THE DESIGN OF AN INTEGRATED
PRODUCTION AND INVENTORY CONTROL
SYSTEM FOR A TRAFFIC SIGN SHOP

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements For the Degree
Master of Science

by
Wagner Barbosa,
June, 1990
ACKNOWLEDGEMENTS

I would like to thank the ROTARY INTERNATIONAL and the CAPES for their support, without which none of this would be possible.

I also want to thank my advisor, Dr. Ken Cutright, for his dedication on this project. I am especially grateful for the orientation and help he gave me during my course of study.

To the other members of my committee, Professor Ralph Sims and Dr. Clarence Martin, I would like to express my deepest appreciation on their help in the completion of this thesis.

I am also grateful to the people in the West Virginia Department of Highways who provided valuable information and assistance.

To my parents, I want to express my deepest gratitude for their caring and confidence.

Lastly, I dedicate this thesis to my wife, Flavia, who gave me all the care and encouragement, especially during my graduate work. Her love and kindness were my strongest support.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>..................................................................................................................</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Statement of Need</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Objective</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Thesis Organization</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 2: STATEMENT OF THE PROBLEM</td>
<td>..................................................................................................................</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Types of Signs</td>
<td>6</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Standard Signs</td>
<td>10</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Special Signs</td>
<td>10</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Interstate Signs</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Types of Production Operation</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Bill of Materials</td>
<td>21</td>
</tr>
<tr>
<td>2.4</td>
<td>Inventory Policy</td>
<td>24</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Standard Signs and Materials</td>
<td>24</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Special and Interstate Signs</td>
<td>25</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Raw Materials</td>
<td>25</td>
</tr>
<tr>
<td>2.5</td>
<td>Production and Materials Planning</td>
<td>25</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Forecasting</td>
<td>25</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Production and Capacity Planning</td>
<td>26</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Raw Materials Planning</td>
<td>26</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Production Schedule</td>
<td>26</td>
</tr>
<tr>
<td>2.5.5</td>
<td>Delivery</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Information Systems</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>Summary</td>
<td>27</td>
</tr>
<tr>
<td>CHAPTER 3: LITERATURE REVIEW</td>
<td>..................................................................................................................</td>
<td>29</td>
</tr>
<tr>
<td>3.1</td>
<td>Integrated Production Planning and Control Systems</td>
<td>29</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Demand Forecast</td>
<td>32</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Customer Order Control</td>
<td>33</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Finished Goods Inventory Control</td>
<td>35</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Master Production Schedule (MPS)</td>
<td>35</td>
</tr>
<tr>
<td>3.1.4.1</td>
<td>Lot Size</td>
<td>36</td>
</tr>
<tr>
<td>3.1.4.2</td>
<td>The MPS Record</td>
<td>38</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>3.1.5 Material Requirements Planning</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>3.1.5.1 Bill of Material (BOM)</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>3.1.5.2 Lot Size</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>3.1.5.3 The MRP Record</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>3.1.6 Capacity Planning</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>3.1.6.1 Capacity Planning Techniques</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>3.1.6.2 The Capacity Plan Adjustments</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>3.1.7 Scheduling</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>3.1.7.1 Sequencing Rules</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>3.1.7.2 Finite Loading</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>3.1.8 Purchasing and Material Control</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>3.1.9 The Shop Floor Control</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>3.1.10 Warehousing and Shipping</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>3.2 Structured Analysis</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>3.2.1 The Data Flow Diagrams (DFD)</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>3.2.2 The Data Dictionary (DD)</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>3.2.3 The Process Narratives</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>CHAPTER 4: SYSTEM DESIGN</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>4.2 The Top Design</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>4.2.1 External Interfaces</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>4.2.2 The Control Inventory Module</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>4.2.3 The Plan Production Module</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>4.2.3.1 The Plan MPS Sub-Module</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>4.2.3.2 The Plan Materials Sub-Module</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>4.2.3.3 The Control Production Sub-Module</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>4.2.4 The Manage Data Module</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>4.3 The Bottom Design</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>4.3.1 The Control Inventory Module</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>4.3.1.1 The Input Order Sub-Module</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>4.3.1.2 The Allocate Order Sub-Module</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>4.3.1.3 The Control Delivery Sub-Module</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>4.3.2 The Plan Production Module</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>4.3.2.1 The Plan MPS Sub-Module</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>4.3.2.2 The Plan Materials Sub-Module</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>4.3.2.3 The Control Production Sub-Module</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>4.3.3 The Manage Data Module</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>4.4 Summary</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>CHAPTER 5: CONCLUSION</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>5.1 Summary of Research</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>5.2 Coding and Implementation</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>5.3 Recommendations for Future Research</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>REFERENCES</td>
<td>109</td>
<td></td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>343</td>
<td></td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>353</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

The focus of this research is the development of a decision support system for controlling inventories and planning the production of traffic signs for a state highway system. The inventory control function, which control the level of raw materials and finished products is a critical aspect of successful management. Providing good customer service while simultaneously maintaining low inventory levels should be the objective of an inventory control function. Production planning is the coordination of all stages of the production process from forecasting demand to shipping the finished product. The objectives of a production planning system would include the efficient utilization of resources, both human and equipment. The efficient utilization of these resources, in any production system, depends on the availability of raw material and work-in-process inventory. The implication here is that the production planning function is dependent upon the inventory control function, at least from a raw material and work-in-process standpoint. On the other hand, the control of finished goods inventory levels is related to the efficiency of the supply source (which includes the production facilities as well as warehouses or external suppliers). Therefore the inventory control function is dependent, at least partially, on the production planning function. Both production planning
and inventory control are dependent upon each other and therefore should not exist separately. This research will design an integrated system for controlling inventories and planning production which will provide management with a decision support tool for analyzing the effects of its decisions in the organization as a whole.

1.1 Statement of Need

The West Virginia Department of Highways (WVDOH) is the agency responsible for managing the construction and maintenance of the state's roadway system. This system includes all state routes, U.S. routes and the interstate highways. The maintenance of the highway system includes, among other things, the maintenance of the necessary highway signs. The proper maintenance of these highway signs requires that an adequate inventory of replacement signs be available. There are primarily two situations which dictate a need for a sign at a roadway location. The first is to supply newly constructed roadways with the required signs, while the second reason is the continual demand for replacing those signs which are either deteriorated, damaged or stolen. The safety and comfort of the driver in the state's roadway system depends on the signs in the field being maintained in good condition. A shortage of the required signs can seriously compromise the safety of the motoring public. Furthermore, the absence of a critical sign may result in an accident, which may result in
the state of West Virginia being involved in an expensive law suit.

To ensure that the state roadway system has an accessible supply of the necessary signs, the WVDOH maintains warehouses in different locations (districts) in the state. A central sign shop is responsible for supplying these warehouses with the required sign inventory. The following problems in the WVDOH operation have recently been identified: 1) The inventory levels at the warehouses and at the sign shop are continually increasing, without a corresponding increase in the signs demand. 2) The sign shop consistently shows a high level of back orders, which has recently led the WVDOH to periodically subcontract a large number of signs from external suppliers. 3) The high level of back orders at the sign shop continually increases the lead times for the manufactured items. For this reason the districts have become more conservative and increased their inventory levels, thereby further affecting the back order level at the sign shop. 4) The sign shop experiences regular shortages of raw material, seriously compromising the production system. 5) There is frequent occurrence of orders which require special fabrication, making the scheduling of production even more difficult. 6) The districts believe that the procedure for the distribution of signs to the districts is not fair.

These factors have contributed to a situation which led to a need to develop some methodology for reducing raw
material inventories, production costs, and finished product inventory costs, while maintaining an adequate supply of signs on the state roadways.

1.2 Objective

The objective of this thesis is the development of the logic for a computerized system for planning and scheduling the production of the traffic signs at the central sign shop, controlling the orders issued by the districts, and planning the purchase of raw materials and finished signs from outside suppliers. The proposed system will provide management support for the following functions: controlling sign orders submitted by the districts; planning the production at the sign shop, based on the demand forecast and available capacity; planning raw material purchases to meet the production plan; planning the purchase of finished signs (based on the demand forecast for these items); scheduling the production at the Sign Shop; and controlling the delivery of the signs to the districts.

This decision support system should produce a reduction in the inventory levels of both raw materials and finished signs as well as a more efficient utilization of the resources at the sign shop; thereby reducing the overall costs of maintaining the supply of signs to the state's highways. Other benefits include a significant reduction in the amount of manual work, an increase in the accuracy of data, and the availability of information (which may be helpful in other
management decisions) which is not readily available in the current system. Operationally, the system will be designed to be simple and user friendly, requiring no computer skills from the user. The use of a data base will enable future modifications in the product or raw material structure to be easily made. Override capabilities will be provided to enable the user to deal with unusual situations which may occur. Although the proposed system will be developed particularly for the WVDOH, the knowledge used in the development of this system can be adapted to any organization with similar operations.

1.3 Thesis Organization

Chapter Two gives a more detailed analysis of the problem, showing an analysis of the WVDOH operations. Examples of the different categories of signs, operations sequence, and bill of materials are also provided. Chapter Three reviews the literature related to the problem as well as the explanation of the tools used for designing the system. In Chapter Four the design of the system is explored, as well as an explanation of the approach used for developing the logic. Chapter Five discusses the expected results and makes recommendations for implementation and future research. The Appendix A contains all the data flow diagrams, the structured english logic of the functions and the data dictionary.
CHAPTER 2
STATEMENT OF THE PROBLEM

The installation and maintenance of the traffic signs in the state roadway system is one of the functions performed by the ten district locations throughout the state. The locations are shown in Figure 1. Each district has its own maintenance crew and a warehouse for storing signs and materials until they are required for installation in the field. In addition to that, each district also maintains a small crew (with its own sign inventory) in each county for small repairs. To obtain signs for the district warehouse inventory and the smaller county locations, the districts issue orders to the central sign shop located in Charleston, West Virginia.

2.1 Types of Signs

There are three major categories of signs as shown in Figure 2. The first category is called standard signs. As the name implies, these signs must obey fixed specifications in their fabrication regarding to their color, size, material and printed message. In the second category are those signs which show individual, one of a kind, messages. Due to their purpose, these signs are called special signs. In the last category are the interstate signs. These signs also show special messages but differ from the special signs by the material from which they are made. Both the special and the
Fig. 2 - Examples of Signs of each Category
interstate signs are standardized in material, color and size, while the message depends on the purpose and the location of the sign in the field. The central sign shop makes the majority of the signs required. All special and interstate signs are made at the sign shop as well as most of the standard signs; however some standard signs are purchased from external suppliers.

Although there are important differences among the three sign categories, the following common procedures are used in the manufacture of each sign:

- Over a flat surface apply reflective tape as the background.

- Over the background apply a reflective message or symbol or both.

Besides these three categories of signs, the WVDOH also uses a great variety of other items in the field. These items include materials used to install the signs (e.g. U channel posts and bolts) or other kinds of signs (e.g. barricades, traffic cones and flashing lights). What differentiates these items from the main categories of signs is their origin. The materials and signs described above cannot be produced at the central sign shop and are always purchased from external suppliers. To distinguish these items from the three kind of signs described, they will be referred to as materials.

Note that there is also a difference between these materials and the raw materials used in producing the three
categories of signs. While these materials have an independent demand, the demand for the raw material used in manufacturing signs is driven by the demand for the finished signs. For the purpose of this study the material used in manufacturing signs will be referred to raw materials.

2.1.1 Standard Signs

The standard signs are all made from an flat aluminum sheet, which is cut in the required shape and dimensions. The sign shop buys these sheets already cut in the required shape and dimensions with the appropriate finish (called pre-cut) from their suppliers. A small inventory of uncut aluminum flat sheets is maintained for special big signs or for use in the event of a raw material shortage in the pre-cut blanks. The reflective background is machine applied over the pre-cut sheet, and the message is then screened using a reflective ink. Some standard signs require double screening (more than two colors signs) or a hand applied message (for example signs showing specific distances or speed limits).

2.1.2 Special Signs

The special signs can be made from either a pre-cut sheet or an aluminum sheet that is cut at the sign shop, depending on the sign size. The reflective background is machine applied, if possible, or hand applied if the sign is too big for the sheeting machine. Some of the larger signs also
require ribbing on the back of the sign for additional support in the field. The message on these signs are made from either reflective tape or special characters called demountable copy. Demountable copy is ready to apply letters and symbols made from reflective material and purchased from external suppliers. These messages are always hand applied since they are one of a kind signs. They are used only on large special signs and interstate signs. All hand applied signs (background or message) must be heat treated to ensure the proper application of the reflective tape.

2.1.3 Interstate Signs

The interstate signs are made from extruded aluminum panels instead of flat sheets. The extruded panels are cut in the desired length before being covered with the reflective tape. With the background applied, these panels are then assembled together to make a sign according to the desired dimensions. The last operation is the application of the demountable copy creating the desired message.

2.2 Types of Production Operation

The sign shop presently uses the following equipment in the fabrication of the signs:

- 1 Shearing Machine
- 1 Sawing Machine
- 1 Cornering Machine
- 2 Hole-Punching Machines
- 1 Sheeting Machine
- 2 Screening Machines
- 2 Ovens (1 Vertical and 1 Horizontal)
- 1 Letter Cutting Machine

A description of each of these operations is given below.

Figures 3 to 7 show the different types of signs which are included in the same category and the possible combinations of operations involved in their fabrication, as well as the sequence of these operations.

01 - Shearing

This describes the process of cutting the aluminum sheet to the required size. This operation is not very common, except when making large signs, since the sign shop usually makes signs from the pre cut flat sheet.

02 - Cornering

This is required for reshaping the 90 degree corners of the aluminum signs into curved corners as specified in the signs manual [67]. If the sign is made from pre-cut flat sheet, this operation is not necessary.

03 - Hole Punching

For installation in the field, holes are necessary for bolting the signs to the posts. This operation is required
Fig. 3 - Sequence of Operations for Standard Signs
Fig. 4 - Sequence of Operations for Special Signs Using Pre-Cut and Hand Applied Message.
Fig. 5 - Sequence of Operations for Special Signs Not Using Pre-Cut and Using Hand Applied Message.
Fig. 6 - Sequence of Operations for Special Signs Not Using Pre-Cut and Using Demountable Copy.
Fig. 7 - Sequence of Operations for Interstate Signs.
only for the aluminum signs which do not use the pre cut flat sheets. Pre cuts have pre-punched holes.

04 - Sawing

Cuts the extruded panels to the required length.

05 - Sanding

Obtain a smoother surface on the extruded panels and aluminum sheets, which were previously sheared and cornered.

06 - Sheeting and Trimming

This operation is responsible for applying the reflective background tape over the aluminum flat sheets (or extruded panels) and trimming the excess material. The sheeting operation is done in the sheeting machine, if the size of the sign permits (if it fits in the machine), or manually otherwise.

There are four basic types of reflective materials used as background:

- Engineering Grade for machine application
- Engineering Grade for hand application
- High Intensity for machine application
- High Intensity for hand application

The type and the color of the background material that must be used is defined in the signs manual [67]. The trimming is done manually following the sheeting operation.
07 - Screen Preparation

Before signs can be screened, the screen must first be prepared. This screen contains the sign symbol or message and has to be made according to the dimensions and details specified in the signs manual [67]. Each different sign type takes its own screen, which has to be prepared every time that specific sign is to be screened. Although this is a time consuming operation, prior analysis made by the WVDOH management showed that it is economically and operationally infeasible to maintain an inventory of all the screens used for screening standard signs (the WVDOH uses approximately 1,000 different standard signs).

08 - Message Screening

This operation screens messages or symbols on the standard signs after the sheeting and trimming operation. A screen with the desired message is prepared and secured in the screening device. The blanks (with the reflective sheeting already applied) are positioned in the machine and the message is printed using reflective ink. The type of the ink used depends on the background material.

09 - Drying

After screening, the sign must be dried before being available for storage or delivery. There are two basic ways to
dry a screened sign. The first is to let the sign dry naturally, at the ambient temperature, while resting in racks. The second is to put the signs into an oven for fast drying. Natural drying is the preferred method, the second method is used only for emergency situations. If a sign has to be screened more than one time (three colors or more), it has to be dried after each screening operation.

10 - Hand-Lettering

This is the process of applying the individual messages on the special signs. The message is typed in a microcomputer attached to a cutting machine. The computer calculates the letter intervals and the size of the message and controls the machine which cuts a reflective self-adhesive tape fed by the operator. After cutting the message, the operator than hand applies the message on the sign. The material used is basically the same as is used in the sheeting process.

11 - Heat-Treatment

As mentioned before, all the signs which use hand applied reflective messages must be heat treated before being available for storage or delivery. This heat treatment prolongs the life of the sign and enforces the reflective material application.

12 - Ribbing
When a special sign is too large, this process is done to ensure that the sign will sustain well in the field. The ribbing reinforces the sign, enabling large signs to be made of flat sheets.

13 - Panel Assembly

This applies only to extruded panel signs. After sheeting and trimming the extruded panels, they are assembled side by side to make the sign in the required size. The panels are bolted together according to instructions given in the sign manual [67]. This operation forms a flat base on which the message is applied.

14 - Button-Down

This also only applies to extruded panel signs. The demountable copy characters are affixed over the flat base of the extruded panel signs to create the message.

2.3 Bill of Materials

The product structure which applies in this situation is known as the inverted pyramid type. In this type of product structure, many items are made from a limited number of raw materials. In Figures 8 and 9 a general bill of materials is shown, depicting the raw materials used in each different category of signs. This general format is used rather than illustrating all product structures due to the large number
Fig. 8 - Bill Of Materials for Standard and Special Signs
Fig. 9 - Bill Of Materials for Interstate Signs
of different signs.

Another important characteristic of the bill of materials for the traffic signs is that it has few levels. This will allow for a simplified materials planning and production scheduling procedure.

2.4 Inventory Policy

The sign category determines the appropriate inventory policy at both the districts and the central sign shop.

2.4.1 Standard Sign and Materials

Since the standard signs and materials are frequently required, the districts, as well as the sign shop, try to maintain them in inventory. Minimum and maximum inventory levels are used to control the reorder point and the order quantity. Theoretically whenever the reorder point is reached (minimum level), the districts should order enough items to replenish their inventory up to the maximum level. However this study determined that this rule was not being followed by the districts due to the following reasons.

- The accounting system being used does not charge the district for signs until they are installed in the field, thereby encouraging them to build high inventory levels.

- The inaccuracy of the minimum and maximum levels (which are based on the average of the previous years usage) induces the districts to apply its own ordering policy.
2.4.2 Special and Interstate Signs

The special and interstate signs are one of a kind. No inventory is kept and an order is issued only when the sign is needed in the field.

2.4.3 Raw Materials

To order the raw materials, the sign shop uses the same policy the districts use for ordering standard signs and materials.

2.5 Production and Material Planning

The sign category also determines the production and purchase policies. Standard signs are made-to-stock due to the long set up time required to prepare the screen and the repeated demand. Interstate and special signs are always made-to-order. Materials, raw materials and subcontracted signs are bought-to-stock.

2.5.1 Forecasting

No formal forecasting procedure was being used at the time this study was initiated. The was due primarily to the large number of finished products, and the relatively low demand for each of them.
2.5.2 Production and Capacity Planning

No formal production planning policy exists. The production orders are issued to satisfy back orders from the districts and replenish the sign shop inventory.

2.5.3 Raw Materials Planning

As already mentioned, the raw materials are purchased based on reorder levels and lot-for-lot order policy. Since no production plan is being used, there is no way of predicting raw material requirements.

2.5.4 Production Schedule

To schedule production at the sign shop, reports from two different computer applications (explained later) are used to manually identify which products the sign shop should make and which it should buy, the quantities to be made or bought, and the priority for fabrication. This process results in a list of items that should be made. The priority rules for standard signs are based on back order level. Those items back ordered longest have the highest priority for fabrication. The scheduling of special and interstate signs are based on the first come-first serve rule.

2.5.5 Delivery

The distribution of materials and signs from the central sign shop to the districts follows a fixed delivery schedule.
Basically, there is one delivery per month to each district, which then must deliver to the counties within the district.

2.6 Information Systems

A computerized accounting system is used to control the inventory levels at both the central sign shop and districts. Each district, as well as the central sign shop, has the capability of retrieving inventory data or updating the inventory system as required. In addition to the accounting system, the WVDOH has developed a program to control orders coming from the districts. This program is used to obtain information on back orders for use in production scheduling and material and sign purchasing. The state accounting system and the WVDOH back order program do not communicate, leaving much of the work to be done manually.

2.7 Summary

The lack of a production and materials planning policy and a weak inventory policy has resulted in the following problems in the WVDOH operation:

- Extremely high levels of inventory at the districts and the central sign shop;
- Erratic and unbalanced work loads at the sign shop;
- Increasing level of subcontracting without increasing demand;
- High level of back orders and permanent "customer"
dissatisfaction.

Additionally, the lack of communications between the two existing computer programs requires a great amount of manual work and data redundancy, without offering the minimum information required for applying better management policies. The aforementioned problems have created the need for a system to plan the production of manufactured signs as well as control the inventory of finished signs, purchased signs, materials, and raw materials. The system will use the data available from both existent applications to reduce the manual work and provide accurate information to provide assistance in managerial decision making.
CHAPTER 3

LITERATURE REVIEW

3.1 Integrated Production Planning and Control Systems

The goal of integrated production planning and control systems is the coordination of all phases of the production process. Figure 10 depicts a simplified version of such a system. Although there can be variations, production planning and control systems generally follow this pattern.

The system is divided into three phases. The first phase, called the front end, includes all the activities related to production planning. This module analyzes the demand forecast, the customer orders received, and finished goods inventory to plan the production requirements over a planning horizon. The result of this phase is a master production schedule (MPS), i.e. the planned production for each period of the planning horizon.

The second phase encompasses those activities related to capacity and materials planning. This phase receives information from the production planning phase and explodes the MPS to determine the requirements for subassemblies, components and materials. The capacity planning module is responsible for checking feasibility in terms of machines and labor, and planning production accordingly. The scheduling activity establishes the priorities for the production plan and schedules the work orders according to available capacity.
Fig. 10 - Integrated Production Planning and Control Systems.
during the scheduling horizon. The results of this phase are the production and material plans and the production schedule.

The third part in this system is the back end. This phase is responsible for controlling the scheduling and materials planning as well as the shipping and warehousing of finished goods. This phase provides feedback to the system regarding completed production orders, material receipts, outstanding purchase orders, and the finished products shipped or stored in inventory.

Because integrated production planning and control systems must often interact with other systems (such as the accounting, purchasing, and inventory systems), most are designed with enough flexibility to interface with databases shared with other parts of the organization. There are many different production planning and control systems being used by companies today. They are known by such names as Master Production Schedule Planning System, Advance Planning and Control System, Manufacturing Resource Planning (MRP-II) to name a few. These systems can be either a general use software package or one designed for a specific company only. Although differences can be found in both complexity and operational procedures, all production planning and control systems follow the basic design shown in Figure 10.

Because of the differences found in the organizations, a system that works well in one company, may not function effectively in others. For this reason software packages have
generally been designed in a modular fashion which permits modifications to allow the software to be adapted to a company's specific operational standards. Tailoring the system to the specific needs and requirements of each company sometimes requires major adaptations in the software, making the use of these software economically infeasible for some companies. This is one of the reasons that, although some companies have been using packaged software successfully for years, many companies have developed their own production planning and control system rather than purchasing general use packaged software.

The following sections discuss each module in the basic design of such systems and the interactions among and between them.

3.1.1 Demand Forecast

The demand forecast module is the module responsible for predicting future demand. This module can be either a computerized or manual procedure which computes demand for a given time horizon. Forecasting techniques can be classified as either qualitative or quantitative. Qualitative forecasting is the prediction of demand for a product based on subjective methods i.e., a marketing group, based on customer preference. Quantitative forecasting analyzes the historical data to predict future demand. Forecasting techniques using historical data assume that future demand will follow historical data
trends.

Forecasting demand for products is sometimes impractical because large variation in the demand causes large forecast errors. Because of this, some companies prefer to forecast families of products rather than forecasting demand for individual products. The forecasting for families, also called aggregated forecasting, can reduce this error. However, the adoption of this technique always implies that disaggregation will be necessary sometime during the planning process.

Other firms prefer to adopt the individual forecasting techniques, but only for class A products. Class A products are those defined by ABC (Pareto's) analysis as the few products that represent the majority of the revenue (or demand). One can find more details about ABC analysis in Slaybaugh [60]. Companies which operate exclusively on a make-to-order basis will not use a formal forecasting procedure, but rather plans production based only on the receipt of customer orders.

Since this thesis does not include the design of a forecast module, we will not treat this subject in detail. Forecasting techniques are explained in Bedworth and Bailey [6], and in Abraham and Ledolter [1].

3.1.2 Customer Order Control

The input for this module is the customer orders. The module monitors these orders until they are completed, i.e.,
when all items in the order are delivered to the customer. The systematic control of the orders is the foundation for effectively controlling the production and inventory of finished goods.

Two methods of controlling orders are the firm planned order and time fencing. According to Vollmann, Berry and Whybark [65], a firm planned order is a planned order with times and quantity that will not be changed (once the order is received and entered in the system, neither the order quantity or due date will be changed). Time fencing is the procedure which establishes a time limit after which no changes in orders or new orders are accepted for that period. The establishment of time fences is related to the lead time. If new orders are accepted after the time fence, there will not be enough time to revise the material and capacity plans.

These techniques give stability to the master production schedule. The adoption of one of these techniques can reduce the necessity for future adjustments in the MPS, an occurrence which can compromise the planning process.

Another concern when controlling orders is the relationship between actual order quantities and the forecasted demand. The order control module must have a systematic procedure to control the receipt of orders and decide the amount of the orders that will be supplied from inventory and the amount that will be produced.
3.1.3 Finished Goods Inventory Control

This activity controls allocation of finished goods inventory to satisfy customer orders. This is necessary to distinguish the inventory which has been promised from that which is available.

The promised units are those which have been allocated to customer orders and are awaiting delivery. On the other hand, the available inventory is those units which are not yet allocated to a customer order. Note that this procedure applies only to make-to-stock or mixed make-to-stock and make-to-order companies, since exclusive make-to-order companies do not store finished goods for future orders and the production planning is linked to the customer order only.

3.1.4 Master Production Scheduling (MPS)

Master production scheduling is the procedure which plans the production of finished goods based on demand forecast, customer orders not yet allocated, and the available finished goods inventory (including scheduled receipts).

It is important to observe that MPS is not a marketing forecast but an anticipated schedule for manufacturing end products. The forecast of sales is a critical input to the MPS, but the MPS does not usually equal the forecast. The MPS considers not only marketing demand but also present inventory levels, capacity limitations and the desire to utilize capacity (as much as is reasonable). The MPS may include items
which are needed to satisfy future demand and omit some items which are required to satisfy current demand.

3.1.4.1 Lot Size

In a make-to-stock company where most items are carried in inventory, MPS provides the quantity and date (time period) that finished goods will be produced. In a make-to-order company, the MPS unit is simply the particular end item (or set of items) that constitute a customer order.

Because of its simplicity, the economic order quantity (EOQ) formula is often used as a decision rule for deciding the lot size for MPS in a make-to-stock environment. Care must be exercised, however when using EOQ to compute the MPS quantities, since the assumption of constant demand in EOQ is not necessarily true in many cases. Another problem with using the EOQ model is that it analyzes each end product individually rather than the overall system. The results are then not necessarily optimum for the overall system, even if the optimum point is found for each individual item. Further, the use of EOQ also assumes an exact determination of the inventory holding and replenishment costs, which is generally not possible.

These facts do not necessarily invalidate the use of the EOQ quantity, but are intended to make the scheduler aware that the EOQ formula is not a precise tool and should not be used as such. The EOQ is a number which can be adjusted as the
practicality of the operation dictates, i.e. rounding the order quantity to the nearest feasible run quantity. Details of the EOQ model can be found in Love [34].

