00000	1.0000	1.0000
COOLING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
COOLING_R Busy	.18232	2.1177	.00000	1.0000	.00000
# in COOLING_R_Q	.00000	--	.00000	.00000	.00000
TURNING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
TURNING_R Busy	.23427	1.8079	.00000	1.0000	.00000
# in TURNING_R_Q	.08681	3.2433	.00000	1.0000	.00000
SCALE OPERATOR Active	1.0000	.00000	1.0000	1.0000	1.0000
SCALE OPERATOR Busy	.54229	.91872	.00000	1.0000	1.0000
# in SCALE OPERATOR_Q	5.1029	1.2503	.00000	19.000	15.000
NEC(CONV1)	6.3088	.53905	.00000	12.000	12.000

COUNTERS

Identifier	Count	Limit
FINISHED ROTOR ASS_C	844	Infinite

Figure F-12. Simulation report for cell-100 rotor assembly.

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 3 of 3

Project: Run execution date : 2/25/1998
Analyst: Model revision date: 2/25/1998

Replication ended at time : 2850.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINISHED ROTOR ASS_Ta	139.76	.07812	117.79	168.08	797

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR Active	1.0000	.00000	1.0000	1.0000	1.0000
OPERATOR Busy	.30792	1.4992	.00000	1.0000	.00000
COOLING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
COOLING_R Busy	.19834	2.0104	.00000	1.0000	.00000
# in COOLING_R_Q	.00000	--	.00000	.00000	.00000
TURNING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
TURNING_R Busy	.42854	1.1547	.00000	1.0000	1.0000
# in TURNING_R_Q	.19778	2.0565	.00000	2.0000	1.0000
SCALE OPERATOR Active	1.0000	.00000	1.0000	1.0000	1.0000
SCALE OPERATOR Busy	.51423	.97193	.00000	1.0000	1.0000
# in SCALE OPERATOR_Q	4.8111	1.3055	.00000	19.000	2.0000
NEC(CONV1)	6.0762	.55193	.00000	12.000	3.0000

COUNTERS

Identifier	Count	Limit
FINISHED ROTOR ASS_C	797	Infinite

Figure F-13. Simulation report for cell-112 rotor assembly.

Summary for Replication 2 of 3

Project: Run execution date : 2/27/1998
Analyst: Model revision date: 2/27/1998

Replication ended at time : 3420.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
F_WOUND PACK_Ta	1499.5	.49279	223.92	2807.8	536

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
# in LAM LABOUR_Q	.49963	1.0007	.00000	1.0000	1.0000
LAM LABOUR Active	1.0000	.00000	1.0000	1.0000	1.0000
LAM LABOUR Busy	1.0000	.00000	.00000	1.0000	1.0000
CLEATER_R Available	1.0000	.00000	1.0000	1.0000	1.0000
CLEATER_R Busy	.22832	1.8384	.00000	1.0000	1.0000
# in CLEATER_R_Q	.00000	--	.00000	.00000	.00000
LAM INSPECT_R Availabl	1.0000	.00000	1.0000	1.0000	1.0000
LAM INSPECT_R Busy	.05412	4.1806	.00000	1.0000	.00000
# in LAM INSPECT_R_Q	.00000	--	.00000	.00000	.00000
INSULATOR_R Available	1.0000	.00000	1.0000	1.0000	1.0000
INSULATOR_R Busy	.15890	2.3007	.00000	1.0000	.00000
# in INSULATOR_R_Q	.00000	--	.00000	.00000	.00000
WINDER1_R Available	1.0000	.00000	1.0000	1.0000	1.0000
WINDER1_R Busy	.68679	.67531	.00000	1.0000	1.0000
# in WINDER1_R_Q	.00000	--	.00000	.00000	.00000
WINDER2_R Available	1.0000	.00000	1.0000	1.0000	1.0000
WINDER2_R Busy	.76471	.55470	.00000	1.0000	1.0000
# in WINDER2_R_Q	.00000	--	.00000	.00000	.00000
WINDER3_R Available	1.0000	.00000	1.0000	1.0000	1.0000
WINDER3_R Busy	.65415	.72712	.00000	1.0000	.00000
# in WINDER3_R_Q	.00000	--	.00000	.00000	.00000
WINDER4_R Available	1.0000	.00000	1.0000	1.0000	1.0000
WINDER4_R Busy	.71979	.62394	.00000	1.0000	.00000
# in WINDER4_R_Q	.00000	--	.00000	.00000	.00000
# in FINISHING OFF OP_	.00000	--	.00000	.00000	.00000
F_OFF_OPERATOR1 Availa	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERATOR1 Busy	.92289	.28905	.00000	1.0000	1.0000
# in F_OFF_OPERATOR1_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERATOR2 Availa	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERATOR2 Busy	.91525	.30429	.00000	1.0000	1.0000
# in F_OFF_OPERATOR2_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERATOR3 Availa	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERATOR3 Busy	.90389	.32608	.00000	1.0000	1.0000
# in F_OFF_OPERATOR3_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERATOR5 Availa	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERATOR5 Busy	.00000	--	.00000	.00000	.00000
# in F_OFF_OPERATOR5_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERATOR6 Availa	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERATOR6 Busy	.00000	--	.00000	.00000	.00000
# in F_OFF_OPERATOR6_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERATOR7 Availa	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERATOR7 Busy	.00000	--	.00000	.00000	.00000
# in F_OFF_OPERATOR7_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERATOR8 Availa	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERATOR8 Busy	.00000	--	.00000	.00000	.00000
# in F_OFF_OPERATOR8_Q	.00000	--	.00000	.00000	.00000
# in CONN_OPER_Q	.00000	--	.00000	.00000	.00000
CON_OPERATOR1 Availabl	1.0000	.00000	1.0000	1.0000	1.0000
CON_OPERATOR1 Busy	.79377	.50971	.00000	1.0000	1.0000
# in CON_OPERATOR1_Q	.00000	--	.00000	.00000	.00000
CON_OPERATOR2 Availabl	1.0000	.00000	1.0000	1.0000	1.0000
CON_OPERATOR2 Busy	.74586	.58373	.00000	1.0000	1.0000
# in CON_OPERATOR2_Q	.00000	--	.00000	.00000	.00000
CON_OPERATOR3 Availabl	1.0000	.00000	1.0000	1.0000	1.0000
CON_OPERATOR3 Busy	.73751	.59659	.00000	1.0000	1.0000
# in CON_OPERATOR3_Q	.00000	--	.00000	.00000	.00000
CON_OPERATOR4 Availabl	1.0000	.00000	1.0000	1.0000	1.0000
CON_OPERATOR4 Busy	.68106	.68432	.00000	1.0000	1.0000
# in CON_OPERATOR4_Q	.00000	--	.00000	.00000	.00000
CON_OPERATOR5 Availabl	1.0000	.00000	1.0000	1.0000	1.0000
CON_OPERATOR5 Busy	.02955	5.7302	.00000	1.0000	.00000
# in CON_OPERATOR5_Q	.00000	--	.00000	.00000	.00000
CON_OPERATOR6 Availabl	1.0000	.00000	1.0000	1.0000	1.0000
CON_OPERATOR6 Busy	.00000	--	.00000	.00000	.00000
# in CON_OPERATOR6_Q	.00000	--	.00000	.00000	.00000
# in PRESSERS_M_Q	7.5179E-04	36.457	.00000	1.0000	.00000
PRESSER_1 Available	1.0000	.00000	1.0000	1.0000	1.0000
PRESSER_1 Busy	.13608	2.5196	.00000	1.0000	.00000
# in PRESSER_1_Q	.00000	--	.00000	.00000	.00000
PRESSER_2 Available	1.0000	.00000	1.0000	1.0000	1.0000
PRESSER_2 Busy	.12672	2.6250	.00000	1.0000	.00000
# in PRESSER_2_Q	.00000	--	.00000	.00000	.00000
TERMING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
TERMING_R Busy	.95204	.22445	.00000	1.0000	1.0000
# in TERMING_R_Q	1.9695	.89435	.00000	9.0000	7.0000
ELEC_TEST_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ELEC_TEST_R Busy	.25389	1.7142	.00000	1.0000	.00000
# in ELEC_TEST_R_Q	.00000	--	.00000	.00000	.00000
# in WYVERTIC MACHINES	.23208	2.7211	.00000	2.0000	.00000
WYVERTIC_1 Available	1.0000	.00000	1.0000	1.0000	1.0000
WYVERTIC_1 Busy	.22969	1.8313	.00000	1.0000	.00000
# in WYVERTIC_1_Q	.00000	--	.00000	.00000	.00000
WYVERTIC_2 Available	1.0000	.00000	1.0000	1.0000	1.0000
WYVERTIC_2 Busy	.23102	1.8244	.00000	1.0000	.00000
# in WYVERTIC_2_Q	.00000	--	.00000	.00000	.00000