Another available lot sizing technique is the lot-for-lot technique. In this method, the lot size is exactly the amount needed. It is frequently more economical to produce amounts that exceed the required units, there are possible situations where the use of lot-for-lot can be used. An example of such a situation is one in which the set up costs are small compared to the total cost of producing an item. In this case, there is no reason to establish a minimum lot size and only the required quantity will be produced. Practical limitations can however limit the use of the lot-for-lot technique, i.e., if items are packaged as a group, the release amounts would have to be in multiples of the lot or group size. In this case adaptations can be made to overcome these limitations.

The periodic order quantity (POQ) is another technique for calculating order quantities or lot size. The basic difference between the EOQ and POQ is that POQ accounts for demand that is not constant over time. POQ determines the average number of time periods (N) covered by the EOQ, and determines the lot size by computing the amount required to cover the expected demand for those N periods. In some situations, the use of POQ can reduce the inventory carrying cost associated with fixed lot sizes.

The Wagner-Whithin algorithm (WW) is a procedure for
determining the minimum-cost ordering plan for a time-phased requirements schedule. This procedure uses dynamic programming to evaluate all the possible ways of ordering material to meet the demand for the planning horizon. Although, in some situations, significant improvements can be obtained by the use of such method, care must be exercised when considering the use of WW, as the computational requirements considerably increase with the number of periods.

More detail on these techniques and other lot sizing techniques can be found in Vollmann, Berry and Whybark [65], Love [34], and Mcleavey and Narasimhan [38].

3.1.4.2 The MPS Record

Figure 11 shows an example of the calculations involved in computing the MPS. Note that the planning procedure is done over a fixed planning horizon, where the length of the planning horizon is related to the cumulative material lead time and the forecast accuracy. That is to say, the planning horizon must be long enough to buy the required materials and produce the final product. This time is known as the cumulative material lead time. On the other hand, as the planning horizon gets longer, the forecast accuracy decreases considerably. The trade off then is to establish a planning horizon that minimizes forecast errors while giving the scheduler time for planning the materials and capacity requirements.
End Product: 021 - 010101

<table>
<thead>
<tr>
<th>Period</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forecast</td>
<td>15</td>
<td>15</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Orders</td>
<td>17</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned Inventory</td>
<td>20</td>
<td>*</td>
<td>**</td>
<td>18</td>
</tr>
<tr>
<td>Available</td>
<td>3</td>
<td>18</td>
<td>6</td>
<td>-9</td>
</tr>
<tr>
<td>Available to Promise</td>
<td></td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>MPS</td>
<td></td>
<td></td>
<td></td>
<td>***</td>
</tr>
</tbody>
</table>

* On Hand = 20 Units

** Scheduled Receipt = 30 Units in period 2

*** Lot Size = 30 units

**** Available to Promise = 30 units in MPS
- 9 promised to period 4

Using a lead time of 3 periods, the MPS should be placed at the end of the first period.

Fig. 11 - The MPS Record
As shown in Figure 11, the forecast row contains the information provided by the demand forecast module and the orders row represents the orders received which are not yet satisfied by inventory. The convention is to use the greater of these two in the MPS computation. The available row is the planned inventory less forecast (or orders). For the first period, the planned inventory is the inventory on hand added to the scheduled receipt. The planned inventory for the remaining periods is the available in the preceding period, if it is positive, or zero, if it is negative.

The MPS appears in those periods where planned inventory added to the available to promise is not enough to satisfy the forecast (or orders). The available to promise is simply the quantity in the MPS issued which has not been committed to any period. Note that there is a major difference between the available to promise and the scheduled receipt. The available to promise represents the units available in the MPS being planned while the scheduled receipt is related to those production or purchase orders already planned and expected to be completed or received during the period. The available to promise in the MPS exists only in situations where the quantity ordered is larger than the units required. It reflects the excess ordered and available to satisfy other period requirements.

The result of the MPS computation is the number of units required, in each period of the planning horizon, to meet
demand. Note then, when planning the production, the cumulative material lead time of the item has to be used to determine when the production and the material orders are to be placed in order to enable the production of the item at the right time. Production lead time structure is given in more detail in Gessner [22].

3.1.5 Material Requirements Planning (MRP)

The material requirements planning is the first activity in the second phase of the integrated production planning and control system framework. The material requirements planning receives the MPS record for all finished goods and explodes them into their subassemblies, components and parts using the bill of material record.

In some productions systems, the capacity planning comes first due to the simplicity of the products. The majority of systems however, plan the subassemblies and components required for production before attempting to check the capacity requirements.

3.1.5.1 Bill of Material (BOM)

According to Mcleavey and Narasimhan [38], the bill of materials can best be described as a list that specifies the quantity of each item, ingredient, or material needed to assemble, mix or produce end products. The BOM record is then a systematic way of storing the information to be used by the
MRP module. An example is shown in Figure 12.a.

There are basically three types of bill of materials structures found in different organizations (Figure 12.b). The first structure is called the pyramid type and is appropriate when a limited number of standard items are made from many components and parts. The second form is appropriate when many items are made from a few common components which use a large number of low level parts. The third form is called the inverted pyramid, and as the name suggests, it is the opposite of the first form. This structure is appropriate when many different end items are made from few components and low level parts.

One important aspect in determining the complexity of the structure of the products is the number of levels in the bill of materials. The higher the number of levels, the more complex the product structure.

3.1.5.2 Lot Size

The lot size used in the MRP computations can also be computed by using the techniques already discussed in section 3.1.4.1 of this thesis. Again EOQ models are the most popular technique due to their simplicity. For parts that are purchased from outside suppliers, lot sizing models such as EOQ can be changed to consider discounts for large quantities, when appropriate. Bedworth and Bailey [6] give more details about EOQ models for purchased parts. The same concerns about
1. Pyramid Structure
   Few different end products using many components.

2. Many different end products using many parts but few components

3. Inverted Pyramid
   Many different end products using few components and parts.

Fig. 12.a - Different Types of Product Structure

Fig. 12.b - Example of the Bill of Material and its Structure
the EOQ model discussed before are also valid here.

3.1.5.3 The MRP Record

The logic for the MRP record calculations are the same as in the MPS calculations. Figure 13 shows the MRP record. Note that it is also computed over a fixed planning horizon (usually the same used in the MPS).

The only basic difference between the MPS and MRP calculations is in the demand computation. In the MPS, a forecast or customer orders are used as the demand, while in the MRP the demand (called gross requirements), is derived from the MPS explosion using the bill of materials. It represents the total number of units required by the MPS.

The planned inventory and the available rows have the same meaning as in the MPS analysis and the MRP available to promise row is also similar to the MPS available to promise in that it represents the units not used to supply gross requirements.

The lead time depends on the nature of the part. If it is a subassembly that has to be produced, the cumulative material lead time must be used. If it is a purchased part, the supplier's lead time is used.

3.1.6 Capacity Planning

The capacity planning module checks the capacity requirements of the production plan against the capacity
Part: 019 - 152124

<table>
<thead>
<tr>
<th>Period</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Requirement</td>
<td>300</td>
<td>250</td>
<td>350</td>
<td>180</td>
</tr>
<tr>
<td>Planned Inventory</td>
<td>590</td>
<td>*</td>
<td>290</td>
<td>40</td>
</tr>
<tr>
<td>Available</td>
<td>290</td>
<td>40</td>
<td>- 310</td>
<td>- 180</td>
</tr>
<tr>
<td>Available to Promise</td>
<td></td>
<td></td>
<td>***</td>
<td>10</td>
</tr>
<tr>
<td>MRP</td>
<td></td>
<td></td>
<td>**</td>
<td>500</td>
</tr>
</tbody>
</table>

* On Hand = 90 units
* Scheduled Receipt = 500 Units in period 1
** Lot Size = 500 units
*** Available to Promise = 500 units in MRP
   - 310 promised to period 3
   - 180 promised to period 4

Using a lead time of 2 periods, the MRP should be placed in the first period.

Fig. 13 - The MRP Record
available in terms of machine and labor resources, and adjusts it in such a way that the available capacity is used effectively.

This activity uses the MRP records from the material requirements planning as input. The capacity required to produce the subassemblies, components and end products planned for each period is computed and checked against the available capacity to detect overloads or underloads in each resource.

The result of this procedure is the detection of the bottleneck and non-bottleneck resources. With this information, the scheduler can take the required action to make the production plan feasible in terms of capacity.

3.1.6.1 Capacity Planning Techniques

Four basic techniques are available for capacity planning. The first technique is called capacity planning using overall factors. This technique is usually done manually. The required input is the MPS records rather than the detailed MRP records. Standard planning factors based on historical data or past experience are used to determine the capacity required for all resources according to the number of end products planned. When the planning factors are applied to the MPS data, overall manpower or machine-hour capacity requirements can be estimated. Overall factors is a very simple technique and the data requirement is minimal. As a consequence, the capacity requirements approximations
calculated for each individual resource are only valid to the extent that product mixes remain constant. More details on capacity planning using overall factors can be found in Vollmann, Berry and Whybark [65].

The second technique is called capacity bills. In this technique, a bill of routing is required, and standard direct labor and machine hour data must be available for each operation. This information is stored in the capacity bills for capacity requirements computations. Figure 14 shows an example of a capacity bill record. The input for the capacity bill planning are the MRP records. Capacity bills requires more data processing than the overall factors technique since it takes into consideration the changes in the production mix. The result is a more accurate capacity requirements calculation. The complexity of the capacity bill record may differ according to the company needs. The one shown in Figure 14 provides the hours required for each production unit in each resource as well the set up time required for the operation.

The computation involved in the capacity bills is simple. The capacity requirement for each MRP record in each resource is calculated by multiplying the MRP quantity by the per unit time required for the resource, and then adding the set up time. The result of this computation is the capacity requirements at each individual resource.

The third technique for capacity planning, resource
### STOP SIGN 30 X 30

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Product Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Sheeting</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Trimming</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Screen Preparation</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Screening</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Drying</td>
</tr>
</tbody>
</table>

### Capacity Bill

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>RESOURCE</th>
<th>OPERAT.TIME</th>
<th>SET UP TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheeting</td>
<td>Sheeting Machine</td>
<td>5 min/unit</td>
<td>15 min</td>
</tr>
<tr>
<td>Sheeting</td>
<td>Sheeting Operator</td>
<td>5 min/unit</td>
<td>15 min</td>
</tr>
<tr>
<td>Trimming</td>
<td>Trimming Operator</td>
<td>4 min/unit</td>
<td>0 min</td>
</tr>
<tr>
<td>Screen Preparation</td>
<td>Scr. Prep Operator</td>
<td>0 min/unit</td>
<td>120 min</td>
</tr>
<tr>
<td>Screening</td>
<td>Screening Machine 1</td>
<td>3 min/unit</td>
<td>15 min</td>
</tr>
<tr>
<td>Screening</td>
<td>Screening Operator</td>
<td>3 min/unit</td>
<td>15 min</td>
</tr>
</tbody>
</table>

Fig. 14 - Capacity Bill and Product Route Example
profiles, is more complex. It takes into account the specific timing of the projected workload at individual centers.

In resource profiles, the production lead time is used to provide time-phased projections of the capacity requirements in each resource. In addition to all the information used in capacity bills, the resource profiles technique also needs the production lead time for each end product and component. The computation is much more complex since it considers the sequence in which the operations are loaded at the resources.

The last technique is called capacity requirements planning (CRP). CRP also utilizes the MRP record as input and follows a procedure similar to that used in resource profiles. Additionally, it considers the capacity of finished or work in process inventories of both components and end products, so that only the capacity needed to complete the remaining work on open orders is considered in calculating the required resources capacity.

The last two techniques, especially the CRP, give the scheduler more accurate information in terms of capacity requirements. However, the price for this accuracy is more complex computations and a larger data base. The decision as to the technique to be used is then related to the desired degree of accuracy and procedure complexity. Before attempting to use complex procedures, one should first analyze the required degree of accuracy since simpler techniques can be used when simpler approximations are sufficient for decision
3.1.6.2 The Capacity Plan Adjustments

The techniques described above provide the scheduler with information on which to base his decision. If there is a difference between the capacity requirements and the available capacity, either the material plan or the capacity available has to be revised.

If the decision is to change available capacity, the options are hiring or laying off employees, requiring overtime or reducing operation times, and increasing or decreasing the number of machines.

Changes in the required capacity can be obtained by increasing the level of parts subcontracted, changing due dates, or influencing demand.

One important aspect to be considered in managing the capacity is to give the scheduler the capability of performing a what if analysis as he makes changes in the material plan or in the capacity available.

The result of the capacity planning should be a feasible production plan for each of the periods in the planning horizon.

3.1.7 Scheduling

The scheduling module receives the production plan from the capacity planning module and uses priority rules to
schedule the production plan.

The capacity plan contains the production orders to be completed during the planning period but does not give the sequence of that production. The responsibility of the scheduling module is to establish this sequence using priority rules. Priority sequencing rules are just what the name suggests: rules for determining the job sequence.

3.1.7.1 Sequencing Rules

The three basic goals, often conflicting, when scheduling are: avoid job lateness, reduce job production lead time, and increase capacity utilization. It is easier to meet due dates if the work load in the production work centers is reduced, but doing so would affect the utilization of capacity. On the other hand, if extra jobs are released to the shop, the capacity utilization would increase considerably but so would the lead times.

Due to the complexity of most scheduling problems, caused in part by these conflicts, optimization methods are generally only applicable to relatively small problems. Large and medium scale problems are usually solved with heuristic procedures called dispatching or sequencing rules. There are a large number of sequencing rules that have been discussed in the research and utilized in practice. Some of the most common rules are described below:

- First come first serve selects the jobs according to
date they arrive in the shop.

- Shortest processing time gives priority to those jobs that require less production capacity in the work centers. This rule is found to reduce work in process inventory and production lead times.

- Earliest due date works well for criteria associated with job lateness.

- Least work remaining is an extension of the shortest processing time rule since it gives priorities to jobs according to the processing time for the remaining operations in the job.

The design of any scheduling module must define the sequencing rules which will be used to give priority for production.

3.1.7.2 Finite Loading

The scheduling horizon is usually shorter than the planning horizon. While the production planning is usually in months or quarters, scheduling is usually done over weeks or even days. Because of this, the scheduling techniques are generally more accurate than those used in the capacity planning on determining the capacity requirements at each resource.

Finite loading is a technique used in scheduling analysis. The concept of finite loading is the simulation of what is likely to happen on the shop floor, given the current
manpower and machine availability. This technique uses jobs operation and set up times to load each job at the resources.

The finite loading approach will only schedule work in a work center until its capacity is exhausted and will not change job due dates. Finite loading does not solve the problem of under capacity but rather informs the scheduler of the results of such for further action. The technique works as a what if analysis to help the scheduler adjust the schedule according to his needs and limitations.

The complexity of the finite loading analysis can vary. In some cases, it is possible to have a detailed schedule for each job through each work center, with the starting and finishing dates. However, when simpler approaches are satisfactory, they should be used, since significant reductions in the computation time and data requirements can be achieved.

3.1.8 Purchasing and Material Control

The back end of an integrated production planning system is the phase responsible for completing the capacity and material plans. The purchasing and material control module is responsible for informing the system of the status of the purchase orders and updating the raw materials inventory.

The information regarding the raw materials orders and the period in which they are expected to be received is crucial in performing the material requirements planning. The
same importance is given to the on hand inventory since it is used by the material requirements planning and possibly the scheduling module (in some designs, the raw material availability is checked before the job is scheduled).

The input regarding the outstanding purchase orders and those received can be done manually or by an interface with the accounting system, if available.

3.1.9 The Shop Floor Control

Executing the production plan and providing feedback regarding the end products produced are the responsibilities of this module. The scheduling module must know which work orders have been completed in order to reconsider those back orders the next time the scheduling is performed. Similarly, the capacity planning module needs to consider the additional capacity requirements for the back orders.

3.1.10 Warehousing and Shipping

This is the last module in the system. The warehousing and shipping module receives the completed work orders from the scheduling module, and issues a warehousing or a shipping order. Note that as products are shipped to the customers it gives feedback not only to the finished goods inventory control but also to the order control, as products are shipped to the customers.
3.2 Structured Analysis

Structured analysis is the use of data flow diagrams, a data dictionary and process narrative techniques to build a structured specification of a system. These tools can be used for developing the logic of a new system as well as for documenting existing systems.

3.2.1 The Data Flow Diagrams (DFD)

Data flow diagrams (DFD) are a network representation of a system showing all its components and the interfaces among them. When a system is too large, it can be partitioned into subsystems. If the subsystems are still large, sub-subsystems can be created, and so on, until the DFD of the subsystem can be clearly represented. The division of the system in these subsystems defines the concept of leveling or top-down analysis.

Leveled DFDs are made of a top, a bottom and a middle level. The top level is called context diagram. The context diagram delineates the domain of the study, with all the external interfaces. The middle level can, in fact, be several levels. These levels are formed by the decomposition of the system network in subsystems. The decomposition starts with the level 0 DFD, the highest level after the context diagram, and continues until processes that cannot be further decomposed are found. The bottom of the leveled DFD consists of these processes. Figure 15 shows how lower levels are found
by decomposing higher level processes.

The notation used on the construction of the DFDs is shown in Figure 16. The processes are represented by so called bubbles and are numbered according to the level in which they are represented. In order to relate the lower level processes to their parent, the process numbers include their parent number. For example, the process number 2-1-2 means that it is the second process in the level 2 DFD that portrays the process number 2-1. The process numbers are inside braces ({}).

The data flows are the representation of the interactions between two processes, between a process and a data store, or a process and an external. The direction of the arrow indicates the direction in which the data flows. Double arrows are used to represent data flows in both directions.

The name of the data flow is included between delimiters (< >) and more than one word names have the words separated by dashes (-). The name of the data flow is placed one character from the flow and can be either in the right or left position for vertical flows, and either top or bottom position for horizontal flows.

Data stores, more commonly known as files, are represented by two parallel lines. The name of the file is included between these two lines.

The connector is illustrated by a drop. These symbols are used to represent net input and output data flows from the DFD.
Fig. 15 - DFD - Top-Down Analysis Concept
Fig. 16 - Data Flow Diagram Notation
The last notation in Figure 16 is the external. This symbol is identified by a rectangle and serves to represent entities that are external to the system, e.g. users and other systems. Externals are found only in the context diagram.

3.2.2 The Data Dictionary (DD)

Data flow diagrams and the data dictionary are considered together. Without the DD, the DFDs are meaningless.

The data dictionary contains the definitions of the data flows. There must be a data dictionary entry for each unique data flow that appears anywhere in the DFD set.

In constructing the data dictionary the following notation is used:

- = means IS EQUIVALENT TO
- + means AND
- [] means EITHER - OR; i.e. select one of the options enclosed in the brackets
- {} means INTERACTIONS OF the component enclosed
- () means that the enclosed component is optional.

3.2.3 The Process Narratives

The bottom level of the leveled DFD consists of processes that cannot be decomposed further. These processes must be described to complete the specification of the system being designed.

There are different methods of describing the procedures
in these processes, one way being the use of Structured English.

Structured English is a specification language that makes use of a limited vocabulary and a limited syntax. The vocabulary used in structured English consists only of imperative English language verbs, terms defined in the data dictionary, and certain reserved words for logic formulation. The syntax of a Structured English statement is limited to simple declarative sentences (e.g.: Read FILE; Write REPORT; Delete RECORD), closed-end decision construct (e.g.: If - Then - Otherwise), closed-end repetition construct (e.g.: Go to; Go Back to; Call Process), or combinations of these.

The data flows entering the process must be read while those flows leaving the process will be sent to a file, a process, or an external.

To facilitate the identification of the internal commands for a closed-end decision or repetition, the structured English commands can be numbered in the same fashion as the topics in the chapters of this thesis (i.e. 1; 1.1; 1.2; 1.2.1; 1.2.2; and so on).

More details about structured analysis can be found in De Marco [17].
CHAPTER 4
SYSTEM DESIGN

4.1 Introduction

This chapter will describe the design of the proposed system in two phases. The first phase is the top design. The top design of the system explains all modules in the system and their interactions among themselves as well as with the external interfaces. The detailed operation of each process, the bottom design, is described later in this chapter.

Although this chapter will be sufficient for a general understanding of the approach used in the design of the proposed system and its operation procedures, details of the logic design are found in the data flows, data dictionary and process narratives in the Appendix A of this thesis.

4.2 The Top Design

4.2.1 External Interfaces

The context diagram of the proposed system shows four entities that interface with the system. The first external entity is the user, called WVDOH PERSONNEL. The users of the system will usually be the workers in the WVDOH Traffic Engineering Department and Central Sign Shop, in Charleston, W.V. The context diagram also indicates that information flows in both ways between the system and its users. The INVENT-PROD-DATA and RECORD-INFO data flows represent all the
information the users provide to the system and the REPORTS data flow contains all the outputs generated by the system.

The second entity, called DISTRICTS, represents all the traffic sign warehouses in the state of West Virginia that issue orders to the Central Sign Shop. The DISTRICT-ORDER data flow indicates that these orders (the physical document) are entered in the system.

The INVENTORY SYSTEM is the third external entity shown in the context diagram. This represents the inventory system described in Chapter Two of this thesis. The interactions between an inventory system and an inventory control and production planning system usually exist in both directions, where information is exchanged so that data redundancy is avoided. The reason for a one direction interface in the new system design is due to the WVDOH operation particularities. Because the WVDOH uses the same inventory system used by other government organizations in the state of West Virginia, this system is read only, and the only possible interface was the inquiry type. Because of this limitation, the proposed system will not be able to update the inventory system, but only read the on hand inventory levels in each district and the central sign shop (represented by the ON-HAND data flow) and the purchase orders information (represented by the PURCHASE-ORDER data flow).

The last external is the FORECASTING SYSTEM, being developed to forecast the demand for class A products, using
data from the inventory system (class A products are those defined by the ABC analysis). The FORECAST SYSTEM is similar to the demand forecast module in the framework described in the Chapter Three.

4.2.2 The Control Inventory Module

The level 0 DFD in the Appendix A shows the three major modules in the new system. The CONTROL INVENTORY module is responsible for controlling the district orders, the finished goods allocation to these orders, and the delivery procedure, as shown in its level 1 DFD. These procedures include all the transactions in the customer order control, finished goods inventory control, and shipping modules described in the framework discussed earlier in Chapter Three.

The INPUT ORDER sub-module is responsible for entering and approving the orders arriving from the districts. The ALLOCATE ORDER procedure allocates the finished goods inventory to the orders. The finished goods can be in the form of inventory on hand, production orders planned, purchase orders planned, scheduled receipts or MPS. MPS stands for master production schedule and it represents the units that have to be produced or purchased to satisfy demand, as we will see later. The CONTROL DELIVERY informs the user which items are available to be delivered and which have been delivered. In the ideal operation procedure, this module would update the inventory system when items are delivered, and the inventory
transference inside that system would be performed automatically. However, due to the practical limitations already discussed, the CONTROL DELIVERY module will only update the orders status and the inventory system will continue to be updated in the same manner as it is presently done (by entering a transfer form document).

4.2.3 The Plan Production Module

This module includes the remainder of the modules in the framework described in the literature review. In other words, the PLAN PRODUCTION module is responsible for the master production planning, the material requirements planning, the capacity planning, the scheduling, the purchasing and material control, the shop floor control, and warehousing procedures. In the plan production level 1 DFD the PLAN MPS, PLAN MATERIALS and SCHEDULE PRODUCTION sub-modules are illustrated.

4.2.3.1 The Plan MPS Sub-Module

The PLAN MPS includes the MANAGE DEMAND, MANAGE MPS and MANAGE CAPACITY procedures. The MANAGE DEMAND procedure calculates the net demand for the end products according to the forecasted demand as given by the forecasting system, the inventory level at each district, and the orders placed by the districts at the sign shop. The result of this calculation is the expected order quantity from each district, for each item
(only class A products) and for each period over the planning horizon.

The MANAGE MPS generates the master production schedule for each item and period in the planning horizon. Input are the net demand data, the inventory levels at the sign shop, and the scheduled receipts. MPS determines the number of units of a finished product that are required to meet demand, and if necessary, which items will be produced and which will be subcontracted. This procedure also generates MPS for lower class products, according to necessity, as will be explained later.

The MANAGE CAPACITY corresponds to the capacity planning module in the scheme shown in Figure 10 (in Chapter Three). This procedure uses bills of capacity to test the capacity required to meet the production plan against the capacity available for each period in the planning horizon. The objective is to help the user find a feasible production plan. Note that this system plans the capacity before performing the material requirements planning. This is due to the simplicity of the products' bill of materials. The bill of materials for traffic signs are structured in only one level, which enables the capacity planning to be based only on the MPS records.

When insufficient capacity exists, it is necessary to either subcontract or delay the production of some items to the next period. When excess capacity is found, it can be used for the production of some items which are required later. The
results of this procedure are production orders (PO) and purchase requirements (PR) over the planning horizon.

4.2.3.2 The Plan Materials Sub-Module

This sub-module manages the raw materials requirements planning, the purchasing of the raw materials and subcontracted products and their receipt.

The MANAGE PR AND RM-PR procedure generates the raw material purchase requirements (RM-PR) by utilizing the MRP calculations over the planning horizon (based on the production orders planned), the bill of materials records, the inventory level of raw materials at the sign shop, and the scheduled receipts. The purchase requirements for subcontracted finished goods are generated at the MANAGE CAPACITY procedure, however, this module provides the user the flexibility of generating any new PR records he deems necessary.

The PURCHASE PR AND RM-PR procedure informs the user of the necessary purchase requirements and updates the system when the items are purchased. The ideal interface with the inventory system would update the purchase requirement records with the purchase order number at the time the items are purchased, without user interface, however, the inability to access the inventory system created the need for this interface. The user must inform the system of the purchase requirements which were purchased and the number of the
purchase orders that were placed.

The RECEIVE PURCH ORDER periodically checks the inventory system, more specifically the purchase orders data, to update the purchase requirements records with the items received from the outside suppliers.

### 4.2.3.3 The Control Production Sub-Module

This last sub-module schedules production, performs the shop floor control and controls the warehousing activities, as shown in the control production level 2 DFD (Figure 38 in the Appendix A).

The SCHEDULE PRODUCTION procedure sorts the planned production orders according to the priority rules and schedules them using a finite loading technique.

The COMPLETE PO is used to update the system when production orders (PO) are completed. This is another situation where an interface with the inventory system, which would inform the system that finished goods were sent to inventory, would be useful. The practical limitations already discussed will again require this information to be manually entered in the inventory system.

The END WEEK procedure lets the user define the start of a new week, and updates the system on which PO's were scheduled and not completed. This seems to be a redundancy of information since the COMPLETE PO procedure updates the system when the PO is completed. However, the END WEEK procedure is
necessary to give the user the flexibility of scheduling the upcoming week before the current week is over.

4.2.4 The Manage Data Module

The MANAGE DATA is the last module in the system. This module gives the user more flexibility in dealing with unexpected situations. When the orders coming from the districts are entered into the system, the orders may be satisfied by inventory on hand, by production orders, by purchase requirements, and by MPS. The REPORT ORDER procedure informs the user of the status of the order records in terms of this allocation. The objective of this is to provide the user with the information for making modifications in these records as needed. For example, if the units of a specific order are presently promised to be satisfied by inventory on hand, and for any reason, the user decides to change this status (making it promised by a production order) the MANAGE PROMISED VALUES enables the user to make these changes in such a way that the integrity of the data is maintained.

4.3 The Bottom Design

This section describes each process in the data flow diagram and the internal interactions of the system. Since the process narrative is provided for each process (in the Appendix A), the descriptions here will serve only as a general explanation of the purpose and operational logic of
these processes. A brief explanation of the data flows that are associated with the processes is also provided when necessary.

4.3.1 The Control Inventory Module

4.3.1.1 The Input Order Sub-Module

As discussed in the top design, the INPUT ORDER sub-module allows orders to be entered into the system and be approved. The ENTER ORDER is the first process in this sub-module. It permits the user to input the DISTRICT-ORDER (the physical document) and store this data in the ORDERS TO APPROVE file. The data stored, called ORDER, contains the date of the order, the order number, the district which issued the order, the items ordered, and the order quantity for each item.

After entering the orders, the GENERATE ORDER REPORT can be run. This process generates a report which helps the user identify orders that are not in accordance with the operational standards. When the user enters the DISTRICT-TO-REPORT, the process checks the total ordered for each item in the ORDER records against the ON HAND level at the district to determine if the district is ordering items for which it has an adequate supply of on-hand inventory. The same logic is used to check the open orders. For each item ordered, the process reads the ORDER-REC and ORDER-APPROVED records to determine if the district is ordering items that it has
accepted, but only until starting the execution of the allocation procedure. Orders received after the execution of this module will be considered in the next month. This will assure that the production plan will not need to be changed due to new orders received.

The objective of allocating the inventory to orders is to provide a more effective and fair distribution system. By establishing the priority for allocation based on the district delivery date, the system tries to allocate any available inventory on hand to those orders that will be delivered first, since those orders that will not be delivered immediately can wait for new receipts. The SORT ORDER process is responsible for sorting the APPROVED-ORDER records according to the DELIVERY-REC, which contains the delivery dates for each district. After sorting the orders for allocation, the ALLOC-PRIORITY-LIST is reported to the user, who can make changes in this sequence by entering the CHANGED-ALLOC-PRIORITY data.

The next step is the CHECK FORECAST. This process determines if the quantity being ordered is higher than the expected order for that period. If this is the case, the ABOVE-FORECAST-ORDER is reported to the user who has two options. The first is to let the order remain as it is and the second is to request that the EXCESS ordered be separated in the ORDER-BROKEN record. The reason for separating the excess ordered in the ORDER-BROKEN is to ensure that the units in
inventory will be distributed fairly. If a large order for a particular item is received and approved, and no action is taken, this large order may take all the available inventory, thereby affecting the other districts orders. To avoid that, the system will separate the excess being ordered and identify it for allocation to a new MPS at a later time. To generate this new MPS, the CHECK FORECAST process calls the GENERATE REQ MPS process, sending the MPS-REQ record indicating the item and the required quantity. After execution, the GENERATE REQ MPS then returns the MPS-NUMBER of the MPS generated for reference in allocation.

When all the orders are checked against the forecast, the GENERATE STD SIGN NET DEMAND in the MANAGE NET DEMAND sub-module can be executed. Figure 17 shows the framework the system follows when executing the allocation of the orders. Note that this procedure involves different processes not only in the ALLOCATE ORDER sub-module but also in other sub-modules.

The allocation of the orders can be executed only after generating the MPS for the class A products. However, to generate the MPS, the net demand for these items must first be calculated. These processes will be described at the appropriate time.

The ALLOCATE BROKEN ORDER process follows the generation of the MPS for all class A products. This process allocates the excess ordered, as detected by the CHECK FORECAST, to the
Fig. 17 - Allocate Order Procedure Framework
MPS generated by GENERATE REQ MPS. This will ensure that the excess being ordered will not consume available inventory, but will wait until the production (or purchase) of these items is planned.

The processes responsible for allocating the inventory to orders is called GENERATE ORDER REC. The ORDER-REC is the record that contains information for each order item regarding its allocation. In other words, the ORDER-REC for an ordered item contains the units promised by inventory on hand, the units promised by production orders, the units promised by purchase requirements, the units promised by MPS, and the units not yet promised. The MPS-REC, PO-REC, PR-REC and ON-HAND-REC are the records that show the available units to promise in the MPS, PO, PR and ON-HAND respectively. The record that contains the number of units for a MPS that are promised to an order is the MPS-ORDER-REC. Figure 18 shows the relationship among ORDER-REC, MPS-REC and MPS-ORDER-REC records. The same relationship exists among the records described below:

ORDER-REC; PO-REC; and PO-ORDER-REC
ORDER-REC; PR-REC; and PR-ORDER-REC
ORDER-REC; ON-HAND-REC; ORDER-ON-HAND-ORDER-REC

The allocation procedure reads the number of units in each ORDER-REC that are not yet promised and tries to find available inventory from either on hand inventory, production orders, purchase requirements, and MPS, in this order, for
**ORDER-REC**

<table>
<thead>
<tr>
<th>ORDER-REC-NUMBER</th>
<th>90-01-77</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER-REC-ITEM</td>
<td>02</td>
</tr>
<tr>
<td>ORDER-REC-TYPE</td>
<td>021</td>
</tr>
<tr>
<td>ORDER-REC-SUBCODE</td>
<td>010101</td>
</tr>
<tr>
<td>ORDER-REC-QUANT</td>
<td>20</td>
</tr>
<tr>
<td>ORDER-REC-PROMISED-BY-MPS</td>
<td>12</td>
</tr>
<tr>
<td>UNITS-STILL-NOT-PROMISED</td>
<td>8</td>
</tr>
</tbody>
</table>

**MPS-REC**

<table>
<thead>
<tr>
<th>MPS-REC-NUMBER</th>
<th>90-004</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPS-REC-TYPE</td>
<td>021</td>
</tr>
<tr>
<td>MPS-REC-SUBCODE</td>
<td>010101</td>
</tr>
<tr>
<td>MPS-REC-UNITS</td>
<td>100</td>
</tr>
<tr>
<td>MPS-AVAIL-TO-PROMISE</td>
<td>88</td>
</tr>
</tbody>
</table>

**MPS-ORDER-REC**

<table>
<thead>
<tr>
<th>MPS-ORDER-REC-ORDER</th>
<th>90-01-77</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPS-ORDER-REC-ITEM</td>
<td>02</td>
</tr>
<tr>
<td>MPS-ORDER-REC-MPS</td>
<td>90-004</td>
</tr>
<tr>
<td>MPS-ORDER-REC-UNITS</td>
<td>12</td>
</tr>
</tbody>
</table>

**Fig. 18** - The ORDER-REC, MPS-REC and MPS-ORDER-REC Relationship
allocate to this order. When allocating the class B product orders, if the order cannot be totally satisfied by the available inventory in one of these forms, the GENERATE REQ MPS process is called to generate a new MPS. The MPS-REQ record informs the GENERATE REQ MPS process as to the product and the units required for the generation of the new MPS. This will ensure that some form of inventory will be available to satisfy the orders. The result of this procedure is all ORDER-REC records allocated to ON-HAND, PO, PR and MPS records. Note that the generation of MPS during the allocation procedure is only required for orders requiring class B products, since class A products will have all the necessary MPS generated previously, as will be explained later.

4.3.1.3 The Control Delivery Sub-Module

This module includes two processes. The first is the GENERATE DELIVERY LIST. The objective of this process is to inform the user of the on hand inventory which is promised to the districts. The user has the capability of confirming the district numbers that are in the DELIVERY-REC record or enter another DELIVERY-DISTRICTS. After knowing for which districts to generate the DELIVERY-LIST, the process then reads the ON-HAND-ORDER-REC records to report the inventory on hand that are promised to these districts.

The second process, called UPDATE ORDER DELIVERED, informs the system of the ORDER-DELIVERED. This data is then
used to update the ORDER-REC, ON-HAND-ORDER-REC and ON-HAND-REC records.

4.3.2 The Plan Production Module

4.3.2.1 The Plan MPS Sub-Module

The PLAN MPS sub-module is divided into three major procedures. The first is the MANAGE DEMAND, which is responsible for computing the NET-DEMAND of class A end products. The low frequency of use for the lower class products makes it difficult and unproductive to accurately forecast demand, and therefore demand will not be forecasted for these products.

There is a major difference between the DEMAND-FORECAST calculated by the FORECAST SYSTEM and the NET-DEMAND calculated here. The DEMAND-FORECAST is the predicted demand for the district. In other words, the DEMAND-FORECAST predicts the number of signs that will be needed for installation in the field, in a certain district. On the other hand, the NET-DEMAND is the predicted orders that the districts will be issuing to the central sign shop. The procedure for calculating the NET-DEMAND for a standard sign is to check the DEMAND-FORECAST, the inventory ON-HAND level and the scheduled receipts, to determine when the reorder point will be reached. The net demand calculation checks not only the period in which the order is expected at the sign shop, but also the number of units in the predicted order. This last calculation takes into
consideration that the district will be ordering the number of units necessary to increase its inventory to the highest allowable level. The process that performs these calculations is called GENERATE STD SIGN NET DEMAND.

The GENERATE SPECIAL SIGN NET DEMAND process converts the DEMAND-FORECAST into NET-DEMAND for special signs. These products are one of a kind and for that reason do not have inventory levels. The forecast for these signs is made by its code, which represents the size of the sign. The DEMAND-FORECAST here indicates the number of special signs of a particular size, that the district will be requiring for installation in the field. The NET-DEMAND in this case is equivalent to the DEMAND-FORECAST.

The last process in this sub-module is the CHANGE NET DEMAND. This process exists to give the user the flexibility of changing the NET-DEMAND as calculated by a PREDICTED-DEMAND, e.g. informing the system about a demand that the user believes will occur.

The next procedure in this sub-module is the MANAGE MPS. This procedure is responsible for generating the master plan schedule for each end item and for each period in the planning horizon. The first process in this procedure is the GENERATE STD MPS, which generates MPS for all standard signs. The logic on the MPS computation followed by this process is similar to that reviewed in Chapter Three, and for that reason will not be detailed here. The planning horizon used in this proposed
system is four months, and the period interval used is one month. The reason for using a four month planning horizon involves particularities in the WVDOH operations. Three factors influenced this decision: The material lead times, the production lead time and forecasting accuracy. Discussions with the WVDOH management revealed that the longest lead time for material supply was lower than two months. The production lead time at the sign shop can not be precisely predicted since it is affected by a large number of factors, however, if no queue exists, the production lead time for most signs is no longer than one day. The use of a four month planning horizon is reasonable if we consider a safety factor for dealing with delays in resources and material receipts and wish to keep the required accuracy of the forecast data.

The GENERATE SPECIAL MPS is similar in some ways to the GENERATE STD MPS, however important differences are found between them. The NET-DEMAND for a special sign contains the number of signs of a particular size that are expected to be ordered in a given month. This data is used for planning the production and the necessary materials required for producing these signs such that, when the order is received, the material and the capacity for producing the signs will be available. Since the special and interstate signs are one of a kind, no inventory on hand is available and the lot size in the MPS is one unit. For this reason, instead of computing the number of units required to meet demand, this process, in
fact, calculates the number of MPS that have to be generated. For example, if the gross requirements for month 1 is 10 signs and the available to promise at that month (in MPS or production orders) is 4, then 6 MPS will be required to complete the gross requirements. One observation about the special and interstate signs MPS size will be made here. When ordering special signs, the district may require more than one sign. One common example is the order for similar signs that will be installed on both sides of a road. When the allocation for special and interstate signs is made, if more than one unit is required, the process ALLOCATE MPS searches for similar MPS records available which can be grouped to form a single record, where the number of units is the number of units being ordered. If there are no similar MPS records available for grouping, a new MPS is then generated by calling the GENERATE REQ MPS, and at this time, the size of the MPS generated will be equivalent to the number of units being ordered. A similar procedure is also used in the ALLOCATE PRODUCTION ORDER process.

The third process in this procedure is the GENERATE USER MPS. The objective of this process is to give the user the ability to generate any MPS he deems necessary. One example of such a situation is when the user wants to plan the material and capacity for a special order he knows will be received soon. In this case, he can create the MPS before receiving the order and the MPS generated will then be included in the
production plan. When the order is finally received, the material and capacity will be available for production. This gives the management a tool for expediting special orders without making changes in the production plan.

The DELETE MPS enables the user to delete a MPS that is not promised to any order and that the user feels will not be required. In this way, the user can reduce the purchase of unnecessary raw material and reduce the capacity load required by the production plan. Although it has advantages, this module must be used wisely, since one can delete a MPS record not yet allocated but that will be required in the future.

The last process in this procedure is the GENERATE REQ MPS. As already described, this process generates the MPS required by the ALLOCATE ORDER sub-module, according to the product and number of units required.

The result of these processes (excluding the DELETE MPS) are MPS-REC records that are updated in the MPS RECORDS file. The planning month in these records is the month for which they are originally required. The production lead time used for planning the MPS is one month. In other words, the MPS is produced in the month it is required. Both MPS generated by the GENERATE REQ MPS and by the GENERATE USER MPS processes will be restricted by their raw material lead time in establishing the monthly plan. When a new MPS is required, the process reads the longest raw material lead time and plans the generated MPS in the month equivalent to the present month.
added to the longest raw material lead time. For example, if the user requires a MPS for a product in month 1, and if the longest lead time of the raw materials required for production of this product is 2 months, the MPS generated will be planned in month 3. This will assure that enough time will be given for purchasing the raw material for this new MPS and that the raw material available for the production of the other MPS will not be consumed. If the product in the MPS required is a subcontracted item, the supplier lead time is used.

The last procedure in the PLAN MPS sub-module is the MANAGE CAPACITY. Figure 19 shows the framework followed by this procedure. Note that the MANAGE CAPACITY works in an interactive procedure to define the capacity plan for each of the periods in the planning horizon. The procedure starts with the process IDENTIFY AVAILABLE CAPACITY, which is used to define the number of working DAYS-TO-PLAN and the hours available for each resource for each month of the planning horizon. The results are RESOURCE-PLAN-REC records for each resource indicating available hours and efficiency rates.

The EXECUTE WORK LOAD is responsible for calculating the capacity required by the production plan. The EXECUTE WORK LOAD uses bills of capacity for computing the capacity requirements for all production orders and MPS planned in each month of the planning horizon. The result of this process is that the work load on each resource is updated in the RESOURCE-PLAN-REC records.
Fig. 19 - The Manage Capacity Procedure Framework
The REPORT PLAN WORK LOAD is executed after computing the capacity requirements. This process generates the PLAN-WORK-LOAD-REPORT, which informs the user of the overloaded and underloaded resources.

As the user analyzes the report, he may decide to change the available capacity. This can be done by re-running the process IDENTIFY AVAILABLE CAPACITY from the start. If the user decides to change the production plan rather than capacity, there are two options from which to choose. The first is to make changes in the MPS and PO due dates or in the MPS destination (e.g. if the MPS will be produced or subcontracted). The process that enables the user to make these changes is the CHANGE MPS AND PO. To deal with overloaded resources, the user can either postpone until the next month the MPS and PO that use that resource or subcontract some MPS. Under capacity problems can be solved by expediting some PO and MPS. However, the expedition is limited by the material lead time for the manufactured products, and by the supplier lead time for subcontracted items. Because of this limitation, the MPS-PO-REPORT generated in this process indicates the maximum possible expedition for the MPS and PO. After making the changes, the user sends the new production plan back to the EXECUTE THE WORK LOAD process to determine the results of the changes. The second option for changing the production plan is to let the process DELETE MPS FROM PLAN find a feasible plan for the month by changing the destiny of
the MPS from production to purchase. This process selects those MPS that can be subcontracted, and sorts them by priority for purchasing. This priority for purchasing is a number associated with every item which can be either subcontracted or manufactured. This priority can be stated as: "If some products have to be subcontracted, which ones are preferred for subcontracting and which ones are preferred for manufacturing?". This preference can be determined either by a make or buy analysis or management decision. After sorting the MPS by purchase priority, the process changes the destination of these MPS and evaluates the impact of this change on the capacity plan until a feasible plan is determined. The new plan is then submitted to the REPORT PLAN WORK LOAD. Note that this process is not helpful in solving under capacity problems.

This interactive procedure continues until the capacity plan for the four periods in the planning horizon are approved. When the user finally approves the capacity plan, those MPS-REC to be manufactured are translated into PO-REC and those to be subcontracted become PR-REC.

4.3.2.2 The Plan Materials Sub-Module

The first procedure in this sub-module is the MANAGE PR AND RM-PR, which generates the RM-PR for each raw material, performing the MRP calculations based on the capacity plan. This process reads the production orders planned for each
period in the planning horizon and computes the total material gross requirements for each raw material, using the BOM-REC records. The BOM-REC for a finished product contains the raw material requirements for manufacturing the product.

After computing the gross requirements for each raw material, the process checks the inventory ON-HAND at the sign shop and the scheduled receipts to calculate the RM-PR that will be required. The scheduled receipts represent those RM-PR which have already been ordered. Note that the planning month associated with the RM-PR is the month in which it is required. Later, when generating the PR-AND-RM-PR-REPORT, the system will inform the user as to when the RM-PR and the PR-REC must be purchased. After the RM-PR records are generated, they are stored in the RAW MATERIALS PR RECORDS for future use.

Both processes GENERATE USER PR and GENERATE USER RM-PR exist to give the user the flexibility of generating finished goods purchase requirements and raw materials purchase requirements whenever deemed necessary. One situation that requires this generation is when, for any reason, the user decides to place a purchase order that was not required. In this case the user can generate a PR-REC (if the product purchased is a finished good) or a RM-PR (if the product is a raw material) and update it with the purchase order number, to inform the system that an additional scheduled receipt exists.
The last process is the DELETE PR which enables the user to delete those PR-REC that are deemed not necessary. In deleting the PR-REC of finished goods, the user will be reducing the available inventory of that item. If used wisely, this process can reduce unnecessary inventory levels. However, care must be exercised here, since the deletion of the PR-REC that will be necessary in the future can cause back orders.

The second procedure in the PLAN MATERIALS sub-module is responsible for updating the system with the purchase requirements that have been received, called PURCHASE PR AND RM-PR.

The first process in this procedure is the GENERATE PR AND RM-PR REPORT. This process is used to report all required finished goods and raw materials, as well as the quantity and the month they need to be purchased. The report for finished goods purchase requirements also shows the percentage of the PR-REC that is already promised to orders. This informs the user as to the portion of the PR-REC which will be delivered as soon the products are received.

When the PR-REC or RM-PR records are purchased, the user must inform the system of the number of the purchase order that is associated with these records. This is done through the PURCHASE SIGNS AND MATERIALS and PURCHASE RAW MATERIALS processes. The purchase order number is then stored for future use in tracking the receipt of the purchased items, as is explained below.
The last procedure in this sub-module is the RECEIVE PURCH ORDER. The processes in this procedure reads the PURCHASE-ORDER records in the INVENTORY SYSTEM to track the new receipts of purchased items. Since it is not possible to know the exact time that the receipts are updated in the inventory system, this procedure keeps the information on the number of units received for each PR-REC and RM-PR purchased. Periodically, the processes reads the PURCHASE-ORDER records and compares the number of units received according to these records with the number of units received according to the PR-REC and RM-PR records. If the first number is higher, a new receipt has been updated in the inventory system and additionally, the RECEIVE SIGNS AND MATERIALS and the RECEIVE RAW MATERIALS processes update their records with the new value. Besides that, the RECEIVE SIGNS AND MATERIALS process also updates the units promised by inventory on hand in the ORDER-REC records that had units promised by the PR-REC record received. If a partial shipment is received, the distribution of the units received is based on the delivery priority. In other words, the orders that will be delivered next have the highest priority for the units received. To have the most current values for units promised by inventory on hand, the user should always run this process before running the GENERATE DELIVERY LIST.
4.3.2.3 The Control Production Sub-Module

The CONTROL PRODUCTION is the last sub-module in the PLAN PRODUCTION MODULE. The activities involved here are those related to the shop floor scheduling and control, and warehousing.

The scheduling of the weekly production involves eight processes that follow the framework pictured in Figure 20. The IDENTIFY AVAILABLE SCHD CAPACITY is the first in this framework, and similar to the IDENTIFY AVAILABLE PLAN CAPACITY, this process interacts with the user to determine the capacity available in each resource for the week to be scheduled. When the DAYS-IN-WEEK is entered, this process computes the work hours for each resource and reports the AVAILABLE-SCHD-CAPACITY back to the user. At this point, there are two options from which the user must choose. The first is to confirm the available capacity shown for each resource, which is based on the RESOURCE-PLAN-REC records. When the capacity plan is finished, the capacity available for each resource is stored in the RESOURCE PLAN RECS file. When scheduling the production, the procedure checks the current month and accesses the RESOURCE-PLAN-REC for each resource. In that way, the procedure will know the resource availability for that specific month. The second option is to change the available capacity. In this case, the procedure reads the RESOURCE-SCHD-CHANGE data and recalculates the available capacity. The last step in this process is to store this
Fig. 20 - The Schedule Production Framework
available capacity in the RESOURCE SCHD RECS file for further use.

The second process in the framework is the SORT PO TO SCHEDULE. This process establishes priority for the production orders. The PO-TO-SCHEDULE are temporary records created during the scheduling procedure. These records contain information about the PO-REC that are planned in the current month and waiting to be scheduled. The PO-TO-SCHEDULE records will store information about the status of the PO-REC during the scheduling procedure. All processes (except the first one) update these records as they advance through them. In the SORT PO TO SCHEDULE process for example, the PO-TO-SCHEDULE receives the priority for scheduling, which is used to sort the production orders. The priority rules defined to sequence the PO-REC for production in this process were established based on the WVDOH management's desire to improve the delivery efficiency, reduce the inventory levels at the sign shop, and also reduce the time the orders take to be completed. To calculate these priorities the process first computes the following ratios:

\[
\text{Ratio I} = \frac{\text{Units in PO Promised to Next Delivery}}{\text{Total Units in PO}}
\]

\[
\text{Ratio II} = \frac{\text{Units in PO Promised to Second Delivery}}{\text{Total Units in PO}}
\]

\[
\text{Ratio III} = \frac{\text{Units in PO Promised to Third Delivery}}{\text{Total Units in PO}}
\]
Ratio IV = \frac{\text{Units in PO Promised to Fourth Delivery}}{\text{Total Units in PO}}

Since every district receives a shipment once every four deliveries, it is not necessary to go beyond that.

After calculating these ratios, the process sorts the production orders using Ratio I, II, III, IV and the PO-REC original month, in such a way that the highest priority will be given to those production orders needed for the next delivery, and then those needed for the second delivery, the third delivery, the fourth delivery, and finally to those production orders that are late. The use of the PO due date will be used to break the ties. Although standard signs PO will rarely show the same value for any of the four ratios, that is not true with the special and interstate signs PO. Since the PO for these signs are usually from only one order, the value for the ratios I, II, III and IV will usually be either zero or one for these production orders. In this case, the use of the PO due date will benefit those PO that are late. Special and interstate sign production orders will be selected for scheduling only if already promised to any ORDER-REC.

Note that the process will sort all the production orders together, regardless of the product to be manufactured. However, when assigning priority to those orders, the process separates the PO in three different priority lists, according to the category of the sign in the production order (standard,
special, and interstate lists).

The result of the use of these rules in establishing the priorities for scheduling production orders is a more efficient delivery system and a lower level of inventory at the central sign shop.

Since the districts which will receive delivery change every week, those orders that show a high priority in one week, but for some reason are not completed, will not necessarily have a high priority in the following week's schedule.

After defining the priority for all PO in the production plan, the process generates the SCHD-PRIORITY-LIST. The user will now have an opportunity to change the priorities for scheduling.

With the PO-TO-SCHEDULE updated with the priority number, the next step is to check the availability of the raw materials required for production. The CHECK RM AVAILABILITY uses the current ON-HAND inventory level of raw materials at the sign shop and generates the temporary record RM-AVAILABLE. This is used to determine which production orders can be completed with the available raw material. It also avoids scheduling PO-REC records that do not have the necessary raw material available for production. The process begins by storing the current ON-HAND levels of each raw material item in the RM-AVAILABLE records. These records will be used to represent the projected raw material inventory level for the
production time. Next, the PO-TO-SCHEDULE records are selected by the process, from the highest to the lowest priority, and the use of the raw material by these PO is simulated by subtracting the quantity required from the RM-AVAILABLE for each item. When the raw material for a determined PO is not available, the PO-TO-SCHEDULE is deleted from the file and the PO-REC is eliminated as a candidate for scheduling in the current week. The result of this process is a PO-TO-SCHEDULE not constrained by lack of raw materials.

The next process in the scheduling procedure is the LOAD PO AT NON-SHARED RESOURCES process. Non-shared resources are those resources utilized by only one category of signs, e.g. the screening machine, which is used only by standard signs. The motivation for determining the load at the non-shared resources is to delete the PO-TO-SCHEDULE that will be constrained by capacity at these resources before attempting to schedule them at the shared resources. The finite loading will use bills of capacity to simulate the work load at the resources until capacity is fully utilized. The process selects the PO-TO-SCHEDULE with the highest priority and computes the required capacity by the PO at each resource, according to the information given in the PROD-ROUTE records. For each resource, the available capacity is checked. If there is still capacity available in all resources used by the production order, the process updates available capacity for each resource by subtracting the required capacity computed
for the PO. If capacity is not available in one of the resources used by the production order, the PO-REC is no longer a candidate for being scheduled during that week. A new PO-TO-SCHEDULE is then selected by the process and this procedure continues until all PO-TO-SCHEDULE are selected. The result of this procedure is a first draft of the scheduling list, which must still be checked at the shared resources.

Discussions with the WVDOH management determined that the scheduling list should contain three different lists, one for each category of sign. The reason for this is that usually, the fabrication of these three categories of signs will be independent and may happen at the same time. In other words, normally the sign shop will be producing standard, special and interstate signs at the same time. Although these three categories of sign have to share some resources (for example the sheeting machine and operator), the high degree of flexibility in assigning jobs to human resources will minimize any problems caused by queues at these shared resources, (the operators waiting for incoming work in process from a shared resource can be shifted to another task while the resource is busy).

After generating these three lists in the LOAD PO AT THE NON-SHARED RESOURCES it is necessary to check the work load at the shared resources. The determination of whether the work load exceeds the available capacity at the shared resources is the responsibility of the LOAD PO AT SHARED RESOURCES. If the
required capacity is lower than the available capacity, then the shared resources are not overloaded, and the scheduling lists generated by loading the production orders at the non-shared resources can be approved. However, if any shared resource is overloaded, some production orders will have to be deleted from these lists. The process DISCHARGE PO FROM LIST is responsible for interacting with the user, to read the PO-DISCHARGE data, which represents the production orders the user wants to eliminate from the scheduling list.

As already mentioned, the production of signs does not have long lead times (some signs can be produced in less than one hour) and for that reason it would not be feasible to track the work in process inventory at the resources. Since the system will not monitor the work in process inventory levels at the resources, the decision of selecting the production orders to delete from the three scheduling lists to find a feasible schedule cannot be made without user interaction. Therefore, instead of deciding which PO not to schedule, when overloaded shared resources are found, the system leaves this task to the user who has the information necessary to make the correct decision.

Another important observation can be made here. The technique used to load the production orders at both non-shared and shared resources does not consider the time the production orders are loaded at the resources and therefore assumes that no resource will be idle while waiting for
incoming work in process. There are basically two reasons for this simplification. The first is the unknown sequence of the production orders at the shared resource. The second is the flexibility in assigning jobs to workers. To accurately define the time each PO will reach the resources, the sequence of the PO at all resources must be known in advance. Although the assumption of no idleness is not likely to be true, the consequences of making such assumption does not seriously affect the schedule. When the system schedules the production orders assuming no idleness, it is probably scheduling more PO than the sign shop will be able to complete during the week, however, having some PO not completed at the end of the week only means that those production orders not completed will have to be re-scheduled.

After deleting some PO, the new lists are checked at the LOAD PO AT SHARED RESOURCES. Note that this procedure can be repeated as many times as necessary, working as a "what if" analysis tool for determining the schedule that best fits the user's needs. To reconsider a production order previously discharged, the user can enter the PO-RECONSIDERED data and that discharged PO will be re-inserted in the scheduling list.

When the overload is resolved, the scheduling lists can then be approved and the SCHD-LIST report can be generated by the process APPROVE SCHEDULE. However, before reporting the scheduling sequence lists, the process must first analyze the product family of each product in those PO and re-sort the
production orders in so that production orders from the same family will be adjacent. Products from the same family are those that require the same set up at the sheeting machine. This procedure is utilized to reduce the set up times at that machine. After generating a new sequence, this process uses the PO-TO-SCHEDULE in the lists to update the status of the associated PO-REC to "scheduled". Finally, the SCHD-LIST is reported to the user. Other information given in the standard signs scheduling report is the percentage of the production order that is already promised to ORDER-REC records. This information allows the sign shop to make only those units that are promised to orders, if that is desired.