COUNTERS

Identifier	Count	Limit
REJECTED_WOUND_C	21	Infinite
F_WOUND PACK_C	536	Infinite
REJECT LAM_C	89	Infinite

Figure F-14. Summary report for windings assembly cell-160/180

Systems Modeling Corporation
Summary for Replication 1 of 5

Project: Run execution date : 2/27/1998
Analyst: Model revision date: 2/27/1998

Replication ended at time : 2865.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINI_WOUND_P_Ta	1223.1	.48735	227.45	2216.3	520

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
LAM_LABOUR Active	1.0000	.00000	1.0000	1.0000	1.0000
LAM_LABOUR Busy	1.0000	.00000	.00000	1.0000	1.0000
CLEATER_R Available	1.0000	.00000	1.0000	1.0000	1.0000
CLEATER_R Busy	.13540	2.5269	.00000	1.0000	.00000
# in CLEATER_R_Q	.00000	--	.00000	.00000	.00000
WINDER_1_R Available	1.0000	.00000	1.0000	1.0000	1.0000
WINDER_1_R Busy	.69012	.67009	.00000	1.0000	.00000
# in WINDER_1_R_Q	.01255	8.8704	.00000	1.0000	.00000
WINDER_2_R Available	1.0000	.00000	1.0000	1.0000	1.0000
WINDER_2_R Busy	.84745	.42428	.00000	1.0000	1.0000
# in WINDER_2_R_Q	1.6469	1.0465	.00000	7.0000	.00000
# in FINISH OPERATORS_	.74855	1.3477	.00000	5.0000	.00000
F_OFF_OPERA._1 Availab	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERA._1 Busy	.94062	.25126	.00000	1.0000	1.0000
# in F_OFF_OPERA._1_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERA._2 Availab	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERA._2 Busy	.94591	.23914	.00000	1.0000	1.0000
# in F_OFF_OPERA._2_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERA._3 Availab	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERA._3 Busy	.93877	.25539	.00000	1.0000	1.0000
# in F_OFF_OPERA._3_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERA._4 Availab	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERA._4 Busy	.93576	.26201	.00000	1.0000	1.0000
# in F_OFF_OPERA._4_Q	.00000	--	.00000	.00000	.00000
F_OFF_OPERA._5 Availab	1.0000	.00000	1.0000	1.0000	1.0000
F_OFF_OPERA._5 Busy	.93414	.26552	.00000	1.0000	1.0000
# in F_OFF_OPERA._5_Q	.00000	--	.00000	.00000	.00000
PRESSING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
PRESSING_R Busy	.20286	1.9822	.00000	1.0000	.00000
# in PRESSING_R_Q	.01678	7.9493	.00000	2.0000	.00000
# in CONNECTORS_Q	.00000	--	.00000	.00000	.00000
CONNECTING_1 Available	1.0000	.00000	1.0000	1.0000	1.0000
CONNECTING_1 Busy	.61538	.79057	.00000	1.0000	1.0000
# in CONNECTING_1_Q	.00000	--	.00000	.00000	.00000
CONNECTING_2 Available	1.0000	.00000	1.0000	1.0000	1.0000
CONNECTING_2 Busy	.60838	.80232	.00000	1.0000	.00000
# in CONNECTING_2_Q	.00000	--	.00000	.00000	.00000
CONNECTING_4 Available	1.0000	.00000	1.0000	1.0000	1.0000
CONNECTING_4 Busy	.59969	.81702	.00000	1.0000	1.0000
# in CONNECTING_4_Q	.00000	--	.00000	.00000	.00000
CONNECTING_3 Available	1.0000	.00000	1.0000	1.0000	1.0000
CONNECTING_3 Busy	.60026	.81606	.00000	1.0000	.00000
# in CONNECTING_3_Q	.00000	--	.00000	.00000	.00000
CONNECTING_5 Available	1.0000	.00000	1.0000	1.0000	1.0000
CONNECTING_5 Busy	.59776	.82032	.00000	1.0000	1.0000
# in CONNECTING_5_Q	.00000	--	.00000	.00000	.00000
LACING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
LACING_R Busy	.14324	2.4456	.00000	1.0000	.00000
# in LACING_R_Q	.00000	--	.00000	.00000	.00000
ELCT_TEST_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ELCT_TEST_R Busy	.61128	.79744	.00000	1.0000	1.0000
# in ELCT_TEST_R_Q	.18896	2.3418	.00000	3.0000	.00000
DIP_WYVERT_R Available	1.0000	.00000	1.0000	1.0000	1.0000
DIP_WYVERT_R Busy	.95052	.22815	.00000	1.0000	1.0000
# in DIP_WYVERT_R_Q	1.9568	.75255	.00000	5.0000	5.0000

COUNTERS

Identifier	Count	Limit
REJECT_WIND_C	46	Infinite
FINI_WOUND_P_C	520	Infinite

Figure F-15. Summary report for windings assembly (cell-100).

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 1 of 5

Project: Run execution date : 2/27/1998
Analyst: Model revision date: 2/27/1998

Replication ended at time : 3100.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINI_WOUND_PACK_Ta	392.22	.16650	243.72	486.40	420

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
INSULATING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
INSULATING_R Busy	.16184	2.2757	.00000	1.0000	.00000
# in INSULATING_R_Q	1.8655	2.8079	.00000	40.000	.00000
WINDING_R Available	.90323	.32733	.00000	1.0000	1.0000
WINDING_R Busy	.90431	.32530	.00000	1.0000	1.0000
# in WINDING_R_Q	31.855	.26882	.00000	40.000	14.000
CONNECTING_R Available	5.0000	.00000	5.0000	5.0000	5.0000
CONNECTING_R Busy	2.2219	.44750	.00000	5.0000	2.0000
# in CONNECTING_R_Q	.00000	--	.00000	.00000	.00000
PRESSING_R Available	5.0000	.00000	5.0000	5.0000	5.0000
PRESSING_R Busy	.12799	2.6510	.00000	2.0000	.00000
# in PRESSING_R_Q	.00000	--	.00000	.00000	.00000
LACING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
LACING_R Busy	.10399	2.9354	.00000	1.0000	.00000
# in LACING_R_Q	.00117	29.170	.00000	1.0000	.00000
ELECT_TEST_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ELECT_TEST_R Busy	.52925	.94311	.00000	1.0000	1.0000
# in ELECT_TEST_R_Q	.07003	3.6737	.00000	2.0000	.00000
DIP_PAKE_WYVER_R Avail	1.0000	.00000	1.0000	1.0000	1.0000
DIP_PAKE_WYVER_R Busy	.67464	.69445	.00000	1.0000	.00000
# in DIP_PAKE_WYVER_R_	.00000	--	.00000	.00000	.00000
FINISHING OFF_R Availa	5.0000	.00000	5.0000	5.0000	5.0000
FINISHING OFF_R Busy	3.7500	.33965	.00000	5.0000	4.0000
# in FINISHING OFF_R_Q	.00619	12.673	.00000	1.0000	.00000