The shop floor control is also the responsibility of this sub-module. To track the units produced, the system uses the process COMPLETE PO to interact with the user, who must inform the system as to which production orders were completed and the number of units produced. This process then updates the ORDER-REC and INVENTORY-REC, and deletes the PO-REC.

The END-WEEK process gives the user the ability to inform the system as to the production orders that are scheduled and expected to be completed. The reason for collecting this information is to allow the user to prepare the schedule for the upcoming week before the end of the current week. The process first informs the user concerning the PO scheduled and not completed, by generating the NOT-EXECUTED-REPORT. The user can enters the PO-TO-KEEP-SCHEDULED data, indicating which PO
are not to be considered in the next schedule since they are expected to be completed before the next week starts. This process must be run each time the scheduling procedure is executed.

The DELETE PO is the last process in the CONTROL PRODUCTION sub-module. This process allows the user to delete the PO-REC that are not promised to any ORDER-REC and are deemed not necessary. As with the other DELETE processes, this one also must be used wisely to avoid deleting PO-REC that are necessary.

4.3.3 The Manage Data Module

The first process in this module enables the user to extract a report with the ORDER-REC records information. One reason for generating this report is to check the UNITS-REQ-ADJUSTMENT data of the ORDER-REC. The units requiring adjustment is the information that stores anomalies in the allocation of the order. More specifically, when the units allocated in the order are greater than the total units in the order, this difference is shown in the UNITS-REQ-ADJUSTMENT of the ORDER-REC records. A value greater than zero in this data means that the ORDER-REC requires adjustment. This difference between the number of units allocated from the total number of units can occur when the user makes a mistake when changing the allocation using the MANAGE PROMISED VALUES sub-module. For example, suppose there is an ORDER-REC record with 20
units of a specific sign, and 10 units of these 20 are promised by MPS and 10 promised by PO. If the user designates 5 units of this order as promised by ON-HAND and does not reduce 5 from the promised by MPS or PO, the UNITS-REQ-ADJUSTMENT will be 5 units. The ability to make these changes is necessary due to the flexibility the system gives the user to adapt to uncommon situations. Another example requiring user's intervention involves the original ORDER-REC described above. Since we have 10 of 20 units in the order promised by MPS-REC and the other 10 promised by PO-REC records, we don't have any units available to be delivered. If, for any reason, the sign shop decides to deliver 5 units of the order, the system will realize that an anomaly occurred and it will store 5 in the UNITS-REQ-ADJUSTMENT in the ON-HAND-REC, which means that inventory not available to be delivered was delivered. To deal with these situations, the user must change the values promised to the ORDER-REC by ON-HAND-REC, PO-REC, PR-REC and MPS-REC records by using the CHANGE PROMISED BY ON-HAND, CHANGE PROMISED BY PO, CHANGE PROMISED BY PR and CHANGE PROMISED by MPS processes respectively. In this example, the user could change the promised by MPS in the ORDER-REC for 5 units and add the 5 units promised by ON-HAND to correct the problem. Note that not only the ORDER-REC but also the ON-HAND-REC, PO-REC, PR-REC and MPS-REC are affected by these processes. These processes exist to give flexibility to the user while maintaining the integrity of the data. However the
extensive use of these processes could reduce the benefits of the proposed system. The user should be aware that these processes are available for dealing with unusual situations only.

The last process in this module is called TRANSFER INFO. This process is used to transfer information from a PO-REC to MPS-REC and from a PR-REC to MPS-REC. This transfer occurs when the user decides to reconsider producing a product that was planned for purchase, or vice-versa. In this case all the promised values are transferred to the MPS, which will have to be planned again. When interacting with the process, the user tells the system which destiny (production or purchase) he wants to give to the MPS.

4.4 Summary

As described in this chapter, the proposed system contains three major sub-modules. The sub-module that controls the inventory is responsible for managing the allocation of inventory to the orders issued by the districts. By allocating the inventory to the orders the system will ensure that an efficient and fair distribution system will be applied. It will also improve the efficiency of the production system as priorities for scheduling are established based on this allocation.

The second sub-module will plan and control the production at the sign shop as well as plan the materials
requirements. The first step is to determine the master production schedule required to satisfy demand over the planning horizon. The required capacity is then checked against the available capacity, to determine which parts will be subcontracted and which will be produced. For those items that will be subcontracted, the system will track their purchase as well as their receipt. For those items to be produced at the sign shop, the system will plan their materials requirements and later schedule the production orders based on priorities.

The last sub-module will enable changes in the inventory allocation to the orders, which will give flexibility to the user to deal with unusual situations.
5.1 Summary of Research

This thesis described the design of an integrated production and inventory control system for the state of West Virginia traffic sign shop. Although the system was designed to satisfy the specific requirements and limitations of the user, the concepts can be applied to any similar production system. The basic framework used in production planning systems was followed and adapted to the environment, i.e. product structures, demand behavior, inventory policy, work force flexibility, interface capabilities, production and material lead times and delivery policy. This design is intended to provide WVDOH management with a simple, easy to understand tool. The participation of the user in this design, through diverse walkthrough meetings, assured that the logic developed was in accordance with the WVDOH requirements and aspirations. The expected results of the system implementation are described below:

1. The control of the orders received at the sign shop will make it possible to identify orders that are not necessary or in accordance with standard procedures. This will reduce unnecessary inventory at the districts and unnecessary work loads at the sign shop.
2. The allocation of the orders will enable the sign shop to provide a more efficient and fair distribution of the finished goods inventory at the sign shop.

3. The production planning, based on forecast data, will reduce order lead times, especially those of the class A products.

4. The capacity planning over a planning horizon will highlight problems such as overload or underload and permit a more efficient use of the production resources and a more effective subcontracting policy.

5. The raw material planning will reduce the orders lead time as raw material shortage is reduced. It will also reduce the inventory levels of such items, since the materials requirement will be based on the capacity plan.

6. The scheduling of production based on delivery priority will also reduce the order lead times and inventory levels at the sign shop.

7. The interfaces with the inventory system will considerably reduce the amount of manual work and provide more accurate information.

8. The override capabilities and flexibility provided for the user, together with the data integrity, will enable him to deal with unusual situations that may occur.

As indirect results, the implementation and use of the system will provide the WVDOH with the following improvements:
1. Considerable reduction in the investment levels in finished goods and raw material inventory, as the overall inventory level is reduced.

2. Lower production and inventory costs, as the inventory level and the lead times are reduced, and the production resources are utilized more efficiently.

3. Capability of analyzing the performance of the production system by comparing completed versus planned.

4. Recovery of the confidence of the districts in the production system with the reduction of the lead times and the use of a fair inventory distribution policy.

5. Better control of the cash flow and budget, as the purchase of raw materials and subcontracted signs are planned in advance.

6. Better "customer" service levels with the reduction of shortage of signs and materials at the districts.

5.2 Coding and Implementation

The logic of the system was developed using structured architecture and documented in data flow diagrams, a data dictionary, and process narratives that can now be used as the input of the coding phase. The use of such methodology proved to be efficient and helpful in this study, assuring data accuracy, reducing errors during the design, and enabling the representation of the entire system in an understandable manner. The system can be coded in any available computer
language that has the ability to interface with the existent inventory system used by the WVDOH. The inventory system resides in their mainframe and is coded in SAS. Also, due to the high level of computation requirements in certain modules, the full implementation of the system in microcomputers may not be feasible. The coding can be done in modules, based on the modules defined in the logic structure. In addition to the processes described in this logic structure, the programming of the system includes the creation of the data files, their updating capabilities, and data input consistency checking.

The testing and implementation of the system is one of the most important stages of the physical design of a system. The testing of the system will determine if minor corrections are necessary before implementation. The modules can be tested separately, however, the integrated system should also be tested. After the corrections dictated by the test phase are made, the system can be implemented. The user's training and the establishment of the procedures to be followed are fundamental for the proper use and operation of the system. User and references manuals should be prepared, and made available during this phase. The present system should be operated in parallel until the implementation is successfully completed.

5.3 Recommendations for Future Research

1. After implementing the system, the comparison of the
order lead times and inventory levels with the present ones can be done. Since the inventory transactions are recorded, at some point in the future, enough data will be available for such a study, and qualitative information about the improvements generated by implementing the system can be given.

2. The analysis of set up procedures in the fabrication of signs (specially the screen preparation for screening standard signs) and the task design to improve the set up times would enable the reduction of the fabrication lot sizes presently used. This would considerably reduce the production lead times and also cause a positive impact in the production system as a whole.

3. The need for a make-or-buy analysis for each end item produced at the sign shop. This analysis would define the products in one of each categories:

   a - Always make;

   b - Make if there is capacity, and subcontract if no capacity is available;

   c - Always subcontract;

   In addition to this definition, the make-or-buy analysis should also define for each category b product, the preference for subcontracting. This information would help the planner define the priority for subcontracting when capacity is not
available for completing the production plan.

4. The production schedule in the proposed design adopts a simplified procedure where three different scheduling lists are generated by loading the production orders at the non-shared resources. The user has the decision of which production order to delete from the schedule if a shared resource is a bottleneck. In order to help the user in this decision for scheduling the shared resources, an expert system can be designed. This expert system would consider the flexibility of the resources and the approximate work-in-process levels at each resource (given by the user) and use rules to define which production orders are best eliminated to make the shared resources schedule feasible.

5. An analysis of the cost of maintaining the screens for standard sign fabrication against the cost reduction caused by smaller lot sizes would determine if there are some screens that should be maintained to reduce the overall cost.
REFERENCES


with Independent Demand", Production and Inventory Management Journal, Second Quarter, 1985, pp 146-151.


[27]. Hong, S., Maleyeff, J., "Production Planning and Master Scheduling with Spreadsheets", *Production and Inventory Management Journal*, First Quarter, 1987, pp 46-54.


[33]. Lamberty, W. E.," MRPII in a Custom Fabrication


[38]. Mcleavey D. W., Narasimhan S. L., Production Planning and Inventory Control, Allyn and Bacon, Inc., Boston, 1985.


[61]. Sounderpandian, J., "MRP on Spreadsheets: A Do-It-


APPENDIX A

DATA FLOW DIAGRAMS
DATA DICTIONARY
PROCESS NARRATIVES
Fig. 21 - Context Diagram
Fig. 22 - Level 0 DFD
Fig. 23 - Level 1 DFD - Control Inventory
Fig. 24 - Level 2 DFD - Input Order
Fig. 25 - Level 2 DFD - Allocate Order
Fig. 26 - Level 3 DFD - Generate Order Rec
Fig. 27 - Level 2 DFD - Control Delivery
<DEMAND-FORECAST>  <PREDICTED-DEMAND>

<MPS-NUMBER>

<MPS-DATA>

<MPS-REQ>

<PLAN-REPORTS>

<DAYS-PLAN>

<PROD-ROUTE>

PLMN

MPS

{2-1.DFD}

<RESOURCES>

<Routes>

<DAYS-SCHD>

<PROD-ROUTE>

<PO-SCHD-DATA>

<SCHD-PRIORITY>

<CONTROL ProduCtiON>

{2-3.DFD}

<RESOURCE-REC>

<RESOURCES>

<RESOURCE-SCHD-CHANGE>

<DELIVERY-DISTRICTS-LIST>

<ON-HAND>

<PO-DATA>

<RESOURCE-PLAN-CHANGE>

<RESOURCE-REC>

<PO-DATA>

<MATERIAL-REPORT>

<PR-DATA>

<PURCHASE-ORDER>

<ON-HAND>

Fig. 28 - Level 1 DFD - Plan Production
Fig. 29 - Level 2 DFD - Plan MPS
Fig. 30 - Level 3 DFD - Manage Demand
Fig. 31 - Level 3 DFD - Manage MPS
Fig. 32 - Level 3 DFD - Manage Capacity
Fig. 33 - Level 4 DFD - Find Feasible Plan
Fig. 34 - Level 2 DFD - Plan Materials
Fig. 35 - Level 3 DFD - Manage PR and RM-PR
Fig. 36 - Level 3 DFD - Purchase PR and RM-PR
Fig. 37 - Level 3 DFD - Receive PR and RM-PR
Fig. 38 - Level 2 DFD - Control Production
Fig. 39 - Level 3 DFD - Schedule Production
Fig. 40 - Level 4 DFD - Find Feasible Schedule
Fig. 41 - Level 1 DFD - Manage Data
Fig. 42 - Level 2 DFD - Manage Promised Values
DISTRICT-ORDER = ORDER-NUMBER
+ ORDER-DATE
+ { ORDER-ITEM }

ORDER = ORDER-NUMBER
+ ORDER-DATE
+ { ORDER-ITEM }

ORDER-APPROVED = ORDER-NUMBER
+ ORDER-DATE
+ { ORDER-ITEM }

ORDER-CHANGED = ORDER-CHANGED-NUMBER
+ ORDER-CHANGED-DATE
+ { ORDER-CHANGED-ITEM }

ORDER-ITEM = ORDER-NUMBER
+ ITEM-NUMBER
+ TYPE
+ SUBCODE
+ ITEM-QUANT

ORDER-CHANGED-ITEM = ORDER-CHANGED-NUMBER
+ CHANGED-ITEM-NUMBER
+ CHANGED-TYPE
+ CHANGED-SUBCODE
+ CHANGED-ITEM-QUANT

ORDER-NUMBER = YEAR-REF
+ DISTRICT
+ ORDER-ID

ORDER-DATE = ORDER-DAY
+ ORDER-MONTH

ORDER-MONTH = YYMM ( Year and Month )

ORDER-NOT-APPROVED = NOT-APPROVED-ORDER-NUMBER
ON-HAND = TYPE
+ SUBCODE
+ DISTRICT
+ ON-HAND-QUANT
+ LOW-LEVEL
+ HIGH-LEVEL
+ UNIT-COST

DEMAND-FORECAST = TYPE
+ SUBCODE
+ DISTRICT
+ MONTH
+ FORECAST-UNITS

NET-DEMAND = TYPE
+ SUBCODE
+ DISTRICT
+ MONTH
+ NET-UNITS

ORDER-REPORT = DISTRICT
+ MONTH
+ TYPE
+ SUBCODE
+ TOTAL-ORDERED
+ EXPECTED-ORDERS
+ USAGE
+ PRESENT-INVENTORY
+ MINIMUM-INVENTORY
+ MAXIMUM-INVENTORY
+ VALUE-ORDERED
+ OPEN-ORDERS
+ ( ( OTHER-DISTRICTS-REP ) )
DATA DICTIONARY

OTHER-DISTRICTS-REP = DISTRICT
  + TYPE
  + SUBCODE
  + PRESENT-INVENTORY
  + MINIMUM-INVENTORY
  + MAXIMUM-INVENTORY
  + USAGE

ORDER-REC = ORDER-REC-NUMBER
  + ORDER-REC-ITEM
  + ORDER-REC-TYPE
  + ORDER-REC-SUBCODE
  + ORDER-REC-QUANT
  + PROMISED-BY-ON-HAND
  + PROMISED-BY-MPS
  + PROMISED-BY-PO
  + PROMISED-BY-PR
  + UNITS-DELIVERED
  + UNITS-STILL-NOT-PROMISED
  + UNITS-REQ-ADJUSTMENT

MPS-REC = MPS-REC-NUMBER
  + MPS-REC-TYPE
  + MPS-REC-SUBCODE
  + MPS-REC-UNITS
  + MPS-PLAN-MONTH
  + MPS-ORIG-MONTH
  + MPS-DESTINY
  + MPS-AVAILABLE-TO-PROMISE

PO-REC = PO-REC-NUMBER
  + PO-REC-TYPE
  + PO-REC-SUBCODE
  + PO-REC-UNITS
  + PO-PLAN-MONTH
  + PO-ORIG-MONTH
  + PO-AVAILABLE-TO-PROMISE
  + UNITS-REQ-ADJUSTMENT
  + PO-REC-STATUS
PO-REC-STATUS = [ Planned | Scheduled ]

ON-HAND-REC = ON-HAND-REC-TYPE
    + ON-HAND-REC-SUBCODE
    + ON-HAND-UNITS
    + ON-HAND-AVAILABLE-TO-PROMISE
    + UNITS-REQ-ADJUSTMENT

PO-ORDER-REC = PO-ORDER-REC-ORDER
    + PO-ORDER-REC-ITEM
    + PO-ORDER-REC-PO
    + PO-ORDER-REC-UNITS

PR-REC = PR-REC-NUMBER
    + PR-REC-TYPE
    + PR-REC-SUBCODE
    + PR-REC-UNITS
    + PR-PLAN-MONTH
    + PR-ORIG-MONTH
    + PR-AVAILABLE-TO-PROMISE
    + UNITS-STILL-NOT-PURCHASED
    + UNITS-RECEIVED

PR-PURCHASE-ORDER = PR-REC-NUMBER
    + PURCHASE-ORDER-NUMBER
    + PURCHASE-ORDER-ITEM
    + PURCHASED-UNITS
    + RECEIVED-UNITS

MPS-ORDER-REC = MPS-ORDER-REC-ORDER
    + MPS-ORDER-REC-ITEM
    + MPS-ORDER-REC-MPS
    + MPS-ORDER-REC-UNITS

PR-ORDER-REC = PR-ORDER-REC-ORDER
    + PR-ORDER-REC-ITEM
    + PR-ORDER-REC-PR
    + PR-ORDER-REC-UNITS
DATA DICTIONARY

ON-HAND-ORDER-REC = ON-HAND-ORDER-REC-ORDER
  + ON-HAND-ORDER-REC-ITEM
  + ON-HAND-ORDER-REC-UNITS

MPS-REQ = MPS-REQ-TYPE
           MPS-REQ-SUBCODE
           MPS-REQ-UNITS

DELIVERY-REC = WEEK-NUMBER
               + DISTRICT

THIS-WEEK = WEEK-NUMBER

ALLOC-PRIORITY-LIST = ORDER-NUMBER
                       + ORDER-PRIORITY

PROD-REC = TYPE
           + SUBCODE
           + DESCRIPTION
           + LOT-SIZE
           + PROD-TYPE
           + PROD-CLASS
           + PROD-DESTINY
           + PROD-FAMILY
           + ( PURCHASE-LEAD-TIME )
           + ( PRIORITY-TO-PURCHASE )

PROD-CLASS = [ a | b ]

PROD-TYPE = [ Standard | Special | Interstate ]

PROD-DESTINY = [ Production | Purchase | Both ]

CHANGED-ALLOC-PRIORITY = ORDER-NUMBER
                         + ORDER-PRIORITY
ABOVE-FORECAST-REP = DISTRICT
+ TYPE
+ SUBCODE
+ QUANT-ORDERED
+ NET-UNITS

ORDER-BROKEN = ORDER-BROKEN-NUMBER
+ BROKEN-ITEM-NUMBER
+ BROKEN-QUANT
+ MPS-NUMBER

DELIVERY-LIST = DISTRICT-TO-DELIVER
+ DELIVERY-PRODUCTS

DELIVERY-PRODUCTS = DELIVERY-TYPE
+ DELIVERY-SUBCODE
+ DELIVERY-DESCRIPTION
+ DELIVERY-UNITS
+ DELIVERY-ORDER
+ DELIVERY-ITEM

WEEK-DELIVERY = WEEK-NUMBER

ORDER-DELIVERED = DELIVERED-ORDER
+ DELIVERED-ITEM
+ DELIVERED-UNITS

DELIVERED-ORDER = ORDER-NUMBER

DELIVERED-ITEM = ITEM-NUMBER

INPUT-ORDER-DATA = ORDER-CHANGED
+ DISTRICT-TO-APPROVE
+ DISTRICT-TO-REPORT
+ ORDER-NOT-APPROVED

ALLOCATION-REPORT = ALLOC-PRIORITY-LIST
+ ABOVE-FORECAST-ORDER
ALLOCATE-ORDER-DATA = THIS-WEEK
  + THIS-MONTH
  + CHANGED-ALLOC-PRIORITY
  + EXCESS

FIRST-MONTH = MONTH

THIS-MONTH = MONTH

DEMAND-REPORT = DEMAND-REPORT-TYPE
  + DEMAND-REPORT-SUBCODE
  + DEMAND-REPORT-DISTRICT
  + DEMAND-REPORT-MONTH
  + DEMAND-REPORT-UNITS

PREDICTED-DEMAND = TYPE
  + SUBCODE
  + DISTRICT
  + MONTH
  + PREDICTED-UNITS

BOM-REC = PARENT-TYPE
  + PARENT-SUBCODE
  + SON-TYPE
  + SON-SUBCODE
  + SON-QUANT (for each unit of the parent)

MPS-USER-DATA = MPS-USER-TYPE
  + MPS-USER-SUBCODE
  + MPS-USER-UNITS
  + MPS-USER-PLAN-MONTH

MPS-TO-DELETE = MPS-REC-NUMBER

DAYS-PLAN = MONTH
  + DAYS
RESOURCE-REC = RESOURCE-NUMBER
+ RESOURCE-DESCRIPTION
+ RESOURCE-WORK-HOURS
+ RESOURCE-QUANT
+ RES-EFFICIENCY

RESOURCE-PLAN-CHANGE = PLAN-MONTH
+ RESOURCE-NUMBER
+ ( RES-CHANGE-WORK-HOURS )
+ ( RES-CHANGE-QUANT )
+ ( RES-CHANGE-EFFICIENCY )

RESOURCE-PLAN-CAPACITY = PLAN-MONTH
+ RESOURCE-NUMBER
+ RESOURCE-DESCRIPTION
+ RESOURCE-WORK-HOURS
+ RESOURCE-QUANT
+ RES-EFFICIENCY
+ RES-HOURS-AVAILABLE

RESOURCE-PLAN-REC = PLAN-MONTH
+ RESOURCE-NUMBER
+ RESOURCE-QUANT
+ RESOURCE-WORK-HOURS
+ RES-HOURS-AVAILABLE
+ RES-EFFICIENCY
+ WORK-LOAD

WORK-LOAD = HOURS

PLAN-WORK-LOAD-REPORT = REPORT-PLAN-MONTH
+ REPORT-RES-NUMBER
+ REPORT-RES-DESCRIPTION
+ REPORT-RES-QUANT
+ REPORT-RES-HOURS-AVAILABLE
+ REPORT-RES-EFFICIENCY
+ REPORT-RES-LOAD-RATIO
+ REPORT-RES-STATUS
DATA DICTIONARY

REPORT-RES-STATUS = [ Bottleneck | Non-Bottleneck ]

PROD-ROUTE = TYPE
  + SUBCODE
  + OPERATION-CODE
  + ROUTE-TYPE

ROUTE-TYPE = [ Shared | Not Shared ]

OPERATION-REC = OPERATION-CODE
  + OPERATION-DESCRIPTION
  + OPERATION-MACHINE
  + OPERATION-WORKER
  + SET-UP-TIME
  + OP-TIME-PER-UNIT

SET-UP-TIME = HOURS

OP-TIME-PER-UNIT = HOURS

OPERATION-MACHINE = RESOURCE-NUMBER

OPERATION-WORKER = RESOURCE-NUMBER

MPS-PO-REPORT = PO-PLANNED-REPORT
  + MPS-PLANNED-REPORT

PO-PLANNED-REPORT = PO-PLANNED-NUMBER
  + PO-PLANNED-TYPE
  + PO-PLANNED-SUBCODE
  + PO-PLANNED-UNITS
  + PO-PLANNED-BOOK
  + PO-PLANNED-MONTH
  + PO-ORIGINAL-MONTH
  + PO-MAX-EXPEDITION
MPS-PLANNED-REPORT = MPS-PLANNED-NUMBER
              + MPS-PLANNED-TYPE
              + MPS-PLANNED-SUBCODE
              + MPS-PLANNED-UNITS
              + MPS-PLANNED-BOOK
              + MPS-PLANNED-MONTH
              + MPS-ORIGINAL-MONTH
              + MPS-MAX-EXPEDITION
              + MPS-PLANNED-DESTINY

PO-DATA = PO-NUMBER
          + PO-NEW-PLAN-MONTH

MPS-DATA = MPS-NUMBER
            ( + MPS-NEW-PLAN-MONTH )
            ( + MPS-NEW-DESTINY )

RM-PR = RM-PR-NUMBER
        + RM-PR-TYPE
        + RM-PR-SUBCODE
        + RM-PR-UNITS
        + RM-PR-PLAN-MONTH
        + RM-PURCHASE-ORDER
        + RM-PURCHASE-ITEM
        + EXPECTED-RECEIPT
        + UNITS-RECEIVED

EXPECTED-RECEIPT = MONTH

RM-PURCHASE-ORDER = PURCHASE-ORDER-NUMBER

NEW-RM-PR-DATA = NEW-RM-PR-TYPE
                 + NEW-RM-PR-SUBCODE
                 + NEW-RM-PR-UNITS
                 + NEW-RM-PR-PLAN-MONTH
RM-PURCHASE-ITEM = PURCHASE-ORDER-ITEM

NEW-PR-DATA = NEW-PR-TYPE
+ NEW-PR-SUBCODE
+ NEW-PR-UNITS
+ NEW-PR-PLAN-MONTH

DELETED-PR = DELETED-PR-NUMBER

PR-AND-RM-PR-REPORT = PR-REPORT
+ RM-PR-REPORT

PR-REPORT = PR-REP-PR
+ PR-REP-TYPE
+ PR-REP-SUBCODE
+ PR-REP-DESCRIPTION
+ PR-REP-UNITS
+ PR-BOOK-LEVEL
+ PR-WHEN-PLANNED
+ PR-REP-WHEN-NEEDED
+ PR-REP-WHEN-PURCHASE

RM-PR-REPORT = RM-PR-REP-RM-PR
+ RM-PR-REP-TYPE
+ RM-PR-REP-SUBCODE
+ RM-PR-REP-DESCRIPTION
+ RM-PR-REP-UNITS
+ RM-PR-REP-WHEN-PLANNED
+ RM-PR-REP-WHEN-PURCHASE

PR-REP-WHEN-NEEDED = MONTH

PR-REP-WHEN-PURCHASE = MONTH

RM-PR-REP-WHEN-NEEDED = MONTH
RM-PR-REP-WHEN-PURCHASE = MONTH

RM-PR-PURCHASED = PURCHASED-RM-PR
  + PURCHASED-ORDER
  + PURCHASED-ITEM
  + PURCHASED-EXPECTING-RECEIPT

PURCHASE-ORDER = PURCHASE-ORDER-NUMBER
  + PURCHASE-ORDER-ITEM
  + PURCHASE-ORDER-TYPE
  + PURCHASE-ORDER-SUBCODE
  + PURCHASE-ORDER-QUANT
  + PURCHASE-ORDER-RECEIVED

PR-PURCHASED = PURCHASED-PR
  + PURCHASED-ORDER
  + PURCHASED-ITEM
  + PURCHASED-EXPECTING-RECEIPT

DAYS-IN-WEEK = MONTH
  + DAYS

RESOURCE-SCHD-CAPACITY = RESOURCE-NUMBER
  + RESOURCE-DESCRIPTION
  + RESOURCE-WORK-HOURS
  + RESOURCE-QUANT
  + RES-EFFICIENCY
  + RES-HOURS-AVAILABLE

RESOURCE-SCHD-REC = RESOURCE-NUMBER
  + RES-HOURS-AVAILABLE
  + RES-EFFICIENCY
  + WORK-LOAD

RESOURCE-SCHD-CHANGE = RESOURCE-NUMBER
  + ( RES-CHANGE-WORK-HOURS )
  + ( RES-CHANGE-QUANT )
  + ( RES-CHANGE-EFFICIENCY )
SCHD-PRIORITY-LIST = DISTRICT-DELIVER-LIST
+ STANDARD-PRIORITY-LIST
+ SPECIAL-PRIORITY-LIST
+ INTERSTATE-PRIORITY-LIST

DISTRICT-DELIVERY-LIST = DISTRICT-DELIVER-LIST-NUMBER
+ DISTRICT-DELIVER-LIST-PRIORITY

STANDARD-PRIORITY-LIST = STD-LIST-PO-NUMBER
STD-LIST-TYPE
STD-LIST-SUBCODE
STD-LIST-UNITS
STD-LIST-PLAN-MONTH
STD-LIST-ORIG-MONTH
STD-LIST-BOOK-LEVEL