COUNTERS

Identifier	Count	Limit
REJ_WOUND_P_C	21	Infinite
FINI_WOUND_PACK_C	420	Infinite

Figure F-16. Summary report for windings assembly (cell-112)

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 1 of 5

Project: Run execution date : 2/28/1998
Analyst: Model revision date: 2/28/1998

Replication ended at time : 3100.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
FINISHED_W_PACK_Ta	244.49	.12533	181.05	303.81	520

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
SCALING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
SCALING_R Busy	.11111	2.8284	.00000	1.0000	.00000
# in SCALING_R_Q	.00872	11.344	.00000	2.0000	.00000
CLEATING_R Available	.96774	.18257	.00000	1.0000	.00000
CLEATING_R Busy	.22884	1.8357	.00000	1.0000	.00000
# in CLEATING_R_Q	.11873	4.7436	.00000	6.0000	.00000
INSULATORS_R Available	2.0000	.00000	2.0000	2.0000	2.0000
INSULATORS_R Busy	.07704	3.4612	.00000	1.0000	.00000
# in INSULATORS_R_Q	.00000	--	.00000	.00000	.00000
WINDINGS_R Available	3.0000	.00000	3.0000	3.0000	3.0000
WINDINGS_R Busy	2.9426	.11020	.00000	3.0000	3.0000
# in WINDINGS_R_Q	11.003	.65988	.00000	31.000	27.000
FINISHING OFF_R Availa	7.0000	.00000	7.0000	7.0000	7.0000
FINISHING OFF_R Busy	4.6363	.21274	.00000	6.0000	3.0000
# in FINISHING OFF_R_Q	.00000	--	.00000	.00000	.00000
CONNECTION_R Available	6.0000	.00000	6.0000	6.0000	6.0000
CONNECTION_R Busy	2.2017	.38196	.00000	5.0000	4.0000
# in CONNECTION_R_Q	.00000	--	.00000	.00000	.00000
PRESSING_R Available	2.0000	.00000	2.0000	2.0000	2.0000
PRESSING_R Busy	.12212	2.7876	.00000	2.0000	.00000
# in PRESSING_R_Q	1.8467E-04	73.580	.00000	1.0000	.00000
LACING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
LACING_R Busy	.22503	1.8557	.00000	1.0000	.00000
# in LACING_R_Q	.01649	7.8788	.00000	2.0000	.00000
ELECT_TEST_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ELECT_TEST_R Busy	.41531	1.1865	.00000	1.0000	1.0000
# in ELECT_TEST_R_Q	.04505	4.6040	.00000	1.0000	.00000
DIP_BAKE_WYVERTIC_R Av	1.0000	.00000	1.0000	1.0000	1.0000
DIP_BAKE_WYVERTIC_R Bu	.84335	.43099	.00000	1.0000	1.0000
# in DIP_BAKE_WYVERTIC	3.8114E-04	51.212	.00000	1.0000	.00000

COUNTERS

Identifier	Count	Limit
REJ_WOUND_P_C	25	Infinite
FINISHED_W_PACK_C	520	Infinite

Figure F-17. Summary report of windings assembly line (cell-132)

Summary for Replication 5 of 5

Project: Run execution date : 3/ 4/1998
Analyst: Model revision date: 3/ 4/1998
Replication ended at time : 14325.0
Statistics were cleared at time: 11460.0
Statistics accumulated for time: 2865.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
BATCHING_Ta	9274.8	.06408	8235.6	10284.	731

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
FR_WIND_ASS_R Availabl	1.0000	.00000	1.0000	1.0000	1.0000
FR_WIND_ASS_R Busy	.70219	.65124	.00000	1.0000	1.0000
# in FR_WIND_ASS_R_Q	.00000	--	.00000	.00000	.00000
MACHINING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
MACHINING_R Busy	.69705	.65926	.00000	1.0000	1.0000
# in MACHINING_R_Q	.00000	--	.00000	.00000	.00000
T_BOX_FITTING_R Availa	1.0000	.00000	1.0000	1.0000	1.0000
T_BOX_FITTING_R Busy	.98820	.10926	.00000	1.0000	1.0000
# in T_BOX_FITTING_R_Q	.00000	--	.00000	.00000	.00000
FIT_ROTORS_R Available	1.0000	.00000	1.0000	1.0000	1.0000
FIT_ROTORS_R Busy	.67308	.69693	.00000	1.0000	1.0000
# in FIT_ROTORS_R_Q	.00000	--	.00000	.00000	.00000
FITTING_ENDS_R Availab	1.0000	.00000	1.0000	1.0000	1.0000
FITTING_ENDS_R Busy	.35675	1.3428	.00000	1.0000	.00000
# in FITTING_ENDS_R_Q	.00000	--	.00000	.00000	.00000
ELECT_TEST_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ELECT_TEST_R Busy	.66004	.71767	.00000	1.0000	1.0000
# in ELECT_TEST_R_Q	.00000	--	.00000	.00000	.00000
PAINTING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
PAINTING_R Busy	.64228	.74630	.00000	1.0000	1.0000
# in PAINTING_R_Q	.00000	--	.00000	.00000	.00000
FAV_FITT_R Available	1.0000	.00000	1.0000	1.0000	1.0000
FAV_FITT_R Busy	.25625	1.7036	.00000	1.0000	1.0000
# in FAV_FITT_R_Q	.00000	--	.00000	.00000	.00000
NEA(CONV1)	33.460	.08788	.00000	36.000	35.000

COUNTERS

Identifier	Count	Limit
REJ_MOTORS_C	77	Infinite
FINISH MOTORS_100	731	Infinite

Figure F-18. Summary report for cell-100 final assembly.

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 4 of 5

Project: Run execution date : 3/ 4/1998
Analyst: Model revision date: 3/ 4/1998

Replication ended at time : 11460.0
Statistics were cleared at time: 8595.0
Statistics accumulated for time: 2865.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
BATCHING_Ta	7285.0	.08293	6252.4	8328.0	698

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
ASS_WIND_FRAM_R Availa	1.0000	.00000	1.0000	1.0000	1.0000
ASS_WIND_FRAM_R Busy	.66664	.70715	.00000	1.0000	1.0000
# in ASS_WIND_FRAM_R_Q	.00000	--	.00000	.00000	.00000
MACHINING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
MACHINING_R Busy	.82437	.46157	.00000	1.0000	1.0000
# in MACHINING_R_Q	.00000	--	.00000	.00000	.00000
T_BOX_FITT_R Available	1.0000	.00000	1.0000	1.0000	1.0000
T_BOX_FITT_R Busy	.95961	.20517	.00000	1.0000	1.0000
# in T_BOX_FITT_R_Q	.00000	--	.00000	.00000	.00000
ROTOR_ASS_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ROTOR_ASS_R Busy	.46226	1.0785	.00000	1.0000	.00000
# in ROTOR_ASS_R_Q	.00000	--	.00000	.00000	.00000
ENDS_ASS_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ENDS_ASS_R Busy	.40097	1.2222	.00000	1.0000	1.0000
# in ENDS_ASS_R_Q	.00000	--	.00000	.00000	.00000
ELECT_TEST_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ELECT_TEST_R Busy	.64867	.73595	.00000	1.0000	1.0000
# in ELECT_TEST_R_Q	.00000	--	.00000	.00000	.00000
PAINTING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
PAINTING_R Busy	.93063	.27302	.00000	1.0000	1.0000
# in PAINTING_R_Q	.00000	--	.00000	.00000	.00000
FAN_ASS_R Available	1.0000	.00000	1.0000	1.0000	1.0000
FAN_ASS_R Busy	.33916	1.3958	.00000	1.0000	1.0000
# in FAN_ASS_R_Q	.00000	--	.00000	.00000	.00000
NEA(CONV1)	32.139	.14785	1.0000	38.000	34.000