SPECIAL-PRIORITY-LIST = SPC-LIST-PO-NUMBER
SPC-LIST-TYPE
SPC-LIST-SUBCODE
SPC-LIST-UNITS
SPC-LIST-PLAN-MONTH
SPC-LIST-ORIG-MONTH
SPC-LIST-BOOK-LEVEL

INTERSTATE-PRIORITY-LIST = INT-LIST-PO-NUMBER
INT-LIST-TYPE
INT-LIST-SUBCODE
INT-LIST-UNITS
INT-LIST-PLAN-MONTH
INT-LIST-ORIG-MONTH
INT-LIST-BOOK-LEVEL

DELIVERY-DISTRICTS-LIST = DISTRICT
+ USER-DELIVER-PRIORITY

USER-DELIVERY-PRIORITY = [ 1 | 2 | 3 | 4 ]
DATA DICTIONARY

SCHD-PRIORITY = SCHD-PRIORITY-PO-NUMBER
  + USER-SCHD-PRIORITY

PO-TO-SCHEDULE = PO-NUMBER
  + PO-CATEGORY
  + PO-PRIORITY
  + PO-STATUS
  + PO-FAMILY

PO-STATUS = [ "Material Checked" | "Loaded at Non-Shared" | "Loaded at Shared" | "Discharged" ]

RM-AVAILABLE = RM-TYPE
  + RM-SUBCODE
  + RM-UNITS

SCHD-WORK-LOAD-REPORT = RESOURCE-SCHED-LOAD-REPORT
  + PO-LOADED-REPORT

RESOURCE-SCHED-LOAD-REPORT = REPORT-RES-NUMBER
  + REPORT-RES-DESCRIPTION
  + REPORT-RES-QUANT
  + REPORT-RES-HOURS-AVAILABLE
  + REPORT-RES-EFFICIENCY
  + REPORT-RES-STATUS
  + REPORT-LOAD-RATIO

PO-LOADED-REPORT = STD-PO-REPORT
  + SPC-PO-REPORT
  + INT-PO-REPORT
STD-PO-REPORT = REPORT-PO-NUMBER + REPORT-PO-TYPE + REPORT-PO-SUBCODE + REPORT-PO-UNITS + REPORT-PO-PLAN-MONTH + REPORT-PO-ORIG-MONTH + REPORT-PO-BOOK-LEVEL

SPC-PO-REPORT = REPORT-PO-NUMBER + REPORT-PO-TYPE + REPORT-PO-SUBCODE + REPORT-PO-UNITS + REPORT-PO-PLAN-MONTH + REPORT-PO-ORIG-MONTH + REPORT-PO-BOOK-LEVEL

INT-PO-REPORT = REPORT-PO-NUMBER + REPORT-PO-TYPE + REPORT-PO-SUBCODE + REPORT-PO-UNITS + REPORT-PO-PLAN-MONTH + REPORT-PO-ORIG-MONTH + REPORT-PO-BOOK-LEVEL

PO-DISCHARGED = PO-NUMBER

PO-DISCHARGED-REPORT = PO-NUMBER

PO-RECONSIDERED = PO-NUMBER

SCHD-LIST = STD-SCHD-LIST + SPC-SCHD-LIST + INT-SCHD-LIST
STD-SCHD-LIST = LIST-PO-NUMBER 
+ LIST-PO-TYPE 
+ LIST-PO-SUBCODE 
+ LIST-PO-UNITS 
+ LIST-PO-DESCRIPTION 
+ LIST-PO-BOOK-LEVEL

SPC-SCHD-LIST = LIST-PO-NUMBER 
+ LIST-PO-TYPE 
+ LIST-PO-SUBCODE 
+ LIST-PO-UNITS 
+ LIST-PO-DESCRIPTION 
+ LIST-PO-BOOK-LEVEL

INT-SCHD-LIST = LIST-PO-NUMBER 
+ LIST-PO-TYPE 
+ LIST-PO-SUBCODE 
+ LIST-PO-UNITS 
+ LIST-PO-DESCRIPTION 
+ LIST-PO-BOOK-LEVEL

FEASIBLE-SCHD-REPORT = WORK-SCHD-LOAD-REPORT 
+ SCHD-LIST 
+ PO-DISCHARGED-REPORT

PO-COMPLETED = PO-COMPLETED-NUMBER 
+ PO-COMPLETED-UNITS

PO-TO-KEEP-SCHEDULED = PO-NUMBER

NOT-EXECUTED-REPORT = PO-NUMBER

ORDER-ON-HAND-INFO = ORDER-REC-NUMBER 
+ ORDER-REC-ITEM 
+ NEW-PROMISED-BY-ON-HAND

REPORT-ON-HAND = ON-HAND-REQ-ADJUSTMENT-REPORT 
+ ON-HAND-AVAILABLE-REPORT
ON-HAND-REQ-ADJUSTMENT-REPORT = ADJ-REP-TYPE
+ ADJ-REP-SUBCODE
+ ADJ-REP-UNITS
+ ADJ-REP-ADJ-UNITS

ON-HAND-AVAILABLE-REPORT = AVAIL-REP-TYPE
+ AVAIL-REP-SUBCODE
+ AVAIL-REP-UNITS
+ AVAIL-REP-UNITS-AVAIL

REPORT-MPS = MPS-AVAILABLE-REPORT

MPS-AVAILABLE-REPORT = AVAIL-REP-TYPE
+ AVAIL-REP-SUBCODE
+ AVAIL-REP-UNITS
+ AVAIL-REP-UNITS-AVAIL

ORDER-MPS-INFO = ORDER-REC-NUMBER
+ ORDER-REC-ITEM
+ MPS-REC-NUMBER
+ NEW-PROMISED-BY-MPS

REPORT-PO = PO-REQ-ADJUSTMENT-REPORT
+ PO-AVAILABLE-REPORT

PO-REQ-ADJUSTMENT-REPORT = ADJ-REP-TYPE
+ ADJ-REP-SUBCODE
+ ADJ-REP-UNITS
+ ADJ-REP-ADJ-UNITS

PO-AVAILABLE-REPORT = AVAIL-REP-TYPE
+ AVAIL-REP-SUBCODE
+ AVAIL-REP-UNITS
+ AVAIL-REP-UNITS-AVAIL

REPORT-PR = PR-STILL-NOT-PURCHASED-REPORT
+ PR-AVAILABLE-REPORT
DATA DICTIONARY

ORDER-PO-INFO = ORDER-REC-NUMBER
+ ORDER-REC-ITEM
+ PO-REC-NUMBER
+ NEW-PROMISED-BY-PO

PR-STILL-NOT-PURCHASED-REPORT = NOT-PURCH-REP-TYPE
+ NOT-PURCH-REP-SUBCODE
+ NOT-PURCH-REP-PR-UNITS
+ NOT-PURCH-REP-UNITS

PR-AVAILABLE-REPORT = AVAIL-REP-TYPE
+ AVAIL-REP-SUBCODE
+ AVAIL-REP-UNITS
+ AVAIL-REP-UNITS-AVAIL

ORDER-PR-INFO = ORDER-REC-NUMBER
+ ORDER-REC-ITEM
+ PR-REC-NUMBER
+ NEW-PROMISED-BY-PR

ORDER-INFO = ORDER-REC-NUMBER
+ ORDER-REC-ITEM

REPORT-ORDER = REP-ORDER-REC
+ ( REP-PO-ORDER-REC )
+ ( REP-PR-ORDER-REC )
+ ( REP-MPS-ORDER-REC )

REP-MPS-ORDER-REC = REP-MPS-NUMBER
+ REP-UNITS

REP-PO-ORDER-REC = REP-PO-NUMBER
+ REP-UNITS

REP-PR-ORDER-REC = REP-PR-NUMBER
+ REP-UNITS
PO-MPS-INFO = PO-MPS-INFO-PO  
+ PO-MPS-INFO-MPS  
+ PO-MPS-INFO-MPS-DESTINY

PR-MPS-INFO = PR-MPS-INFO-PR  
+ PR-MPS-INFO-MPS  
+ PR-MPS-INFO-MPS-DESTINY

OTHER-RECORDS-INFO = PO-MPS-INFO  
+ PR-MPS-INFO  
+ ORDER-PR-INFO  
+ ORDER-PO-INFO  
+ ORDER-MPS-INFO  
+ ORDER-ON-HAND-INFO

REP-ORDER-REC = REP-ORDER-REC-NUMBER  
+ REP-ORDER-REC-ITEM  
+ REP-ORDER-REC-TYPE  
+ REP-ORDER-REC-SUBCODE  
+ REP-ORDER-REC-QUANT  
+ REP-PROMISED-BY-ON-HAND  
+ REP-PROMISED-BY-MPS  
+ REP-PROMISED-BY-PO  
+ REP-PROMISED-BY-PR  
+ REP-UNITS-DELIVERED  
+ REP-STILL-NOT-PROMISED  
+ REP-REQ-ADJUSTMENT

REPORT-OTHER-RECORDS = REPORT-ON-HAND  
REPORT-MPS  
REPORT-PO  
REPORT-PR

REPORT-RECORDS = REPORT-ORDER  
+ REPORT-OTHER-RECORDS
RECORDS-INFO = ORDER-INFO
+ OTHER-RECORDS-INFO

INPUT-ORDER-DATA = ORDER-CHANGED
+ DISTRICT-TO-APPROVE
+ ORDER-NOT-APPROVED
+ DISTRICT-TO-REPORT

ALLOCATION-REPORT = ALLOC-PRIORITY-LIST
+ ABOVE-FORECAST-ORDER

ALLOCATE-ORDER-DATA = THIS-WEEK
+ CHANGED-ALLOC-PRIORITY
+ EXCESS
+ THIS-MONTH

ORDER-DATA = INPUT-ORDER-DATA
+ ALLOCATE-ORDER-DATA

INVENTORY-REPORTS = ORDER-REPORT
+ ALLOCATION-REPORT
+ DELIVERY-LIST

DELIVERY-DATA = DELIVERY-DISTRICTS
+ DELIVERY-WEEK
+ ORDER-DELIVERED

MANAGE-MPS-DATA = MPS-USER-DATA
+ MPS-TO-DELETE

CAPACITY-REPORTS = RESOURCE-PLAN-CAPACITY
+ PLAN-WORK-LOAD-REPORT
+ MPS-PO-REPORT

PLAN-REPORTS = DEMAND-REPORT
+ CAPACITY-REPORT
DATA DICTIONARY

PR-DATA = NEW-PR-DATA
+ NEW-RM-PR-DATA
+ DELETED-PR
+ PR-PURCHASED

MATERIAL-REPORT = PR-AND-RM-PR-REPORT

SCHD-REPORTS = AVAILABLE-SCHD-CAPACITY
+ SCHD-PRIORITY-LIST
+ FEASIBLE-SCHD-REPORT

PO-SCHD-DATA = PO-DISCHARGED
+ PO-RECONSIDERED

FEASIBLE-SCHD-REPORT = WORK-SCHD-LOAD-REPORT
+ SCHD-LIST
+ PO-DISCHARGED-REPORT

RESOURCE-CHANGE = RESOURCE-PLAN-CHANGE
+ RESOURCE-SCHD-CHANGE

PRODUCTION-REPORTS = PLAN-REPORTS
+ MATERIAL-REPORT
+ SCHD-REPORTS

PRODUCTION-DATA = MPS-DATA
+ MANAGE-MPS-DATA
+ PO-DATA
+ PO-SCHD-DATA
+ PR-DATA
+ DAYS-PLAN
+ DAYS-SCHD
+ SCHD-PRIORITY
+ DELETED-PO

DAYS-SCHD = DAYS-IN-WEEK
DELETED-PO = DELETED-PO-NUMBER

REPORTS = INVENTORY-REPORTS
          + PRODUCTION-REPORTS
          + REPORT-RECORDS

INVENTORY-PROD-DATA = ORDER-DATA
                       + DELIVERY-DATA
                       + PREDICTED-DEMAND
                       + PRODUCTION-DATA
1. For each DISTRICT-ORDER

1.1 Read ORDER-NUMBER
   ORDER-DATE

1.1.1 For each ORDER-ITEM

1.1.1.1 Read ITEM-NUMBER
    ITEM-QUANT
    TYPE
    SUBCODE

1.2 Set ORDER = DISTRICT-ORDER

1.3 Write ORDER in ORDERS TO APPROVE.
1. For each ORDER-CHANGED
1.1 Read ORDER-CHANGED-NUMBER
1.1.1 Select the policy which applies:
1.1.1.1 Case 1 (Delete Order)
   1.1.1.1.1 Delete ORDER in ORDERS TO APPROVE with
   ORDER-NUMBER = ORDER-CHANGED-NUMBER.
1.1.1.2 Case 2 (Delete Item)
   1.1.1.2.1 For each ORDER-CHANGED-ITEM
   1.1.1.2.1.1 Read CHANGED-ITEM-NUMBER
   1.1.1.2.1.2 Delete ORDER-ITEM in ORDERS TO APPROVE with
   ORDER-NUMBER = ORDER-CHANGED-NUMBER and
   ITEM-NUMBER = CHANGED-ITEM-NUMBER.
1.1.1.3 Case 3 (Change Item Data)
   1.1.1.3.1 For each ORDER-CHANGED-ITEM
   1.1.1.3.1.1 Read CHANGED-ITEM-NUMBER
   CHANGED-ITEM-QUANT
   CHANGED-TYPE
   CHANGED-SUBCODE
   1.1.1.3.1.2 Set ORDER-NUMBER = ORDER-CHANGED-NUMBER
   ITEM-NUMBER = CHANGED-ITEM-NUMBER
   ITEM-QUANT = CHANGED-ITEM-QUANT
   ITEM-TYPE = CHANGED-TYPE
   ITEM-SUBCODE = CHANGED-SUBCODE
PROCESS NARRATIVE

Process Name: CHANGE ORDER

Process Number: 1-1-2.PN

1.1.1.3.1.3 Write ORDER-ITEM in ORDERS TO APPROVE

1.1.1.4 Case 4 (Change Order Date)

1.1.1.4.1 Read ORDER-CHANGED-DATE

1.4.1.4.2 Set ORDER-NUMBER = ORDER-CHANGED-NUMBER
ORDER-DATE = ORDER-CHANGED-DATE

1.4.1.4.3 Write ORDER in ORDERS TO APPROVE.
# PROCESS NARRATIVE

**Process Name:** APPROVE ORDER GROUP  
**Process Number:** 1-1-3.PN

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>For each DISTRICT-TO-APPROVE</td>
</tr>
<tr>
<td>1.1</td>
<td>Read DISTRICT-TO-APPROVE</td>
</tr>
<tr>
<td>1.2</td>
<td>Select the policy which applies</td>
</tr>
<tr>
<td>1.2.1</td>
<td>Case 1 (Approve all Orders in a group)</td>
</tr>
<tr>
<td>1.2.1.1</td>
<td>For each ORDER in ORDER TO APPROVE with DISTRICT = DISTRICT-TO-APPROVE</td>
</tr>
<tr>
<td>1.2.1.1.1</td>
<td>Set ORDER-APPROVED = ORDER</td>
</tr>
<tr>
<td>1.2.1.1.2</td>
<td>Write ORDER-APPROVED in APPROVED ORDERS</td>
</tr>
<tr>
<td>1.2.1.1.3</td>
<td>Delete ORDER in ORDERS TO APPROVE</td>
</tr>
<tr>
<td>1.2.2</td>
<td>Case 2 (Approve Orders in a group but excluding some)</td>
</tr>
<tr>
<td>1.2.2.1</td>
<td>For each ORDER-NOT-APPROVED</td>
</tr>
<tr>
<td>1.2.2.1.1</td>
<td>Read ORDER-NOT-APPROVED-NUMBER</td>
</tr>
<tr>
<td>1.2.2.2</td>
<td>For each ORDER in ORDER TO APPROVE with DISTRICT = DISTRICT-TO-APPROVE and ORDER-NUMBER not equal ORDER-NOT-APPROVED-NUMBER</td>
</tr>
<tr>
<td>1.2.2.2.1</td>
<td>Set ORDER-APPROVED = ORDER</td>
</tr>
<tr>
<td>1.2.2.2.2</td>
<td>Write ORDER-APPROVED in APPROVED ORDERS</td>
</tr>
<tr>
<td>1.2.2.2.3</td>
<td>Delete ORDER in ORDERS TO APPROVE</td>
</tr>
</tbody>
</table>
1. For each DISTRICT-TO-REPORT entered

1.1 Read ORDER in ORDERS TO APPROVE with
   DISTRICT in ORDER-NUMBER = DISTRICT-TO-REPORT

1.1.1 For each ORDER

1.1.1.1 For each ORDER-ITEM

1.1.1.1.1 Set TOTAL-ORDERED = TOTAL-ORDERED + ITEM-QUANT
   using TYPE, SUBCODE and MONTH as index

1.1.2 For each TYPE in any ORDER

1.1.2.1 For each SUBCODE in any ORDER

1.1.2.1.1 Read PROD-CLASS in PRODUCTS using TYPE and
   SUBCODE as key.

1.1.2.1.2.1 If PROD-CLASS = "a" Then

1.1.2.1.2.1.1 Read ON-HAND-QUANT, LOW-LEVEL, HIGH-LEVEL
   and UNIT-COST in INVENTORY SYSTEM
   using TYPE, SUBCODE and
   DISTRICT = DISTRICT-TO-REPORT as key

1.1.2.1.2.1.2 Set PRESENT-INVENTORY = ON-HAND-QUANT
   MINIMUM-INVENTORY = LOW-LEVEL
   MAXIMUM-INVENTORY = HIGH-LEVEL
   using TYPE and SUBCODE as index

1.1.2.1.2.2 If PROD-CLASS = "b" Then

1.1.2.1.2.2.1 For each DISTRICT
PROCESS NARRATIVE

Process Name: GENERATE ORDER REPORT

Process Number: 1-1-4.PN

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.2.1.2.1.1</td>
<td>Read ON-HAND-QUANT, LOW-LEVEL and HIGH-LEVEL in INVENTORY SYSTEM using TYPE and SUBCODE as key.</td>
</tr>
<tr>
<td>1.1.2.1.2.1.2</td>
<td>Set PRESENT-INVENTORY = ON-HAND-QUANT MINIMUM-INVENTORY = LOW-LEVEL MAXIMUM-INVENTORY = HIGH-LEVEL using TYPE, SUBCODE and DISTRICT as index.</td>
</tr>
<tr>
<td>1.1.2.1.2.1.3</td>
<td>Read UNIT-COST in INVENTORY SYSTEM using TYPE, SUBCODE and DISTRICT = DISTRICT-TO-REPORT as key</td>
</tr>
<tr>
<td>1.1.2.1.3</td>
<td>For each MONTH in any ORDER</td>
</tr>
<tr>
<td>1.1.2.1.3.1</td>
<td>Read NET-DEMAND in NET DEMAND using TYPE, SUBCODE, MONTH and DISTRICT-TO-REPORT as key</td>
</tr>
<tr>
<td>1.1.2.1.3.2</td>
<td>Set EXPECTED-ORDERS = NET-DEMAND using TYPE, SUBCODE and MONTH as index.</td>
</tr>
<tr>
<td>1.1.2.1.3.3</td>
<td>Read DEMAND-FORECAST in FORECASTING SYSTEM using TYPE, SUBCODE, MONTH and DISTRICT = DISTRICT-TO-REPORT as key</td>
</tr>
<tr>
<td>1.1.2.1.3.4</td>
<td>Set USAGE = DEMAND-FORECAST using TYPE, SUBCODE, and MONTH as index</td>
</tr>
<tr>
<td>1.1.2.1.3.5</td>
<td>Set VALUE-ORDERED = TOTAL-ORDERED * UNIT-COST using TYPE, SUBCODE and MONTH as index</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: GENERATE ORDER REPORT

Process Number: 1-1-4.PN

1.2 Read ORDER-APPROVED in APPROVED ORDERS with
   DISTRICT in ORDER-NUMBER = DISTRICT-TO-REPORT

1.2.1 For each ORDER-APPROVED

1.3.1.1 For each ORDER-ITEM

1.3.1.1.1 If TYPE is in any ORDER

1.3.1.1.1.1 If SUBCODE is in any ORDER then

1.3.1.1.1.1.1 Set OPEN-ORDERS = OPEN-ORDERS + ITEM-QUANT
   using TYPE and SUBCODE as index

1.4 Read ORDER-REC in ORDER RECORDS using
   DISTRICT-TO-REPORT as key.

1.4.1 For each ORDER-REC

1.4.1.1 If ORDER-REC-TYPE is in any ORDER then

1.4.1.1.1 If ORDER-REC-SUBCODE is in any ORDER then

1.4.1.1.1.1 Set OPEN-ORDERS = OPEN-ORDERS + ITEM-QUANT
   using TYPE and SUBCODE as index

1.5 For each TYPE in any ORDER

1.5.1 For each SUBCODE in any ORDER
**PROCESS NARRATIVE**

Process Name: GENERATE ORDER REPORT  

Process Number: 1-1-4.PN

| 1.5.1.1 Write TOTAL-ORDERED  
EXPECTED-ORDERS  
USAGE  
MINIMUM-INVENTORY  
MAXIMUM-INVENTORY  
PRESENT-INVENTORY  
VALUE-ORDERED  
OPEN-ORDERS |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.1.2 If PRESENT-INVENTORY &gt; MINIMUM-INVENTORY then</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1.2.1 Write &quot;Reorder Point not Reached&quot;.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1.3 If EXPECTED-ORDERS &lt; TOTAL-ORDERED then</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1.3.1 Write &quot;Total Ordered Greater than Predicted&quot;.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1.4 If (OPEN-ORDERS + PRESENT-INVENTORY) &gt; MAXIMUM-INVENTORY then</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1.4.1 Write &quot;Total Ordered Exceeds Maximum Allowed&quot;.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1.5 If PROD-CLASS = &quot;b&quot; then</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1.5.1 For each DISTRICT not equal DISTRICT-TO-REPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 1.5.1.5.1.1 Write DISTRICT  
PRESENT-INVENTORY  
MINIMUM-INVENTORY  
MAXIMUM-INVENTORY  
USAGE. |
### PROCESS NARRATIVE

**Process Name:** SORT ORDER  
**Process Number:** 1-2-1.PN  
**Page:** 01

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Select the policy which applies</td>
<td></td>
</tr>
<tr>
<td>1.1 Case 1 (Compute Orders Allocation Priority)</td>
<td></td>
</tr>
<tr>
<td>1.1.1 Read THIS-WEEK</td>
<td></td>
</tr>
<tr>
<td>1.1.2. For each DELIVERY-REC in DELIVERY</td>
<td></td>
</tr>
<tr>
<td>1.1.2.1 For each DISTRICT</td>
<td></td>
</tr>
<tr>
<td>1.1.2.1.1 Set NEXT-DELIVERY-WEEK = MIN(WEEK-NUMBER)</td>
<td>using DISTRICT as index</td>
</tr>
<tr>
<td>1.1.3. Set PRIORITY = (NEXT-DELIVERY-WEEK) - (THIS-WEEK) + 1</td>
<td>using DISTRICT as index</td>
</tr>
<tr>
<td>1.1.4. For each ORDER-APPROVED in APPROVED ORDERS</td>
<td></td>
</tr>
<tr>
<td>1.1.4.1 Read DISTRICT</td>
<td></td>
</tr>
<tr>
<td>1.1.4.2 Set ORDER-PRIORITY = PRIORITY (DISTRICT)</td>
<td></td>
</tr>
<tr>
<td>1.1.4.2 Write ORDER-PRIORITY in APPROVED ORDERS</td>
<td></td>
</tr>
<tr>
<td>1.2 Case 2 (Generate Allocation Priority List)</td>
<td></td>
</tr>
<tr>
<td>1.2.5. For each ORDER-APPROVED in APPROVED ORDERS</td>
<td></td>
</tr>
<tr>
<td>1.2.5.1 Sort ORDER-APPROVED by ORDER-PRIORITY</td>
<td></td>
</tr>
<tr>
<td>1.2.5.2 Write ORDER-APPROVED ORDER-PRIORITY in ALLOC-PRIORITY-LIST</td>
<td></td>
</tr>
</tbody>
</table>
1.2.6 Write ALLOC-PRIORITY-LIST

1.3 Case 3 (Make Changes in Allocation Priority)

1.3.1 For each CHANGED-ALLOC-PRIORITY

1.3.1.1 Read ORDER-NUMBER
ORDER-PRIORITY

1.3.1.2 Write ORDER-PRIORITY in APPROVED ORDERS
using ORDER-NUMBER as key

2. Write THIS-WEEK in THIS WEEK

3. Go To Process CHECK FORECAST.
PROCESS NARRATIVE

Process Name: CHECK FORECAST
Process Number: 1-2-2.PN

2.2.1.1 Select "User Option"
2.2.1.1.1 If "User Option" is "Break the Order" then
2.2.1.1.1.1 Select the policy which applies:
2.2.1.1.1.1.1 Case 1 (Break the Order According to the Forecast)
2.2.1.1.1.1.1.1 Set EXCESS = (QUANT-ORDERED) - (NET-DEMAND)
2.2.1.1.1.1.2 Case 2 (Break the Order According to User Desired Quantity)
2.2.1.1.2.1 Read EXCESS from User
2.2.1.1.2.1 Read ORDER-ITEM in APPROVED ORDERS using TYPE, SUBCODE and DISTRICT as key
2.2.1.1.2.1.1 Sort ORDER-ITEM by ITEM-QUANT (higher quant = first position)
2.2.1.1.2.1.1.1 Select ORDER-ITEM from sorted group
2.2.1.1.2.1.1.1.1 If ITEM-QUANT > EXCESS then
2.2.1.1.2.1.1.1.1.1 Set ORDER-BROKEN-NUMBER = ORDER-NUMBER BROKEN-ITEM-NUMBER = ITEM-NUMBER BROKEN-QUANT = EXCESS
2.2.1.1.2.1.1.2 Write ORDER-BROKEN in BROKEN ORDERS
# PROCESS NARRATIVE

**Process Name:** CHECK FORECAST  
**Process Number:** 1-2-2.PN  
**Page:** 03

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1.1.2.1.1.1.1.3</td>
<td>Call Process GENERATE REQ MPS giving TYPE SUBCODE BROKEN-QUANT</td>
</tr>
<tr>
<td>2.2.1.1.2.1.1.1.1.4</td>
<td>Read MPS-NUMBER from Process GENERATE REQ MPS</td>
</tr>
<tr>
<td>2.2.1.1.2.1.1.1.1.5</td>
<td>Write MPS-NUMBER in BROKEN ORDERS using ORDER-BROKEN-NUMBER and BROKEN-ITEM-NUMBER as key</td>
</tr>
<tr>
<td>2.2.1.1.2.1.1.1.2</td>
<td>Otherwise (Excess divided in more items)</td>
</tr>
<tr>
<td>2.2.1.1.2.1.1.2.1</td>
<td>Do the following until EXCESS = 0</td>
</tr>
<tr>
<td>2.2.1.1.2.1.1.2.1.1</td>
<td>Set ORDER-BROKEN-NUMBER = ORDER-NUMBER</td>
</tr>
<tr>
<td></td>
<td>BROKEN-ITEM-NUMBER = ITEM-NUMBER</td>
</tr>
<tr>
<td></td>
<td>BROKEN-QUANT = ITEM-QUANT</td>
</tr>
<tr>
<td></td>
<td>EXCESS = EXCESS - ITEM-QUANT</td>
</tr>
<tr>
<td>2.2.1.1.2.1.1.2.1.2</td>
<td>Write ORDER-BROKEN in BROKEN ORDERS</td>
</tr>
<tr>
<td>2.2.1.1.2.1.1.2.1.3</td>
<td>Call Process GENERATE REQ MPS giving TYPE SUBCODE BROKEN-QUANT</td>
</tr>
<tr>
<td>Process Name: CHECK FORECAST</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---</td>
</tr>
<tr>
<td>Process Number: 1-2-2.PN</td>
<td>Page: 04</td>
</tr>
</tbody>
</table>

2.2.1.1.2.1.1.2.1.4 Read MPS-NUMBER from Process
GENERATE REQ MPS

2.2.1.1.2.1.1.2.1.5 Write MPS-NUMBER in BROKEN ORDERS
using ORDER-BROKEN-NUMBER
and BROKEN-ITEM-NUMBER as key

2.2.1.1.2.1.1.2.1.6 Delete ORDER-ITEM from sorted group

3. Write THIS-MONTH in THIS MONTH

4. Go To Process GENERATE STD SIGN NET DEMAND
1. For each ORDER-BROKEN in BROKEN ORDERS

1.1 Read MPS-NUMBER

1.2 Read ORDER-BROKEN-NUMBER
    BROKEN-ITEM-NUMBER
    BROKEN-ITEM-QUANT

1.3 Read ORDER-ITEM in APPROVED ORDERS with
    ORDER-NUMBER = ORDER-BROKEN-NUMBER and
    ITEM-NUMBER = ORDER-ITEM-NUMBER as key

1.4 Set ORDER-REC-NUMBER = ORDER-NUMBER
    ORDER-REC-ITEM = ITEM-NUMBER
    ORDER-REC-TYPE = TYPE
    ORDER-REC-SUBCODE = SUBCODE
    ORDER-REC-QUANT = ITEM-QUANT
    PROMISED-BY-MPS = BROKEN-ITEM-QUANT
    PROMISED-BY-ON-HAND = 0
    PROMISED-BY-PO = 0
    PROMISED-BY-PR = 0
    UNITS-DELIVERED = 0
    UNTIS-STILL-NOT-PROMISED = (ITEM-QUANT) -
                               (BROKEN-ITEM-QUANT)
    UNITS-REQ-ADJUSTMENT = 0

1.5 If UNTIS-STILL-NOT-PROMISED = 0 then
### PROCESS NARRATIVE

**Process Name:** ALLOCATE BROKEN ORDER  
**Process Number:** 1-2-3-1.PN  
**Page:** 02

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.1</td>
<td>Delete ORDER-ITEM in APPROVED ORDERS</td>
</tr>
</tbody>
</table>
| 1.6 | Set  
MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER  
MPS-ORDER-REC-ITEM = ORDER-REC-ITEM  
MPS-ORDER-REC-MPS = MPS-NUMBER  
MPS-ORDER-REC-UNITS = PROMISED-BY-MPS |
| 1.7 | Write MPS-ORDER-REC in MPS-ORDER |
| 1.8 | Write ORDER-REC in ORDER RECORDS |
| 1.9 | Delete ORDER-BROKEN in BROKEN ORDERS |
| 2. Go To Process | ALLOCATE ON HAND |
PROCESS NARRATIVE

Process Name: ALLOCATE PRODUCTION ORDER

Process Number: 1-2-3-3.PN

Page: 01

1. For each ORDER-APPROVED in APPROVED ORDERS
   1.1 For each ORDER-ITEM
      1.1.1 Read TYPE
           SUBCODE
      1.1.2 Read PROD-DESTINY in PRODUCTS using TYPE and SUBCODE as key
      1.1.3 If PROD-DESTINY = " Both " (Production and Purchase)
             or PROD-DESTINY = " Production " then
         1.1.3.1 Select ORDER-ITEM
   2. For all ORDER-ITEM selected
      2.1 Sort ORDER-ITEM by TYPE, SUBCODE and ORDER-PRIORITY
   3. For ORDER-ITEM sorted
      3.1 Select ORDER-ITEM with highest priority
      3.1.1 Read ORDER-REC in ORDER RECORDS using ORDER-NUMBER, and ITEM-NUMBER as key
      3.1.1.1 If can’t find ORDER-REC then
         3.1.1.1.1 Set ORDER-REC-NUMBER = ORDER-NUMBER
                         ORDER-REC-ITEM = ITEM-NUMBER
                         ORDER-REC-TYPE = TYPE
                         ORDER-REC-SUBCODE = SUBCODE
                         ORDER-REC-QUANT = ITEM-QUANT
                         PROMISED-BY-MPS = 0
PROCESS NARRATIVE

Process Name: ALLOCATE PRODUCTION ORDER

Process Number: 1-2-3-3.PN

Page: 02

PROMISED-BY-ON-HAND = 0
PROMISED-BY-PO = 0
PROMISED-BY-PR = 0
UNITS-DELIVERED = 0
UNITS-STILL-NOT-PROMISED = ITEM-QUANT
UNITS-REQ-ADJUSTMENT = 0

3.1.2 If PROD-TYPE = " Standard "
or UNITS-STILL-NOT-PROMISED = " 1 "

3.1.2.1 Then

3.1.2.1.1 Select PO-REC in PO RECORDS with
PO-AVAILABLE-TO-PROMISE > 0
using TYPE and SUBCODE as key

3.1.2.1.2 Sort PO-REC by PO-PLAN-MONTH ( The earlier the
PO is planned, the higher it is priority )

3.1.2.1.3 For each PO-REC sorted

3.1.2.1.3.1 Select first PO-REC

3.1.2.1.3.2 If ( PO-AVAILABLE-TO-PROMISE ) >=
( UNITS-STILL-NOT-PROMISED )

3.1.2.1.3.2.1 Then Set PROMISED-BY-PO =
UNITS-STILL-NOT-PROMISED
UNITS-STILL-NOT-PROMISED = 0
( PO-AVAILABLE-TO-PROMISE ) =
( PO-AVAILABLE-TO-PROMISE )
- ( UNITS-ALLOCATED )
### PROCESS NARRATIVE

**Process Name:** ALLOCATE PRODUCTION ORDER  
**Process Number:** 1-2-3-3.PN  
**Page:** 03

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.1.3.2.2 | Otherwise Set PROMISED-BY-PO = PO-AVAILABLE-TO-PROMISE  
| | PO-AVAILABLE-TO-PROMISE = 0  
| | ( UNITS-STILL-NOT-PROMISED ) = ( UNITS-STILL-NOT-PROMISED ) - ( UNITS-ALLOCATED ) |
| 3.1.2.1.3.3 | Set PO-ORDER-REC-ORDER = ORDER-REC-NUMBER  
| | PO-ORDER-REC-ITEM = ORDER-REC-ITEM  
| | PO-ORDER-REC-PO = PO-NUMBER  
| | PO-ORDER-REC-UNITS = PROMISED-BY-PO  
| | UNITS-ALLOCATED = 0 |
| 3.1.2.1.3.4 | Write PO-ORDER-REC in PO-ORDER |
| 3.1.2.1.3.5 | Write PO-REC in PO RECORDS |
| 3.1.2.1.3.6 | Delete PO-REC from sorted group |
| 3.1.2.2 | Otherwise ( Special and Interstate Signs with ( more than 1 sign been ordered ) |
| 3.1.2.2.1 | Select PO-REC in PO RECORDS with ( more than 1 sign been ordered ) using TYPE and SUBCODE as key |
| 3.1.2.2.2 | Sort PO-REC by PO-PLAN-MONTH ( The earlier the PO is planned, the higher is its priority ) |
| 3.1.2.2.3 | Select the PO-REC with highest priority |
### PROCESS NARRATIVE

**Process Name:** ALLOCATE PRODUCTION ORDER  
**Process Number:** 1-2-3-3.PN  
**Page:** 04

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.2.4 | Set \( \text{UNITS-AVAILABLE} = \text{PO-AVAILABLE-TO-PROMISE} \)  
|  | \( \text{PO-CONTROL-NUMBER} = \text{PO-REC-NUMBER} \)  
|  | \( I = 0 \)  
| 3.1.2.2.5 | Delete PO-REC from sorted group |
| 3.1.2.2.4 | Repeat the following until find  
|  | \( ( \text{UNITS-AVAILABLE} ) = \)  
|  | \( ( \text{UNITS-STILL-NOT-PROMISED} ) \)  
|  | or until there is no more PO-REC in the sorted group |
| 3.1.2.2.4.1 | Select the PO-REC with highest priority |
| 3.1.2.2.4.2 | Set \( \text{UNITS-AVAILABLE} = \text{UNITS-AVAILABLE} + \)  
|  | \( \text{PO-AVAILABLE-TO-PROMISE} \) |
| 3.1.2.2.4.3 | Set \( \text{PO-TO-DELETE}(I) = \text{MPS-REC-NUMBER} \)  
|  | \( I = I + 1 \)  
| 3.1.2.2.4.4 | Delete PO-REC from sorted group |
| 3.1.2.2.5 | If \( \text{UNITS-AVAILABLE} = \text{UNITS-STILL-NOT-PROMISED} \) |
| 3.1.2.2.5.1 | Then |
| 3.1.2.2.5.1.1 | Set \( \text{PO-REC-NUMBER} = \text{PO-CONTROL-NUMBER} \)  
|  | \( \text{PO-REC-UNITS} = \text{UNITS-AVAILABLE} \)  
|  | \( \text{PO-AVAILABLE-TO-PROMISE} = 0 \) |
| 3.1.2.2.5.1.2 | Set \( \text{PROMISED-BY-PO} = \)  
|  | \( \text{PROMISED-BY-PO} + \text{UNITS-STILL-NOT-PROMISED} \)  
|  | \( \text{PO-ORDER-REC-ORDER} = \text{ORDER-REC-NUMBER} \) |
PROCESS NARRATIVE

Process Name: ALLOCATE PRODUCTION ORDER

Process Number: 1-2-3-3.PN

Page: 05

PO-ORDER-REC-ITEM = ORDER-REC-ITEM
PO-ORDER-REC-PO = PO-NUMBER
PO-ORDER-REC-UNITS = UNITS-STILL-NOT-PROMISED

UNITS-STILL-NOT-PROMISED = 0

3.1.2.2.5.1.3 Write PO-REC in PO RECORDS
3.1.2.2.5.1.4 Delete PO-REC in PO RECORDS using all PO-TO-DELETE as key
3.1.2.2.5.1.5 Write PO-ORDER-REC in PO-ORDER

3.2 Write ORDER-REC in ORDER RECORDS

3.3 If UNITS-STILL-NOT-PROMISED = 0 then
3.3.1 Delete ORDER-ITEM in APPROVED-ORDERS
3.4 Delete ORDER-ITEM in sorted group

4. Go To Process ALLOCATE PURCHASE REQUIREMENT
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>For each ORDER-APPROVED in APPROVED ORDERS</td>
</tr>
<tr>
<td>1.1</td>
<td>For each ORDER-ITEM</td>
</tr>
<tr>
<td>1.1.1</td>
<td>Read TYPE SUBCODE</td>
</tr>
<tr>
<td>1.1.2</td>
<td>Read PROD-DESTINY in PRODUCTS using TYPE and SUBCODE as key</td>
</tr>
<tr>
<td>1.1.3</td>
<td>If PROD-DESTINY = &quot;Both&quot; (Production and Purchase) or PROD-DESTINY = &quot;Production&quot; then</td>
</tr>
<tr>
<td>1.1.3.1</td>
<td>Select ORDER-ITEM</td>
</tr>
<tr>
<td>2.</td>
<td>For all ORDER-ITEM selected</td>
</tr>
<tr>
<td>2.1</td>
<td>Sort ORDER-ITEM by TYPE, SUBCODE and ORDER-PRIORITY</td>
</tr>
<tr>
<td>3.</td>
<td>For ORDER-ITEM sorted</td>
</tr>
<tr>
<td>3.1</td>
<td>Select ORDER-ITEM with highest priority</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Read ORDER-REC in ORDER RECORDS using ORDER-NUMBER, and ITEM-NUMBER as key</td>
</tr>
<tr>
<td>3.1.1.1</td>
<td>If can't find ORDER-REC then</td>
</tr>
<tr>
<td>3.1.1.1.1</td>
<td>Set ORDER-REC-NUMBER = ORDER-NUMBER</td>
</tr>
<tr>
<td></td>
<td>ORDER-REC-ITEM = ITEM-NUMBER</td>
</tr>
<tr>
<td></td>
<td>ORDER-REC-TYPE = TYPE</td>
</tr>
<tr>
<td></td>
<td>ORDER-REC-SUBCODE = SUBCODE</td>
</tr>
<tr>
<td></td>
<td>ORDER-REC-QUANT = ITEM-QUANT</td>
</tr>
<tr>
<td></td>
<td>PROMISED-BY-MPS = 0</td>
</tr>
</tbody>
</table>
### PROCEDURE NARRATIVE

**Process Name:** ALLOCATE PRODUCTION ORDER  
**Process Number:** 1-2-3-3.PN  
**Page:** 02

<table>
<thead>
<tr>
<th>PROMISED-BY-ON-HAND</th>
<th>=</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROMISED-BY-PO</td>
<td>=</td>
<td>0</td>
</tr>
<tr>
<td>PROMISED-BY-PR</td>
<td>=</td>
<td>0</td>
</tr>
<tr>
<td>UNITS-DELIVERED</td>
<td>=</td>
<td>0</td>
</tr>
<tr>
<td>UNITS-STILL-NOT-PROMISED</td>
<td>=</td>
<td>ITEM-QUANT</td>
</tr>
<tr>
<td>UNITS-REQ-ADJUSTMENT</td>
<td>=</td>
<td>0</td>
</tr>
</tbody>
</table>

**3.1.2** If PROD-TYPE = "Standard"  
or UNITS-STILL-NOT-PROMISED = "1"

**3.1.2.1** Then

**3.1.2.1.1** Select PO-REC in PO RECORDS with  
PO-AVAILABLE-TO-PROMISE > 0  
using TYPE and SUBCODE as key

**3.1.2.1.2** Sort PO-REC by PO-PLAN-MONTH (The earlier the  
PO is planned, the higher it is priority)

**3.1.2.1.3** For each PO-REC sorted

**3.1.2.1.3.1** Select first PO-REC

**3.1.2.1.3.2** If (PO-AVAILABLE-TO-PROMISE) >=  
(UNITS-STILL-NOT-PROMISED)

**3.1.2.1.3.2.1** Then Set PROMISED-BY-PO =  
UNITS-REQ-ADJUSTMENT

UNITS-STILL-NOT-PROMISED = 0

(PO-AVAILABLE-TO-PROMISE) =  
(PO-AVAILABLE-TO-PROMISE)  
- (UNITS-ALLOCATED)
### PROCESS NARRATIVE

**Process Name:** ALLOCATE PRODUCTION ORDER  
**Process Number:** 1-2-3-3.PN  
**Page:** 03

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.1.3.2.2 | Otherwise Set PROMISED-BY-PO = PO-AVAILABLE-TO-PROMISE  

\[
PO-AVAILABLE-TO-PROMISE = 0
\]

\[
(\text{UNITS-STILL-NOT-PROMISED}) = (\text{UNITS-STILL-NOT-PROMISED}) - (\text{UNITS-ALLOCATED})
\]

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.1.3.3 | Set PO-ORDER-REC-ORDER = ORDER-REC-NUMBER  
PO-ORDER-REC-ITEM = ORDER-REC-ITEM  
PO-ORDER-REC-PO = PO-NUMBER  
PO-ORDER-REC-UNITS = PROMISED-BY-PO  
UNITS-ALLOCATED = 0

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.1.3.4 | Write PO-ORDER-REC in PO-ORDER

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.1.3.5 | Write PO-REC in PO RECORDS

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.1.3.6 | Delete PO-REC from sorted group

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.2 | Otherwise (Special and Interstate Signs with more than 1 sign been ordered)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.2.1 | Select PO-REC in PO RECORDS with PO-AVAILABLE-TO-PROMISE > 0 using TYPE and SUBCODE as key

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.2.2 | Sort PO-REC by PO-PLAN-MONTH (The earlier the PO is planned, the higher is its priority)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.2.2.3 | Select the PO-REC with highest priority
3.1.2.2.4 Set UNITS-AVAILABLE = PO-AVAILABLE-TO-PROMISE
   PO-CONTROL-NUMBER = PO-REC-NUMBER
   I = 0

3.1.2.2.5 Delete PO-REC from sorted group

3.1.2.2.4 Repeat the following until find
   ( UNITS-AVAILABLE ) =
   ( UNITS-STILL-NOT-PROMISED )
   or until there is no more PO-REC in the
   sorted group

3.1.2.2.4.1 Select the PO-REC with highest priority

3.1.2.2.4.2 Set UNITS-AVAILABLE = UNITS-AVAILABLE +
   PO-AVAILABLE-TO-PROMISE

3.1.2.2.4.3 Set PO-TO-DELETE(I) = MPS-REC-NUMBER
   I = I + 1

3.1.2.2.4.4 Delete PO-REC from sorted group

3.1.2.2.5 If UNITS-AVAILABLE = UNITS-STILL-NOT-PROMISED

3.1.2.2.5.1 Then

3.1.2.2.5.1.1 Set PO-REC-NUMBER = PO-CONTROL-NUMBER
   PO-REC-UNITS = UNITS-AVAILABLE
   PO-AVAILABLE-TO-PROMISE = 0

3.1.2.2.5.1.2 Set PROMISED-BY-PO =
   PROMISED-BY-PO + UNITS-STILL-NOT-PROMISED
   PO-ORDER-REC-ORDER = ORDER-REC-NUMBER
PROCESS NARRATIVE

Process Name: ALLOCATE PRODUCTION ORDER

Process Number: 1-2-3-3.PN

Page: 05

\[
\begin{align*}
\text{PO-ORDER-REC-ITEM} &= \text{ORDER-REC-ITEM} \\
\text{PO-ORDER-REC-PO} &= \text{PO-NUMBER} \\
\text{PO-ORDER-REC-UNITS} &= \text{UNITS-STILL-NOT-PROMISED} \\
\text{UNITS-STILL-NOT-PROMISED} &= 0
\end{align*}
\]

3.1.2.2.5.1.3 Write \text{PO-REC} in \text{PO RECORDS}

3.1.2.2.5.1.4 Delete \text{PO-REC} in \text{PO RECORDS} using all \text{PO-TO-DELETE} as key

3.1.2.2.5.1.5 Write \text{PO-ORDER-REC} in \text{PO-ORDER}

3.2 Write \text{ORDER-REC} in \text{ORDER RECORDS}

3.3 If \text{UNITS-STILL-NOT-PROMISED} = 0 then

3.3.1 Delete \text{ORDER-ITEM} in \text{APPROVED-ORDERS}

3.4 Delete \text{ORDER-ITEM} in sorted group

4. Go To Process ALLOCATE PURCHASE REQUIREMENT
# PROCESS NARRATIVE

**Process Name:** ALLOCATE PURCHASE REQUIREMENT  
**Process Number:** 1-2-3-4.PN  
**Page:** 01

1. For each ORDER-APPROVED in APPROVED ORDERS
   1.1 For each ORDER-ITEM
      1.1.1 Read TYPE  
          SUBCODE
      1.1.2 Read PROD-REC in PRODUCTS using TYPE and SUBCODE  
          as key
      1.1.3 If PROD-DESTINY = " Both " (Production and Purchase)  
          or PROD-DESTINY = " Purchase " then
         1.1.3.1 Select ORDER-ITEM
   2. For all ORDER-ITEM selected
      2.1 Sort ORDER-ITEM by TYPE, SUBCODE and ORDER-PRIORITY
   3. For ORDER-ITEM sorted
      3.1 Select ORDER-ITEM with highest priority
         3.1.1 Read ORDER-REC in ORDER RECORDS using ORDER-NUMBER,  
             and ITEM-NUMBER as key
         3.1.1.1 If can’t find ORDER-REC then
            3.1.1.1.1 Set ORDER-REC-NUMBER = ORDER-NUMBER  
                ORDER-REC-ITEM = ITEM-NUMBER  
                ORDER-REC-TYPE = TYPE  
                ORDER-REC-SUBCODE = SUBCODE  
                ORDER-REC-QUANT = ITEM-QUANT  
                PROMISED-BY-MPS = 0
### PROCESS NARRATIVE

**Process Name:** ALLOCATE PURCHASE REQUIREMENT

**Process Number:** 1-2-3-4.PN

#### Promised-By-On-Hand

- PROMISED-BY-ON-HAND = 0
- PROMISED-BY-PO = 0
- PROMISED-BY-PR = 0
- UNITS-DELIVERED = 0
- UNITS-STILL-NOT-PROMISED = ITEM-QUANT
- UNITS-REQ-ADJUSTMENT = 0

#### 3.1.2 Select PR-REC in PR RECORDS with

- PR-AVAILABLE-TO-PROMISE > 0

  using TYPE and SUBCODE as key

#### 3.1.3 Sort PR-REC by PR-PLAN-MONTH (The earlier the PR is planned, the higher is its priority)

#### 3.1.4 For each PR-REC sorted

3.1.4.1 Select first PR-REC

3.1.4.2 If (PR-AVAILABLE-TO-PROMISE) >= (UNITS-STILL-NOT-PROMISED)

3.1.4.2.1 Then Set

- PROMISED-BY-PR = UNITS-STILL-NOT-PROMISED
- UNITS-STILL-NOT-PROMISED = 0

  (PR-AVAILABLE-TO-PROMISE) =
  (PR-AVAILABLE-TO-PROMISE) - (UNITS-ALLOCATED)

3.1.4.2.2 Otherwise Set

- PROMISED-BY-PR = PR-AVAILABLE-TO-PROMISE

  PR-AVAILABLE-TO-PROMISE = 0
( UNITS-STILL-NOT-PROMISED ) =
( UNITS-STILL-NOT-PROMISED ) -
( UNITS-ALLOCATED )

3.1.4.3 Set PR-ORDER-REC-ORDER = ORDER-REC-NUMBER
    PR-ORDER-REC-ITEM = ORDER-REC-ITEM
    PR-ORDER-REC-PR = PR-NUMBER
    PR-ORDER-REC-UNITS = PROMISED-BY-PR
    UNITS-ALLOCATED = 0

3.1.4.4 Write PR-ORDER-REC in PR-ORDER
3.1.4.5 Write PR-REC in PR RECORDS
3.1.4.6 Delete PR-REC from sorted group
3.2 Write ORDER-REC in ORDER RECORDS
3.3 If UNITS-STILL-NOT-PROMISED = 0 then
3.3.1 Delete ORDER-ITEM in APPROVED ORDERS
3.4 Delete ORDER-ITEM from sorted group
4. Go To Process ALLOCATE MPS.
PROCESS NARRATIVE

Process Name: ALLOCATE MPS

Process Number: 1-2-3-5.PN

1. For each ORDER-APPROVED in APPROVED ORDERS
   1.1 For all ORDER-ITEM
      1.1.1 Sort ORDER-ITEM by TYPE, SUBCODE and ORDER-PRIORITY
   2. For all ORDER-ITEM sorted
      2.1 Select ORDER-ITEM with highest priority in the group
      2.1.1 Read ORDER-REC in ORDER RECORDS using ORDER-NUMBER, and ITEM-NUMBER as key
      2.1.1.1 If can't find ORDER-REC then
      2.1.1.1 Set ORDER-REC-NUMBER = ORDER-NUMBER
                  ORDER-REC-ITEM = ITEM-NUMBER
                  ORDER-REC-TYPE = TYPE
                  ORDER-REC-SUBCODE = SUBCODE
                  ORDER-REC-QUANT = ITEM-QUANT
                  PROMISED-BY-MPS = 0
                  PROMISED-BY-ON-HAND = 0
                  PROMISED-BY-PO = 0
                  PROMISED-BY-PR = 0
                  UNITS-DELIVERED = 0
                  UNITS-STILL-NOT-PROMISED = ITEM-QUANT
                  UNITS-REQ-ADJUSTMENT = 0
      2.1.2 Read PROD-REC in PRODUCTS using TYPE and SUBCODE as key
      2.1.3 If PROD-TYPE = " Standard "
             or UNITS-STILL-NOT-PROMISED = " 1 "
      2.1.3.1 Then
PROCESS NARRATIVE

Process Name: ALLOCATE MPS

Process Number: 1-2-3-5.PN

2.1.3.1.1 Select MPS-REC in MPS RECORDS with
MPS-AVAILABLE-TO-PROMISE > 0
using TYPE and SUBCODE as key

2.1.3.1.2 Sort MPS-REC by MPS-PLAN-MONTH (The earlier the
MPS is planned, the higher is its priority)

2.1.3.1.3 For each MPS-REC sorted

2.1.3.1.3.1 Select the MPS-REC with highest priority

2.1.3.1.3.2 If (MPS-AVAILABLE-TO-PROMISE) >=
(UNITS-STILL-NOT-PROMISED)

2.1.3.1.3.2.1 Then Set PROMISED-BY-MPS =
PROMISED-BY-MPS +
UNITS-STILL-NOT-PROMISED

UNITS-ALLOCATED =
UNITS-STILL-NOT-PROMISED

UNITS-STILL-NOT-PROMISED = 0

(MPS-AVAILABLE-TO-PROMISE) =
(MPS-AVAILABLE-TO-PROMISE)
- (UNITS-ALLOCATED)

2.1.3.1.3.2.2 Otherwise Set PROMISED-BY-MPS =
PROMISED-BY-MPS +
MPS-AVAILABLE-TO-PROMISE

UNITS-ALLOCATED =
MPS-AVAILABLE-TO-PROMISE
PROCESS NARRATIVE

Process Name: ALLOCATE MPS

MPS-AVAILABLE-TO-PROMISE = 0

( UNITS-STILL-NOT-PROMISED ) =

( UNITS-STILL-NOT-PROMISED )

- ( UNITS-ALLOCATED )

2.1.3.1.3.3 Set MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER
MPS-ORDER-REC-ITEM = ORDER-REC-ITEM
MPS-ORDER-REC- MPS = MPS-NUMBER
MPS-ORDER-REC-UNITS = UNITS-ALLOCATED
UNITS-ALLOCATED = 0

2.1.3.1.3.4 Write MPS-ORDER-REC in MPS-ORDER

2.1.3.1.3.5 Write MPS-REC in MPS RECORDS

2.1.3.1.3.6 Delete MPS-REC from sorted group

2.1.3.1.4 If UNITS-STILL-NOT-PROMISED > 0

2.1.3.1.4.1 Then

2.1.3.1.4.1.1 Call Process GENERATE REQ MPS giving
MPS-REQ-TYPE = TYPE
MPS-REQ-SUBCODE = SUBCODE
MPS-REQ-UNITS = UNITS-STILL-NOT-PROMISED

2.1.3.1.4.1.2 Read MPS-NUMBER from Process
GENERATE REQ MPS

2.1.3.1.4.1.3 Read MPS-REC in MPS RECORDS using
MPS-NUMBER as key
Process Name: ALLOCATE MPS

2.1.3.1.4.1.4 Set PROMISED-BY-MPS = PROMISED-BY-MPS + UNITS-STILL-NOT-PROMISED

UNITS-ALLOCATED = UNITS-STILL-NOT-PROMISED

UNITS-STILL-NOT-PROMISED = 0

(MPS-AVAILABLE-TO-PROMISE) =
  (MPS-AVAILABLE-TO-PROMISE) - (UNITS-ALLOCATED)

2.1.3.1.4.1.5 Set MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER
MPS-ORDER-REC-ITEM = ORDER-REC-ITEM
MPS-ORDER-REC-MPS = MPS-NUMBER
MPS-ORDER-REC-UNITS = UNITS-ALLOCATED
UNITS-ALLOCATED = 0

2.1.3.1.4.1.6 Write MPS-ORDER-REC in MPS-ORDER

2.1.3.1.4.1.7 Write MPS-REC in MPS RECORDS

2.1.3.2 Otherwise (Special and Interstate Signs with
  more than 1 sign been ordered)

2.1.3.2.1 Select MPS-REC in MPS RECORDS with
  MPS-AVAILABLE-TO-PROMISE > 0
  using TYPE and SUBCODE as key

2.1.3.2.2 Sort MPS-REC by MPS-PLAN-MONTH (The earlier the
  MPS is planned, the higher is its priority)

2.1.3.2.3 Select the MPS-REC with highest priority
PROCESS NARRATIVE

Process Name: ALLOCATE MPS

Process Number: 1-2-3-5.PN

2.1.3.2.4 Set UNITS-AVAILABLE = MPS-AVAILABLE-TO-PROMISE
MPS-CONTROL-NUMBER = MPS-REC-NUMBER
I = 0

2.1.3.2.5 Delete MPS-REC from sorted group

2.1.3.2.4 Repeat the following until find
( UNITS-AVAILABLE ) =
( UNITS-STILL-NOT-PROMISED )
or until there is no more MPS-REC in the
sorted group