COUNTERS

Identifier	Count	Limit
REJ_MOTORS_C	88	Infinite
FINI_MOTORS112	698	Infinite

Figure F-19. Summary report for cell-112 final assembly.

SIMAN V - License #9999999
Systems Modeling Corporation

Summary for Replication 5 of 5

Project: Run execution date : 3/ 4/1998
Analyst: Model revision date: 3/ 4/1998

Replication ended at time : 14325.0
Statistics were cleared at time: 11460.0
Statistics accumulated for time: 2865.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
BATCHING_Ta	9797.5	.06463	8706.4	10882.	627

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
ASS_WIND_FRAM_R Availa	1.0000	.00000	1.0000	1.0000	1.0000
ASS_WIND_FRAM_R Busy	.78319	.52615	.00000	1.0000	1.0000
# in ASS_WIND_FRAM_R_Q	.00000	--	.00000	.00000	.00000
MACHINING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
MACHINING_R Busy	.96506	.19027	.00000	1.0000	1.0000
# in MACHINING_R_Q	.00000	--	.00000	.00000	.00000
T_BOX_FITT_R Available	1.0000	.00000	1.0000	1.0000	1.0000
T_BOX_FITT_R Busy	.83784	.43994	.00000	1.0000	1.0000
# in T_BOX_FITT_R_Q	.00000	--	.00000	.00000	.00000
ROTOR_ASS_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ROTOR_ASS_R Busy	.58452	.84309	.00000	1.0000	1.0000
# in ROTOR_ASS_R_Q	.00000	--	.00000	.00000	.00000
ENDS_ASS_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ENDS_ASS_R Busy	.46175	1.0796	.00000	1.0000	.00000
# in ENDS_ASS_R_Q	.00000	--	.00000	.00000	.00000
ELECT_TEST_R Available	1.0000	.00000	1.0000	1.0000	1.0000
ELECT_TEST_R Busy	.56591	.87582	.00000	1.0000	1.0000
# in ELECT_TEST_R_Q	.00000	--	.00000	.00000	.00000
PAINTING_R Available	1.0000	.00000	1.0000	1.0000	1.0000
PAINTING_R Busy	.61590	.78971	.00000	1.0000	1.0000
# in PAINTING_R_Q	.00000	--	.00000	.00000	.00000
FAN_ASS_R Available	1.0000	.00000	1.0000	1.0000	1.0000
FAN_ASS_R Busy	.30945	1.4938	.00000	1.0000	.00000
# in FAN_ASS_R_Q	.00000	--	.00000	.00000	.00000
NEA(CONV1)	23.967	.14104	.00000	27.000	25.000