2.1.3.2.4.1 Select the MPS-REC with highest priority

2.1.3.2.4.2 Set UNITS-AVAILABLE = UNITS-AVAILABLE +
MPS-AVAILABLE-TO-PROMISE

2.1.3.2.4.3 Set MPS-TO-DELETE(I) = MPS-REC-NUMBER
I = I + 1

2.1.3.2.4.4 Delete MPS-REC from sorted group

2.1.3.2.5 If UNITS-AVAILABLE = UNITS-STILL-NOT-PROMISED

2.1.3.2.5.1 Then

2.1.3.2.5.1.1 Set MPS-REC-NUMBER = MPS-CONTROL-NUMBER
MPS-REC-UNITS = UNITS-AVAILABLE
MPS-AVAILABLE-TO-PROMISE = 0

2.1.3.2.5.1.2 Set PROMISED-BY-MPS =
PROMISED-BY-MPS + UNITS-STILL-NOT-PROMISED
MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER
PROCESS NARRATIVE

Process Name: ALLOCATE MPS

MPS-ORDER-REC-ITEM = ORDER-REC-ITEM
MPS-ORDER-REC-MPS = MPS-NUMBER
MPS-ORDER-REC-UNITS = UNITS-STILL-NOT-PROMISED

UNITS-STILL-NOT-PROMISED = 0

2.1.3.2.5.1.3 Write MPS-REC in MPS RECORDS
2.1.3.2.5.1.4 Delete MPS-REC in MPS RECORDS using all
MPS-TO-DELETE as key
2.1.3.2.5.1.5 Write MPS-ORDER-REC in MPS-ORDER
2.1.3.2.5.2 Otherwise ( Could not find enough available in
MPS then generate MPS )
2.1.3.2.5.2.1 Call Process GENERATE REQ MPS giving
MPS-REQ-TYPE = TYPE
MPS-REQ-SUBCODE = SUBCODE
MPS-REQ-UNITS = UNITS-STILL-NOT-PROMISED
2.1.3.2.5.2.2 Read MPS-NUMBER from Process
GENERATE REQ MPS
2.1.3.2.5.2.3 Read MPS-REC in MPS RECORDS using
MPS-NUMBER as key
2.1.3.2.5.2.4 Set PROMISED-BY-MPS = PROMISED-BY-MPS +
UNITS-STILL-NOT-PROMISED
### PROCESS NARRATIVE

**Process Name**: ALLOCATE MPS  
**Process Number**: 1-2-3-5.PN  
**Page**: 07

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{UNITS-ALLOCATED} = \text{UNITS-STILL-NOT-PROMISED} )</td>
<td></td>
</tr>
<tr>
<td>( \text{UNITS-STILL-NOT-PROMISED} = 0 )</td>
<td></td>
</tr>
<tr>
<td>( (\text{MPS-AVAILABLE-TO-PROMISE}) = (\text{MPS-AVAILABLE-TO-PROMISE}) - (\text{UNITS-ALLOCATED}) )</td>
<td></td>
</tr>
</tbody>
</table>

2.1.3.2.5.2.5 Set
- \( \text{MPS-ORDER-REC-ORDER} = \text{ORDER-REC-NUMBER} \)
- \( \text{MPS-ORDER-REC-ITEM} = \text{ORDER-REC-ITEM} \)
- \( \text{MPS-ORDER-REC-MPS} = \text{MPS-NUMBER} \)
- \( \text{MPS-ORDER-REC-UNITS} = \text{UNITS-ALLOCATED} \)
- \( \text{UNITS-ALLOCATED} = 0 \)

2.1.3.2.5.2.6 Write \( \text{MPS-ORDER-REC} \) in \( \text{MPS-ORDER} \)

2.1.3.2.5.2.7 Write \( \text{MPS-REC} \) in \( \text{MPS RECORDS} \)

2.2 Write \( \text{ORDER-REC} \) in \( \text{ORDER RECORDS} \)

2.3 Delete \( \text{ORDER-ITEM} \) in \( \text{APPROVED-ORDERS} \)

2.4 Delete \( \text{ORDER-ITEM} \) in sorted group
1. Read WEEK-DELIVERY
2. Read DELIVERY-REC in DELIVERY using WEEK-DELIVERY as key
   2.1 For each DISTRICT
      2.1.1 Write DISTRICT-TO-DELIVER
3. Read user option
   3.1 If user option is "Enter new districts to deliver"
      3.1.1 Then
      3.1.1.1 Read DELIVERY-DISTRICTS
      3.1.1.1.1 For each DISTRICT in DELIVERY-DISTRICTS
      3.1.1.1.1.1 Set DISTRICT-TO-DELIVER = DISTRICT
4. For each DISTRICT-TO-DELIVER
   4.1 Read all ON-HAND-ORDER-REC with
      DISTRICT in ORDER-NUMBER = DISTRICT-TO-DELIVER
   4.1.1 For each ON-HAND-ORDER-REC
      4.1.1.1 Read PROD-REC in PRODUCTS using TYPE and
             SUBCODE as key
## PROCESS NARRATIVE

**Process Name:** GENERATE DELIVERY LIST  
**Process Number:** 1-3-1.PN  
**Page:** 02

4.1.1.2 Set

<table>
<thead>
<tr>
<th>DELIVERY-TYPE</th>
<th>ON-HAND-ORDER-TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELIVERY-SUBCODE</td>
<td>ON-HAND-ORDER-REC-SUBCODE</td>
</tr>
<tr>
<td>DELIVERY-DESCRIPTION</td>
<td>DESCRIPTION in PROD-REC</td>
</tr>
<tr>
<td>DELIVERY-UNITS</td>
<td>ON-HAND-ORDER-REC-UNITS</td>
</tr>
<tr>
<td>DELIVERY-ORDER</td>
<td>ON-HAND-ORDER-REC-ORDER</td>
</tr>
<tr>
<td>DELIVERY-ITEM</td>
<td>ON-HAND-ORDER-REC-ITEM</td>
</tr>
</tbody>
</table>

4.2 Write DELIVERY-LIST
1. For each ORDER-DELIVERED entered

1.1 Read ON-HAND-ORDER-REC using
    ON-HAND-ORDER-REC-ORDER = DELIVERED-ORDER
    ON-HAND-ORDER-REC-ITEM = DELIVERED-ITEM as key

1.2 Read ORDER-REC using
    ORDER-REC-NUMBER = DELIVERED-ORDER
    ORDER-REC-ITEM = DELIVERED-ITEM

1.3 Read ON-HAND-REC using
    ON-HAND-REC-TYPE = ON-HAND-REC-TYPE
    ON-HAND-REC-SUBCODE = ON-HAND-REC-SUBCODE
    as key

1.4 Set UNITS-DELIVERED = UNITS-DELIVERED
    + DELIVERED-UNITS
    ON-HAND-UNITS = ON-HAND-UNITS
    - DELIVERED-UNITS

1.5 If DELIVERED-UNITS = ON-HAND-ORDER-REC-UNITS

1.5.1 Then

1.5.1.1 Set PROMISED-BY-ON-HAND = 0

1.5.1.2 Delete ON-HAND-ORDER-REC in ON HAN ORDER using
    ON-HAND-ORDER-REC-ORDER and
    ON-HAND-ORDER-REC-ITEM as key

1.5.2 Otherwise ( DELIVERED-UNITS not equal
    ON-HAND-ORDER-REC-UNITS )
PROCESS NARRATIVE

Process Name: UPDATE ORDER DELIVERED

Process Number: 1-3-2.PN Page: 02

1.5.2.1 If DELIVERED-UNITS < ON-HAND-ORDER-REC-UNITS

1.5.2.1.1 Then

1.5.2.1.1.1 Set PROMISED-BY-ON-HAND = PROMISED-BY-ON-HAND
- DELIVERED-UNITS

ON-HAND-ORDER-REC-UNITS =
ON-HAND-ORDER-REC-UNITS
- DELIVERED-UNITS

1.5.2.1.1.2 Write ON-HAND-ORDER-REC in ON HAND-ORDER using
ON-HAND-ORDER-REC-ORDER and
ON-HAND-ORDER-REC-ITEM as key

1.5.2.1.2 Otherwise ( DELIVERED-UNITS greater than
ON-HAND-ORDER-REC-UNITS )

1.5.2.1.2.1 If ON-HAND-AVAILABLE-TO-PROMISE > 0

1.5.2.1.2.1.1 Then

1.5.2.1.2.1.1.1 If ON-HAND-AVAILABLE-TO-PROMISE >
( ON-HAND-ORDER-REC-UNITS -
DELIVERED-UNITS )

1.5.2.1.2.1.1.1.1 Then

1.5.2.1.2.1.1.1.1.1 Set ON-HAND-AVAILABLE-TO-PROMISE =
( ON-HAND-AVAILABLE-TO-PROMISE )
- ( ON-HAND-ORDER-REC-UNITS )
+ ( DELIVERED-UNITS )

PROMISED-BY-ON-HAND = 0
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.2.1.2.1.1.1.1.2</td>
<td>Delete ON-HAND-ORDER-REC in ON HAND-ORDER using ON-HAND-ORDER-REC-ORDER and ON-HAND-ORDER-REC-ITEM as key</td>
</tr>
<tr>
<td>1.5.2.1.2.1.1.1.2 Otherwise</td>
<td>(Not enough available to promise to compensate the difference)</td>
</tr>
<tr>
<td>1.5.2.1.2.1.1.1.2.1</td>
<td>Set UNITS-REQ-ADJUSTMENT = (UNITS-REQ-ADJUSTMENT) - (ON-HAND-AVAILABLE-TO-PROMISE) - (ON-HAND-ORDER-REC-UNITS) + (DELIVERED-UNITS)</td>
</tr>
<tr>
<td>1.5.2.1.2.1.1.1.2.2</td>
<td>Delete ON-HAND-ORDER-REC in ON HAND-ORDER using ON-HAND-ORDER-REC-ORDER and ON-HAND-ORDER-REC-ITEM as key</td>
</tr>
<tr>
<td>1.6</td>
<td>Write ORDER-REC in ORDER-RECORDS</td>
</tr>
<tr>
<td>1.7</td>
<td>Write ON-HAND-REC in ON HAND RECORDS</td>
</tr>
</tbody>
</table>
1. Read FIRST-MONTH = THIS-MONTH from THIS MONTH

1.1 Read DEMAND-FORECAST in FORECASTING SYSTEM with MONTH = (FIRST-MONTH)
or MONTH = (FIRST-MONTH + 1)
or MONTH = (FIRST-MONTH + 2)
or MONTH = (FIRST-MONTH + 3)

1.1.1 For each group of DEMAND-FORECAST with same TYPE, SUBCODE and DISTRICT

1.1.1.1 Read PROD-REC in PRODUCTS using TYPE and SUBCODE as key

1.1.1.1.1 If PROD-TYPE = "Standard" then

1.1.1.1.1.1 Read ON-HAND in INVENTORY SYSTEM using TYPE, SUBCODE and DISTRICT as key

1.1.1.1.2 Read ORDER-REC in ORDER REcORS with ORDER-REC-TYPE = TYPE
ORDER-REC-SUBCODE = SUBCODE
DISTRICT in ORDER-REC-NUMBER = DISTRICT

1.1.1.1.2.1 If PROMISED-BY-MPS > 0 Then

1.1.1.1.2.1.1 Read MPS-ORDER-REC in MPS-ORDER with MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER
MPS-ORDER-REC-ITEM = ORDER-REC-ITEM

1.1.1.1.2.1.1.1 For each MPS-ORDER-REC

1.1.1.1.2.1.1.1.1 Read MPS-REC in MPS REcORS using MPS-REC-NUMBER = MPS-ORDER-REC-MPS
1.1.1.1.2.1.1.1.1.1 Set SCHEDULE-RECEIPT =
SCHEDULE-RECEIPT + MPS-ORDER-REC-UNITS
using MONTH = MPS-PLAN-MONTH as index

1.1.1.1.1.2.2 If PROMISED-BY-ON-HAND > 0 Then

1.1.1.1.1.2.2.1 Read ON-HAND-ORDER-REC in ON HAND-ORDER
with ON-HAND-ORDER-REC-ORDER = ORDER-REC-NUMBER
and ON-HAND-ORDER-REC-ITEM = ORDER-REC-ITEM

1.1.1.1.1.2.2.1.1 Set SCHEDULE-RECEIPT =
SCHEDULE-RECEIPT + ON-HAND-ORDER-REC-UNITS
using MONTH = FIRST-MONTH as index

1.1.1.1.1.2.3 If PROMISED-BY-PO > 0 Then

1.1.1.1.1.2.3.1 Read PO-ORDER-REC in PO-ORDER with
PO-ORDER-REC-ORDER = ORDER-REC-NUMBER
PO-ORDER-REC-ITEM = ORDER-REC-ITEM

1.1.1.1.1.2.3.1.1 For each PO-ORDER-REC

1.1.1.1.1.2.3.1.1.1 Read PO-REC in PO RECORDS using
PO-REC-NUMBER = PO-ORDER-REC-PO
as key

1.1.1.1.1.2.3.1.1.1.1 Set SCHEDULE-RECEIPT =
SCHEDULE-RECEIPT + PO-ORDER-REC-UNITS
using MONTH = PO-PLAN-MONTH as index

1.1.1.1.1.2.3 If PROMISED-BY-PR > 0 Then
**PROCESS NARRATIVE**

Process Name: GENERATE STD SIGN NET DEMAND

<table>
<thead>
<tr>
<th>Process Number: 2-1-1-1.PN</th>
<th>Page: 03</th>
</tr>
</thead>
</table>

1.1.1.1.1.2.3.1 Read PR-ORDER-REC in PR-ORDER with
   PR-ORDER-REC-ORDER = ORDER-REC-NUMBER
   PR-ORDER-REC-ITEM = ORDER-REC-ITEM

1.1.1.1.1.2.3.1.1 For each PR-ORDER-REC

1.1.1.1.1.2.3.1.1.1 Read PR-REC in PR RECORDS using
   PR-REC-NUMBER = PR-ORDER-REC-PR
   as key

1.1.1.1.1.2.3.1.1.1.1 Set SCHEDULE-RECEIPT =
   SCHEDULE-RECEIPT + PR-ORDER-REC-UNITS
   using MONTH = PR-PLAN-MONTH as index

1.1.1.1.1.3 Set MONTH = FIRST-MONTH
   PLANNED-INVENTORY( FIRST-MONTH ) =
   ON-HAND-QUANT
   AVAIL-TO-PROM = 0

1.1.1.1.4 Repeat the following 4 times

1.1.1.1.4.1 Set PLANNED-INVENTORY( MONTH ) =
   ( PLANNED-INVENTORY( MONTH ) ) +
   ( SCHEDULE-RECEIPT( MONTH ) )
   AVAILABLE( MONTH ) =
   ( PLANNED-INVENTORY( MONTH ) )
   - ( DEMAND-FORECAST( MONTH ) )

1.1.1.1.4.1.1 If AVAILABLE( MONTH ) < LOW-LEVEL

1.1.1.1.4.1.1.1 Then
PROCESS NARRATIVE

Process Name: GENERATE STD SIGN NET DEMAND

1. If ( AVAILABLE( MONTH ) + AVAIL-TO-PROM ) < LOW-LEVEL

Then Set NET-UNITS( MONTH ) =
( HIGH-LEVEL )
- ( AVAILABLE ( MONTH ))
- ( AVAIL-TO-PROM )

AVAIL-TO-PROM = AVAIL-TO-PROM
+ AVAILABLE( MONTH )
+ NET-UNITS( MONTH ))

PLANNED-INVENTORY( MONTH + 1 ) = 0

Otherwise Set NET-UNITS( MONTH ) = 0

PLANNED-INVENTORY( MONTH + 1 ) = 0

AVAIL-TO-PROM = AVAIL-TO-PROM
+ AVAILABLE( MONTH )

Otherwise Set NET-UNITS( MONTH ) = 0

PLANNED-INVENTORY( MONTH + 1 ) = AVAILABLE ( MONTH )

Set MONTH = MONTH + 1

Write NET-DEMAND in NET DEMAND
using TYPE, SUBCODE, DISTRICT and MONTH as key

2. Go To Process GENERATE SPECIAL SIGN NET DEMAND
PROCESS NARRATIVE

Process Name: GENERATE SPECIAL SIGN NET DEMAND

Process Number: 2-1-1-2.PN

1. Read FIRST-MONTH = THIS-MONTH from THIS MONTH

1.1 Read DEMAND-FORECAST in FORECASTING SYSTEM
   with MONTH = ( FIRST-MONTH )
   or MONTH = ( FIRST-MONTH + 1 )
   or MONTH = ( FIRST-MONTH + 2 )
   or MONTH = ( FIRST-MONTH + 3 )

1.1.1 For each group of DEMAND-FORECAST with same TYPE, SUBCODE and DISTRICT

1.1.1.1 Read PROD-REC in PRODUCTS using TYPE and SUBCODE as key

1.1.1.1.1 If PROD-TYPE = " Special " or " Interstate "

   then

1.1.1.1.1.1 Set NET-UNITS( MONTH ) =
   DEMAND-FORECAST( MONTH )

1.1.1.1.2 Write NET-DEMAND in NET DEMAND using TYPE and SUBCODE as key.

2. Go To Process CHANGE NET DEMAND
PROCESS NARRATIVE

Process Name: CHANGE NET DEMAND

Page: 01

1. Read FIRST-MONTH = THIS-MONTH from THIS MONTH

1.1 For all TYPE

1.1.1 For all SUBCODE

1.1.1.1 For all DISTRICTS

1.1.1.1.1 Read NET-DEMAND in NET DEMAND using TYPE, SUBCODE, DISTRICT and MONTH = ( FIRST-MONTH ) or MONTH = ( FIRST-MONTH + 1 ) or MONTH = ( FIRST-MONTH + 2 ) or MONTH = ( FIRST-MONTH + 3 ) as key

1.1.1.1.2 Set DEMAND-REPORT-UNITS = NET-UNITS
DEMAND-REPORT-TYPE = TYPE
DEMAND-REPORT-SUBCODE = SUBCODE
DEMAND-REPORT-DISTRICT = DISTRICT
DEMAND-REPORT-MONTH = MONTH

1.2 Write DEMAND-REPORT

2. For each PREDICTED-DEMAND entered

2.1 Read NET-DEMAND using TYPE, SUBCODE, DISTRICT and MONTH as key

2.2 Set NET-UNITS = PREDICTED-UNITS

2.3 Write NET-DEMAND in NET DEMAND using TYPE, SUBCODE, DISTRICT and MONTH as key.

3. Go To Process GENERATE STD MPS
1. Read FIRST-MONTH = THIS-MONTH from THIS MONTH

1.1 Read NET-DEMAND in NET DEMAND
   with MONTH = ( FIRST-MONTH + 1 )
   or MONTH = ( FIRST-MONTH + 2 )
   or MONTH = ( FIRST-MONTH + 3 )

1.1.1 For each group of NET-DEMAND with same TYPE and SUBCODE

1.1.1.1 Read PROD-REC in PRODUCTS using TYPE and SUBCODE as key

1.1.1.1.1 If PROD-TYPE = "Standard" then

1.1.1.1.1.1 Set GROSS-REQ( MONTH ) = GROSS-REQ( MONTH ) + NET-DEMAND( MONTH )
   using TYPE and SUBCODE as key

1.1.1.1.2 Read ORDER-ITEM in APPROVED ORDERS
   using TYPE and SUBCODE as key

1.1.1.1.2.1 For each ORDER-ITEM

1.1.1.1.2.1.1 Read ORDER-BROKEN in BROKEN ORDERS with
   ORDER-BROKEN-ITEM-NUMBER = ITEM-NUMBER

1.1.1.1.2.1.2 Set GROSS-REQ( FIRST-MONTH ) =
   GROSS-REQ( FIRST-MONTH ) + ITEM-QUANT - BROKEN-QUANT
   using TYPE and SUBCODE as index
1.1.1.1.1.3 Read ON-HAND in INVENTORY SYSTEM using TYPE, SUBCODE and DISTRICT = "Sign Shop " as key

1.1.1.1.1.4 Read ON-HAND-REC in ON HAND RECORDS using TYPE and SUBCODE as key

1.1.1.1.1.5 Read MPS-REC in MPS RECORDS with MPS-REC-TYPE = TYPE MPS-REC-SUBCODE = SUBCODE

1.1.1.1.1.5.1 For each MPS-REC

1.1.1.1.1.5.1.1 Set SCHEDULE-RECEIPT = SCHEDULE-RECEIPT + MPS-AVAILABLE-TO-PROMISE using MONTH = MPS-PLAN-MONTH as index

1.1.1.1.1.6 If PROD-DESTINY = "Production" or "Both"
Then

1.1.1.1.1.6.1 Read PO-REC in PO RECORDS with PO-REC-TYPE = TYPE PO-REC-SUBCODE = SUBCODE

1.1.1.1.1.6.1.1 For each PO-REC

1.1.1.1.1.6.1.1.1 Set SCHEDULE-RECEIPT = SCHEDULE-RECEIPT + PO-AVAILABLE-TO-PROMISE using MONTH = PO-PLAN-MONTH as index

1.1.1.1.1.7 If PROD-DESTINY = "Purchase" or "Both"
Then
<table>
<thead>
<tr>
<th>Process Name: GENERATE STD MPS</th>
<th>Page: 03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Number: 2-1-2-1.PN</td>
<td></td>
</tr>
</tbody>
</table>

1.1.1.1.1.7.1 Read PR-REC in PR RECORS with
   PR-REC-TYPE = TYPE
   PR-REC-SUBCODE = SUBCODE

1.1.1.1.1.7.1.1 For each PR-REC

1.1.1.1.1.7.1.1.1 Set SCHEDULE-RECEIPT = SCHEDULE-RECEIPT + PR-AVAILABLE-TO-PROMISE
   using MONTH = PR-PLAN-MONTH as index

1.1.1.1.1.8 Set MONTH = FIRST-MONTH
   PLANNED-INVENTORY( FIRST-MONTH ) = ON-HAND-AVAILABLE-TO-PROMISE
   AVAIL-TO-PROM = 0

1.1.1.1.1.8.1 Repeat the following 4 times

1.1.1.1.1.8.1.1 Set PLANNED-INVENTORY( MONTH ) =
   ( PLANNED-INVENTORY( MONTH ) ) +
   ( SCHEDULE-RECEIPT( MONTH ) )
   AVAILABLE( MONTH ) =
   ( PLANNED-INVENTORY( MONTH ) )
   - ( GROSS-REQ( MONTH ) )

1.1.1.1.1.8.1.1.1 If AVAILABLE( MONTH ) < LOW-LEVEL

1.1.1.1.8.1.1.1 Then

1.1.1.1.8.1.1.1.1 If ( AVAILABLE( MONTH ) + AVAIL-TO-PROM ) < LOW-LEVEL
PROCESS NARRATIVE

Process Name: GENERATE STD MPS

Process Number: 2-1-2-1.PN Page: 04

1.1.1.1.8.1.1.1.1.1.1 Then

1.1.1.1.8.1.1.1.1.1.1.1 Set NET-REQ( MONTH ) =
LOT-SIZE

AVAIL-TO-PROM = AVAIL-TO-PROM
+ AVAILABLE( MONTH )
+ NET-REQ( MONTH )

PLANNED-INVENTORY( MONTH + 1 ) = 0

1.1.1.1.8.1.1.1.1.2 Otherwise

1.1.1.1.8.1.1.1.1.2.1 Set

PLANNED-INVENTORY( MONTH + 1 ) = 0

AVAIL-TO-PROM = AVAIL-TO-PROM
+ AVAILABLE( MONTH )

1.1.1.1.8.1.1.1.2 Otherwise Set

PLANNED-INVENTORY( MONTH + 1 ) =
AVAILABLE( MONTH + 1 )

1.1.1.1.8.1.1.2 Set MONTH = MONTH + 1

1.1.1.1.9 Set MONTH = FIRST-MONTH

1.1.1.1.9.1 Repeat the following 4 times

1.1.1.1.9.1.1 If NET-REQ( MONTH ) > 0
1.1.1.1.9.1.1.1 Select MPS-REC-NUMBER in MPS RECORDS which has not been used

1.1.1.1.9.1.1.2 Set MPS-REC-TYPE = TYPE
MPS-REC-SUBCODE = SUBCODE
MPS-PLAN-MONTH = MONTH as key
MPS-REC-UNITS = NET-REQ(MONTH)
MPS-AVAILABLE-TO-PROMISE = MPS-REC-UNITS

1.1.1.1.9.1.1.2.1 If PROD-DESTINY = "Production" or "Both"

1.1.1.1.9.1.1.2.1.1 Then Set MPS-DESTINY = "Production"

1.1.1.1.9.1.1.2.1.2 Otherwise Set MPS-DESTINY = "Purchase"

1.1.1.1.9.1.1.3 Write MPS-REC in MPS RECORDS

2. Go To Process GENERATE SPECIAL MPS.
**PROCESS NARRATIVE**

Process Name: GENERATE SPECIAL MPS

Process Number: 2-1-2-2.PN

<table>
<thead>
<tr>
<th>Page: 01</th>
</tr>
</thead>
</table>

1. Read `FIRST-MONTH = THIS-MONTH` from `THIS MONTH`

1.1 Read `NET-DEMAND` in `NET DEMAND`
   - with `MONTH = ( FIRST-MONTH + 1 )`
   - or `MONTH = ( FIRST-MONTH + 2 )`
   - or `MONTH = ( FIRST-MONTH + 3 )`

1.1.1 For each group of `NET-DEMAND` with same `TYPE` and `SUBCODE`

1.1.1.1 Read `PROD-REC` in `PRODUCTS` using `TYPE` and `SUBCODE` as key

1.1.1.1.1 If `PROD-TYPE = "Special "` or `" Interstate "` then

1.1.1.1.1.1 Set `GROSS-REQ( MONTH ) = GROSS-REQ( MONTH ) + NET-DEMAND( MONTH )`
   - using `TYPE` and `SUBCODE` as key

1.1.1.1.1.2 Read `ORDER-ITEM` in `APPROVED ORDERS`
   - using `TYPE` and `SUBCODE` as key

1.1.1.1.2.1 For each `ORDER-ITEM`

1.1.1.1.2.1.1 Read `ORDER-BROKEN` in `BROKEN ORDERS`
   - with `ORDER-BROKEN-ITEM-NUMBER = ITEM-NUMBER`

1.1.1.1.2.1.2 Set `GROSS-REQ( FIRST-MONTH ) = GROSS-REQ( FIRST-MONTH ) + ITEM-QUANT - BROKEN-ORDER`
   - using `TYPE` and `SUBCODE` as index
1.1.1.1.1.3 Read MPS-REC in MPS RECORS with
   MPS-REC-TYPE = TYPE
   MPS-REC-SUBCODE = SUBCODE

1.1.1.1.1.3.1 For each MPS-REC

1.1.1.1.1.3.1.1 Set SCHEDULE-RECEIPT = SCHEDULE-RECEIPT +
   MPS-AVAILABLE-TO-PROMISE
   using MONTH = MPS-PLAN-MONTH as index

1.1.1.1.1.4 Read PO-REC in PO RECORS with
   PO-REC-TYPE = TYPE
   PO-REC-SUBCODE = SUBCODE

1.1.1.1.1.4.1 For each PO-REC

1.1.1.1.1.4.1.1 Set SCHEDULE-RECEIPT = SCHEDULE-RECEIPT +
   PO-AVAILABLE-TO-PROMISE
   using MONTH = PO-PLAN-MONTH as index