COUNTERS

Identifier	Count	Limit
REJ_MOTORS_C	60	Infinite
FINI_MOTORS132	627	Infinite

Figure F-20. Summary report for cell-132 final assembly.

```
                         SIMAN V - License #9999999
                         Systems Modeling Corporation

                       Summary for Replication 5 of 5

Project:                                        Run execution date :    3/ 4/1998
Analyst:                                        Model revision date:    3/ 4/1998

Replication ended at time      : 14325.0
Statistics were cleared at time: 11460.0
Statistics accumulated for time: 2865.0

                                TALLY VARIABLES

Identifier                 Average   Variation   Minimum    Maximum   Observations
_________________________________________________________________________________

TIS_AL_M                   10805.     .06470     9635.8     12025.        177
TIS_CI_M                   10333.     .06355     9200.8     11455.        175

                           DISCRETE-CHANGE VARIABLES

Identifier                 Average   Variation   Minimum    Maximum   Final Value
_________________________________________________________________________________

ASS_WIND_FRA_R Availab     1.0000     .00000     1.0000     1.0000     1.0000
ASS_WIND_FRA_R Busy        1.0000     .00000     .00000     1.0000     1.0000
# in ASS_WIND_FRA_R_Q      19.794     .02043     19.000     20.000     20.000
MACHINING_R Available      1.0000     .00000     1.0000     1.0000     1.0000
MACHINING_R Busy           .95267     .22289     .00000     1.0000     1.0000
# in MACHINING_R_Q         .00000       --       .00000     .00000     .00000
ASS_WIND_CI_R Availabl     1.0000     .00000     1.0000     1.0000     1.0000
ASS_WIND_CI_R Busy         1.0000     .00000     .00000     1.0000     1.0000
# in ASS_WIND_CI_R_Q       19.841     .01841     19.000     20.000     19.000
MACHINING_CI_R Availab     1.0000     .00000     1.0000     1.0000     1.0000
MACHINING_CI_R Busy        .55183     .90119     .00000     1.0000     1.0000
# in MACHINING_CI_R_Q      .00000       --       .00000     .00000     .00000
# in T_BOX_FIT_S_Q         2.2150     1.1593     .00000     9.0000     8.0000
T_BOX_ST1 Available        1.0000     .00000     1.0000     1.0000     1.0000
T_BOX_ST1 Busy             .99141     .09310     .00000     1.0000     1.0000
# in T_BOX_ST1_Q           .00000       --       .00000     .00000     .00000
T_BOX_ST2 Available        1.0000     .00000     1.0000     1.0000     1.0000
T_BOX_ST2 Busy             .98878     .10654     .00000     1.0000     1.0000
# in T_BOX_ST2_Q           .00000       --       .00000     .00000     .00000
ROTOR_ASS_R Available      1.0000     .00000     1.0000     1.0000     1.0000
ROTOR_ASS_R Busy           .80320     .49499     .00000     1.0000     1.0000
# in ROTOR_ASS_R_Q         .00000       --       .00000     .00000     .00000
# in ENDS_FITTING_S_Q      .00000       --       .00000     .00000     .00000
ENDS_FIT_ST1 Available     1.0000     .00000     1.0000     1.0000     1.0000
ENDS_FIT_ST1 Busy          .53391     .93432     .00000     1.0000     1.0000
# in ENDS_FIT_ST1_Q        .00000       --       .00000     .00000     .00000
ENDS_FIT_ST2 Available     1.0000     .00000     1.0000     1.0000     1.0000
ENDS_FIT_ST2 Busy          .00000       --       .00000     .00000     .00000
# in ENDS_FIT_ST2_Q        .00000       --       .00000     .00000     .00000
ELECT_TEST_R Available     1.0000     .00000     1.0000     1.0000     1.0000
ELECT_TEST_R Busy          .40572     1.2102     .00000     1.0000     .00000
# in ELECT_TEST_R_Q        .00000       --       .00000     .00000     .00000
PAINTING_R Available       1.0000     .00000     1.0000     1.0000     1.0000
PAINTING_R Busy            .48711     1.0261     .00000     1.0000     .00000
# in PAINTING_R_Q          .00000       --       .00000     .00000     .00000
FAN_ASS_R Available        1.0000     .00000     1.0000     1.0000     1.0000
FAN_ASS_R Busy             .33251     1.4168     .00000     1.0000     .00000
# in FAN_ASS_R_Q           .00000       --       .00000     .00000     .00000

                                    COUNTERS

                     Identifier                 Count   Limit
                     ________________________________________

                     REJ_MOTORS_C                  33   Infinite
                     MOTORS_AL                    177   Infinite
                     MOTORS_CI                    175   Infinite
```

Figure F-21. Summary report for cell-160/180 assembly line.