1.1.1.1.1.5 Set MONTH = FIRST-MONTH
   AVAIL-TO-PROM = 0

1.1.1.1.1.5.1 Repeat the following 4 times

1.1.1.1.1.5.1.1 Set AVAILABLE( MONTH ) =
   ( SCHEDULE-RECEIPT( MONTH ) )
   - ( GROSS-REQ ( MONTH ) )

1.1.1.1.1.5.1.1.1 If AVAILABLE( MONTH ) < 0

1.1.1.1.1.5.1.1.1.1 Then
PROCESS NARRATIVE

Process Name: GENERATE SPECIAL MPS

Process Number: 2-1-2-2.PN

1.1.1.1.5.1.1.1.1.1 If ( AVAILABLE( MONTH ) + AVAIL-TO-PROM ) < 0

1.1.1.1.5.1.1.1.1.1 Then

1.1.1.1.5.1.1.1.1.1.1 Set NET-REQ( MONTH ) =
                 ABS( AVAILABLE( MONTH ) + AVAIL-TO-PROM )

1.1.1.1.5.1.1.1.1.1.1.2 Otherwise

1.1.1.1.5.1.1.1.1.1.2.1 Set AVAIL-TO-PROM =
         ( AVAILABLE( MONTH ) + AVAIL-TO-PROM )

1.1.1.1.5.1.2 Set MONTH = MONTH + 1

1.1.1.1.6 Set MONTH = FIRST-MONTH

1.1.1.1.6.1 If NET-REQ( MONTH ) > 0 Then

1.1.1.1.6.1.1 Set N = NET-REQ( MONTH )

1.1.1.1.6.1.1.1 Repeat the following N times

1.1.1.1.6.1.1.1.1 Select MPS-REC-NUMBER in MPS RECORDS
                      which has not being used

1.1.1.1.6.1.1.1.2 Set MPS-REC-TYPE = TYPE
              MPS-REC-SUBCODE = SUBCODE
              MPS-PLAN-MONTH = MONTH
              MPS-REC-UNITS = 1
MPS-AVAILABLE-TO-PROMISE = 1
MPS-DESTINY = "Production"

1.1.1.1.1.6.1.1.1.3 Write MPS-REC in MPS RECORDS

2. Go To ALLOCATE BROKEN ORDERS
PROCESS NARRATIVE

Process Name: GENERATE USER MPS

1. Read THIS-MONTH
2. Read MPS-USER-DATA
   2.1 For each MPS-USER-DATA entered
      2.1.1 Read MPS-REC-NUMBER in MPS RECORDS which has not been used
      2.1.2 Read PROD-REC in PRODUCTS using
             TYPE = MPS-USER-TYPE
             and SUBCODE = MPS-USER-SUBCODE
      2.1.2.1 If PROD-DESTINY = " Purchase "
         2.1.2.1.1 Then
            2.1.2.1.1.1 Set MAXIMUM-LEAD-TIME = PURCHASE-LEAD-TIME
                            MPS-DESTINY = " Purchase "
         2.1.2.1.2 Otherwise
            2.1.2.1.2.1 Set MAXIMUM-LEAD-TIME = 0
      2.1.2.1.2 Read BOM-REC in BILL OF MATERIALS using
                  PARENT-TYPE = MPS-USER-TYPE
                 PARENT-SUBCODE = MPS-USER-SUBCODE
      2.1.2.1.2.1 For each BOM-REC
         2.1.2.1.2.1.1 Read PROD-REC in PRODUCTS using
                           TYPE = SON-TYPE
                           and SUBCODE = SON-SUBCODE
PROCESS NARRATIVE

Process Name: GENERATE USER MPS

2.1.2.1.2.1.1.1.1.1 If PURCHASE-LEAD-TIME > MAXIMUM-LEAD-TIME then

2.1.2.1.2.1.1.1.1.1.1 Set MAXIMUM-LEAD-TIME = PURCHASE-LEAD-TIME

2.1.2.1.2.2 Set MPS-DESTINY = "Production"

2.1.3 Set EARLIEST-PLAN-MONTH = MAXIMUM-LEAD-TIME + THIS-MONTH

2.1.3.1 If MPS-USER-PLAN-MONTH < EARLIEST-PLAN-MONTH then

2.1.3.1.1 Write "MPS cannot be generated"
"Earliest Plan-Month Possible = " EARLIEST-PLAN-MONTH

2.1.3.2 Otherwise

2.1.3.2.1 Set MPS-REC-TYPE = MPS-USER-TYPE
MPS-REC-SUBCODE = MPS-USER-SUBCODE
MPS-REC-UNITS = MPS-USER-UNITS
MPS-PLAN-MONTH = MPS-USER-PLAN-MONTH
MPS-AVAILABLE-TO-PROMISE = MPS-REC-UNITS

2.1.3.2.2 Write MPS-REC in MPS RECORDS.
PROCESS NARRATIVE

Process Name: DELETE MPS

Process Number: 2-1-2-4.PN

1. Read MPS-TO-DELETE
   1.1 For each MPS-TO-DELETE
      1.1.1 Read MPS-REC in MPS RECORDS using
           MPS-REC-NUMBER = MPS-TO-DELETE
      1.1.1.1 If MPS-AVAILABLE-TO-PROMISE = MPS-REC-UNITS
      1.1.1.1.1 Then
         1.1.1.1.1.1 Delete MPS-REC in MPS RECORDS using
                        MPS-REC-NUMBER = MPS-TO-DELETE
      1.1.1.1.2 Otherwise
         1.1.1.1.2.1 Write " MPS cannot be deleted "
PROCESS NARRATIVE

Process Name: GENERATE REQ MPS
Process Number: 2-1-2-5.PN

1. Read THIS-MONTH from THIS-MONTH

2. Read MPS-REQ from Process CHECK FORECAST or from Process ALLOCATE MPS

2.1 Read MPS-REC-NUMBER in MPS RECORDS which has not been used

2.2 Read PROD-REC in PRODUCTS using
   TYPE = MPS-REQ-TYPE
   and SUBCODE = MPS-REQ-TYPE

2.2.1 If PROD-DESTINY = "Purchase"

2.2.1.1 Then

2.2.1.1.1 Set MAXIMUM-LEAD-TIME = PURCHASE-LEAD-TIME
    MPS-DESTINY = "Purchase"

2.2.1.2 Otherwise

2.2.1.2.1 Set MAXIMUM-LEAD-TIME = 0

2.2.1.2.1.1 Read BOM-REC in BILL OF MATERIALS using
   PARENT-TYPE = MPS-USER-TYPE
   PARENT-SUBCODE = MPS-USER-SUBCODE

2.2.1.2.1.1 For each BOM-REC

2.2.1.2.1.1.1 Read PROD-REC in PRODUCTS using
   TYPE = SON-TYPE
   SUBCODE = SON-SUBCODE
2.2.1.2.1.1.1.1.1 If PURCHASE-LEAD-TIME > MAXIMUM-LEAD-TIME then
2.2.1.2.1.1.1.1.1.1 Set MAXIMUM-LEAD-TIME = PURCHASE-LEAD-TIME
2.2.1.2.2 Set MPS-DESTINY = "Production"
2.3 Set MPS-PLAN-MONTH = MAXIMUM-LEAD-TIME + THIS-MONTH
  MPS-REC = MPS-REQ-TYPE
  MPS-REC-SUBCODE = MPS-REQ-SUBCODE
  MPS-REC-UNITS = MPS-REQ-UNITS
  MPS-AVAILABLE-TO-PROMISE = 0
2.3.1 Write MPS-REC in MPS RECORDS.
3. Return MPS-NUMBER to Process CHECK FORECAST or Process ALLOCATE MPS.
<table>
<thead>
<tr>
<th>PROCESS NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Process Name:</strong> IDENTIFY AVAILABLE PLAN CAPACITY</td>
</tr>
<tr>
<td><strong>Process Number:</strong> 2-1-3-1.PN</td>
</tr>
<tr>
<td><strong>Page:</strong> 01</td>
</tr>
</tbody>
</table>

1. Read THIS-MONTH from THIS MONTH

2. Set MONTH = THIS-MONTH

2.1 Repeat the following 4 times

2.1.1 Read DAYS-TO-PLAN using MONTH

2.1.2 Set MONTH = MONTH + 1

3. Read all RESOURCE-REC in RESOURCES

3.1 For each RESOURCE-REC

3.1.1 For each MONTH

3.1.1.1 Set RESOURCE-HOURS-AVAILABLE = 
DAYS-TO-PLAN * RESOURCE-QUANT * RESOURCE-WORK-HOURS 
using MONTH as index

3.1.1.2 Write RESOURCE-PLAN-CAPACITY

3.1.1.3 Ask User Option

3.1.1.3.1 If User Option is " Change Resource Plan Capacity "

3.1.1.3.1.1 Then Read RES-CHANGE-WORK-HOURS 
RES-CHANGE-QUANT 
RES-CHANGE-EFFICIENCY

3.1.1.3.1.1.1 If RES-CHANGE-WORK-HOURS is entered
PROCESS NARRATIVE

Process Name: IDENTIFY AVAILABLE PLAN CAPACITY

Process Number: 2-1-3-1.PN

3.1.1.1.3.1.1.1.1 Then Set RESOURCE-WORK-HOURS = RES-CHANGE-WORK-HOURS

3.1.1.1.3.1.1.2 If RES-CHANGE-QUANT is entered

3.1.1.1.3.1.1.2.1 Then Set RESOURCE-QUANT = RES-CHANGE-QUANT

3.1.1.1.3.1.1.3 If RES-CHANGE-EFFICIENCY

3.1.1.1.3.1.1.3.1 Then Set RES-EFFICIENCY = RES-CHANGE-EFFICIENCY

3.1.1.4 Set RES-HOURS-AVAILABLE = DAYS-TO-PLAN * RESOURCE-QUANT * RESOURCE-WORK-HOURS

3.1.1.5 Write RESOURCE-PLAN-REC in RESOURCE PLAN RECS using RESOURCE-NUMBER and MONTH as key.

4. Go To Process EXECUTE PLAN WORK LOAD.
PROCESS NARRATIVE

Process Name: EXECUTE PLAN WORK LOAD

Process Number: 2-1-3-2-1.PN

1. Read THIS-MONTH from THIS MONTH

2. Set FIRST-MONTH = THIS-MONTH

2.1 Read PO-REC in PRODUCTS RECORDS with
   PO-PLAN-MONTH < = FIRST-MONTH
   and PO-REC-STATUS not equal "Scheduled"

2.1.1 For each PO-REC

2.1.1.1 Read PROD-ROUTE in ROUTES using
       TYPE = PO-REC-TYPE
       and SUBCODE = PO-REC-SUBCODE

2.1.1.1.1 For each PROD-ROUTE

2.1.1.1.1.1 Read OPERATION-REC in OPERATIONS
                using OPERATION-CODE as key

2.1.1.1.1.1 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS
                using OPERATION-MACHINE = RESOURCE-NUMBER
                and PLAN-MONTH = FIRST-MONTH

2.1.1.1.1.1.1 Set WORK-LOAD = WORK-LOAD
                     + SET-UP-TIME
                     + ( OP-TIME-PER-UNIT * PO-REC-UNITS )

2.1.1.1.1.1.1 Write RESOURCE-PLAN-REC in
                 RESOURCE PLAN RECS using
                 RESOURCE-NUMBER and PLAN-MONTH as key

2.1.1.1.1.2 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS
                 using OPERATION-WORKER = RESOURCE-NUMBER
                 and PLAN-MONTH = FIRST-MONTH
## PROCESS NARRATIVE

**Process Name:** EXECUTE PLAN WORK LOAD

**Process Number:** 2-1-3-2-1.PN

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1.1.1.2.1</td>
<td>Set ( \text{WORK-LOAD} = \text{WORK-LOAD} + \text{SET-UP-TIME} + (\text{OP-TIME-PER-UNIT} \times \text{PO-REC-UNITS}) )</td>
</tr>
<tr>
<td>2.1.1.1.1.2.1.1</td>
<td>Write ( \text{RESOURCE-PLAN-REC} ) in ( \text{RESOURCE PLAN RECS} ) using ( \text{RESOURCE-NUMBER} ) and ( \text{PLAN-MONTH} ) as key</td>
</tr>
<tr>
<td>2.2</td>
<td>Read ( \text{MPS-REC} ) in ( \text{MPS RECORDS} ) with ( \text{MPS-PLAN-MONTH} \leq \text{FIRST-MONTH} )</td>
</tr>
<tr>
<td>2.2.1</td>
<td>For each ( \text{MPS-REC} )</td>
</tr>
<tr>
<td>2.2.1.1</td>
<td>Read ( \text{PROD-ROUTE} ) in ( \text{ROUTES} ) using ( \text{TYPE} = \text{MPS-REC-TYPE} ) and ( \text{SUBCODE} = \text{MPS-REC-SUBCODE} )</td>
</tr>
<tr>
<td>2.2.1.1.1</td>
<td>For each ( \text{PROD-ROUTE} )</td>
</tr>
<tr>
<td>2.2.1.1.1.1</td>
<td>Read ( \text{OPERATION-REC} ) in ( \text{OPERATIONS} ) using ( \text{OPERATION-CODE} ) as key</td>
</tr>
<tr>
<td>2.2.1.1.1.1.1</td>
<td>Read ( \text{RESOURCE-PLAN-REC} ) in ( \text{RESOURCE PLAN RECS} ) using ( \text{OPERATION-MACHINE} = \text{RESOURCE-NUMBER} ) and ( \text{PLAN-MONTH} = \text{FIRST-MONTH} )</td>
</tr>
<tr>
<td>2.2.1.1.1.1.1.1</td>
<td>Set ( \text{WORK-LOAD} = \text{WORK-LOAD} + \text{SET-UP-TIME} + (\text{OP-TIME-PER-UNIT} \times \text{PO-REC-UNITS}) )</td>
</tr>
<tr>
<td>2.2.1.1.1.1.1.1.1</td>
<td>Write ( \text{RESOURCE-PLAN-REC} ) in ( \text{RESOURCE PLAN RECS} ) using ( \text{RESOURCE-NUMBER} ) and ( \text{PLAN-MONTH} ) as key</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: EXECUTE PLAN WORK LOAD

2.2.1.1.1.1.2 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS using OPERATION-WORKER = RESOURCE-NUMBER and PLAN-MONTH = FIRST-MONTH

2.2.1.1.1.1.2.1 Set WORK-LOAD = WORK-LOAD + SET-UP-TIME + ( OP-TIME-PER-UNIT * PO-REC-UNITS )

2.2.1.1.1.1.2.1.1 Write RESOURCE-PLAN-REC in RESOURCE PLAN RECS using RESOURCE-NUMBER and PLAN-MONTH as key

3. Set MONTH = FIRST-MONTH + 1

4. Repeat the following 3 times

4.1 Read PO-REC in PRODUCTS RECORDS with PO-PLAN-MONTH = MONTH and PO-REC-STATUS not equal "Scheduled"

4.1.1 For each PO-REC

4.1.1.1 Read PROD-ROUTE in ROUTES using TYPE = PO-REC-TYPE and SUBCODE = PO-REC-SUBCODE

4.1.1.1.1 For each PROD-ROUTE

4.1.1.1.1.1 Read OPERATION-REC in OPERATIONS using OPERATION-CODE as key

4.1.1.1.1.1.1 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS using OPERATION-MACHINE = RESOURCE-NUMBER and PLAN-MONTH = MONTH
PROCESS NARRATIVE

Process Name: EXECUTE PLAN WORK LOAD

Process Number: 2-1-3-2-1.PN

Page: 05

4.2.1.1.1.1.1 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS
    using OPERATION-MACHINE = RESOURCE-NUMBER
    and PLAN-MONTH = MONTH

4.2.1.1.1.1.1.1 Set WORK-LOAD = WORK-LOAD
    + SET-UP-TIME
    + ( OP-TIME-PER-UNIT * PO-REC-UNITS )

4.2.1.1.1.1.1.1.1 Write RESOURCE-PLAN-REC in
    RESOURCE PLAN RECS using
    RESOURCE-NUMBER and PLAN-MONTH as key

4.2.1.1.1.2 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS
    using OPERATION-WORKER = RESOURCE-NUMBER
    and PLAN-MONTH = MONTH

4.2.1.1.1.2.1 Set WORK-LOAD = WORK-LOAD
    + SET-UP-TIME
    + ( OP-TIME-PER-UNIT * PO-REC-UNITS )

4.2.1.1.1.2.1.1 Write RESOURCE-PLAN-REC in
    RESOURCE PLAN RECS using
    RESOURCE-NUMBER and PLAN-MONTH as key

4.3 Set MONTH = MONTH + 1

5. Go To Process REPORT PLAN WORK LOAD.
PROCESS NARRATIVE

Process Name: REPORT PLAN WORK LOAD

1. Read THIS-MONTH from THIS MONTH

2. Set MONTH = THIS-MONTH

2.1 Repeat the following 4 times

2.1.1 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS using PLAN-MONTH = MONTH

2.1.1.1 For each RESOURCE-PLAN-REC

2.1.1.1.1 Set RATIO = (WORK-LOAD) / (RES-HOURS-AVAILABLE)

2.1.1.1.1.1 If RATIO > RES-EFFICIENCY then

2.1.1.1.1.1.1 Set RES-STATUS = "Bottleneck"

2.1.1.1.1.2 Otherwise

2.1.1.1.2.1 Set RES-STATUS = "Non-Bottleneck"

2.1.1.2 Set REPORT-PLAN-MONTH = MONTH

    REPORT-RES-NUMBER = RESOURCE-NUMBER
    REPORT-RES-HOURS-AVAILABLE = RESOURCE-HOURS-AVAILABLE
    REPORT-RES-EFFICIENCY = RES-EFFICIENCY
    REPORT-LOAD-RATIO = RATIO
    REPORT-RES-STATUS = RES-STATUS
2.1.1.1.3 Read RESOURCE-REC in RESOURCES using RESOURCE-NUMBER as key

2.1.1.1.3.1 Set REPORT-RES-QUANT = RESOURCE-QUANT

    REPORT-RES-DESCRIPTION = RESOURCE-DESCRIPTION

2.1.1.1.4 Write PLAN-WORK-LOAD-REPORT

2.1.2 Set MONTH = MONTH + 1

3. Go To Process APPROVE CAPACITY PLAN
## PROCESS NARRATIVE

**Process Name:** CHANGE MPS AND PO  
**Process Number:** 2-1-3-2-3.PN  
**Page:** 01

1. Read THIS-MONTH from THIS MONTH  
2. Set FIRST-MONTH = THIS-MONTH  

2.1 Read PO-REC in PO RECORDS with  
   PO-PLAN-MONTH < = FIRST-MONTH  
   and PO-REC-STATUS not equal "Scheduled"  

2.1.1 For each PO-REC  

   2.1.1.1 Set  
      PO-PLANNED-NUMBER = PO-REC-NUMBER  
      PO-PLANNED-MONTH = PO-PLAN-MONTH  
      PO-PLANNED-TYPE = PO-REC-TYPE  
      PO-PLANNED-SUBCODE = PO-REC-SUBCODE  
      PO-PLANNED-UNITS = PO-REC-UNITS  

   2.1.1.1.1 If PO-ORIG-MONTH = "Null"  

   2.1.1.1.1 Then Set  
      PO-ORIGINAL-MONTH = PO-PLAN-MONTH  

   2.1.1.1.2 Otherwise  
      Set  
      PO-ORIGINAL-MONTH = PO-ORIG-MONTH  

   2.1.1.2 Set PO-PLANNED-BOOK =  
      1 - (PO-AVAILABLE-TO-PROMISE)/(PO-REC-UNITS)  

   2.1.1.3 Set MAX-LEAD-TIME = 0  

   2.1.1.3.1 Read BOM-REC in BILL OF MATERIALS using  
      PARENT-TYPE = PO-REC-TYPE  
      and PARENT-SUBCODE = PO-REC-SUBCODE  

   2.1.1.3.1.1 For each BOM-REC
<table>
<thead>
<tr>
<th>PROCESS NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name: CHANGE MPS AND PO</td>
</tr>
<tr>
<td>Process Number: 2-1-3-2-3.PN</td>
</tr>
</tbody>
</table>

2.1.1.3.1.1.1 Read PROD-REC in PRODUCTS using
TYPE = SON-TYPE
and SUBCODE = SON-SUBCODE

2.1.1.3.1.1.1 If PURCHASE-LEAD-TIME > MAX-LEAD-TIME
2.1.1.3.1.1.1.1 Then Set MAX-LEAD-TIME = PURCHASE-LEAD-TIME

2.1.1.4 Set PO-MAX-EXPEDITION = THIS-MONTH + MAX-LEAD-TIME

2.1.1.5 Write PO-PLANNED-REPORT

2.2 Read MPS-REC in MPS RECORDS with
MPS-PLAN-MONTH <= FIRST-MONTH

2.2.1 For each MPS-REC
2.2.1.1 Set MPS-PLANNED-NUMBER = MPS-REC-NUMBER
MPS-PLANNED-MONTH = MPS-PLAN-MONTH
MPS-PLANNED-TYPE = MPS-REC-TYPE
MPS-PLANNED-SUBCODE = MPS-REC-SUBCODE
MPS-PLANNED-UNITS = MPS-REC-UNITS
MPS-PLANNED-DESTINY = MPS-DESTINY

2.2.1.1.1 If MPS-ORIG-MONTH = " Null ">
2.2.1.1.1.1 Then Set MPS-ORIGINAL-MONTH = MPS-PLAN-MONTH
2.2.1.1.1.2 Otherwise
Set MPS-ORIGINAL-MONTH = MPS-ORIG-MONTH

2.2.1.2 Set MPS-PLANNED-BOOK =
1 - ( MPS-AVAILABLE-TO-PROMISE )/( MPS-REC-UNITS )
### PROCESS NARRATIVE

**Process Name:** CHANGE MPS AND PO  
**Process Number:** 2-1-3-2-3.PN  
**Page:** 03

#### 2.2.1.3 Set MAX-LEAD-TIME = 0

#### 2.2.1.3.1 Read BOM-REC in BILL OF MATERIALS using  
- **PARENT-TYPE** = MPS-REC-TYPE  
- **PARENT-SUBCODE** = MPS-REC-SUBCODE

#### 2.2.1.3.1.1 For each BOM-REC

#### 2.2.1.3.1.1.1 Read PROD-REC in PRODUCTS using  
- **TYPE** = SON-TYPE  
- **SUBCODE** = SON-SUBCODE

#### 2.2.1.3.1.1.1.1 If PURCHASE-LEAD-TIME > MAX-LEAD-TIME

#### 2.2.1.3.1.1.1.1.1 Then Set MAX-LEAD-TIME = PURCHASE-LEAD-TIME

#### 2.2.1.4 Set MPS-MAX-EXPEDITION = THIS-MONTH + MAX-LEAD-TIME

#### 2.2.1.5 Write MPS-PLANNED-REPORT

#### 3. Set MONTH = THIS-MONTH + 1

#### 4. Repeat the following 3 times

##### 4.1 Read PO-REC in PO RECORDS with  
- **PO-PLAN-MONTH** = MONTH  
- **PO-REC-STATUS** not equal "Scheduled"

##### 4.1.1 For each PO-REC

##### 4.1.1.1 Set PO-PLANNED-NUMBER = PO-REC-NUMBER  
- **PO-PLANNED-MONTH** = PO-PLAN-MONTH
PROCESS NARRATIVE

Process Name: CHANGE MPS AND PO

Process Number: 2-1-3-2-3.PN

Page: 04

PO-PLANNED-TYPE = PO-REC-TYPE
PO-PLANNED-SUBCODE = PO-REC-SUBCODE
PO-PLANNED-UNITS = PO-REC-UNITS

4.1.1.1.1 If PO-ORIG-MONTH = " Null "

4.1.1.1.1.1 Then Set PO-ORIGINAL-MONTH = PO-PLAN-MONTH

4.1.1.1.1.2 Otherwise

Set PO-ORIGINAL-MONTH = PO-ORIG-MONTH

4.1.1.2 Set PO-PLANNED-BOOK =

1 - ( PO-AVAILABLE-TO-PROMISE )/( PO-REC-UNITS )

4.1.1.3 Set MAX-LEAD-TIME = 0

4.1.1.3.1 Read BOM-REC in BILL OF MATERIALS using

   PARENT-TYPE = PO-REC-TYPE
   and PARENT-SUBCODE = PO-REC-SUBCODE

4.1.1.3.1.1 For each BOM-REC

   4.1.1.3.1.1.1 Read PROD-REC in PRODUCTS using

       TYPE = SON-TYPE
       and SUBCODE = SON-SUBCODE

4.1.1.3.1.1.1 If PURCHASE-LEAD-TIME > MAX-LEAD-TIME

4.1.1.3.1.1.1.1 Then Set MAX-LEAD-TIME = PURCHASE-LEAD-TIME

4.1.1.4 Set PO-MAX-EXPEDITION = THIS-MONTH + MAX-LEAD-TIME

4.1.1.5 Write PO-PLANNED-REPORT
Process Name: CHANGE MPS AND PO

Process Number: 2-1-3-2-3.PN

4.2 Read MPS-REC in MPS RECORDS with
   MPS-PLAN-MONTH = MONTH

4.2.1 For each MPS-REC

4.2.1.1 Set
   MPS-PLANNED-NUMBER = MPS-REC-NUMBER
   MPS-PLANNED-MONTH = MPS-PLAN-MONTH
   MPS-PLANNED-TYPE = MPS-REC-TYPE
   MPS-PLANNED-SUBCODE = MPS-REC-SUBCODE
   MPS-PLANNED-UNITS = MPS-REC-UNITS
   MPS-PLANNED-DESTINY = MPS-DESTINY

4.2.1.1.1 If MPS-ORIG-MONTH = " Null "

4.2.1.1.1.1 Then Set MPS-ORIGINAL-MONTH = MPS-PLAN-MONTH

4.2.1.1.2 Otherwise
   Set MPS-ORIGINAL-MONTH = MPS-ORIG-MONTH

4.2.1.2 Set MPS-PLANNED-BOOK =
   1 - ( MPS-AVAILABLE-TO-PROMISE )/( MPS-REC-UNITS )

4.2.1.3 Set MAX-LEAD-TIME = 0

4.2.1.3.1 Read BOM-REC in BILL OF MATERIALS using
   PARENT-TYPE = MPS-REC-TYPE
   and PARENT-SUBCODE = MPS-REC-SUBCODE

4.2.1.3.1.1 For each BOM-REC

4.2.1.3.1.1.1 Read PROD-REC in PRODUCTS using
   TYPE = SON-TYPE
   and SUBCODE = SON-SUBCODE
### PROCESS NARRATIVE

**Process Name:** CHANGE MPS AND PO  
**Process Number:** 2-1-3-2-3.PN

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4.2.1.3.1.1.1.1 | If PURCHASE-LEAD-TIME > MAX-LEAD-TIME  
| Then Set MAX-LEAD-TIME = PURCHASE-LEAD-TIME |
| 4.2.1.4 | Set MPS-MAX-EXPEDITION = THIS-MONTH + MAX-LEAD-TIME |
| 4.2.1.5 | Write MPS-PLANNED-REPORT |
| 4.3 | Set MONTH = MONTH + 1 |
| 5. | Ask User Option |
| 5.1 | If User Option is "Change PO"
| 5.1.1 | Then Read PO-DATA |
| 5.1.1.1 | For each PO-DATA entered |
| 5.1.1.1.1 | Read PO-REC in PO RECORDS  
*with PO-REC-NUMBER = PO-NUMBER* |
| 5.1.1.1.1.1 | Set PO-ORIG-MONTH = PO-PLAN-MONTH  
PO-PLAN-MONTH = PO-NEW-PLAN-MONTH |
| 5.1.1.2 | Write PO-REC in PO RECORDS  
*using PO-REC-NUMBER as key* |
| 5.2 | If User Option is "Change MPS"
| 5.2.1 | Then Read MPS-DATA |
| 5.2.1.1 | For each MPS-DATA entered |
5.2.1.1.1 Then Read MPS-REC in MPS RECORDS with MPS-REC-NUMBER = MPS-NUMBER

5.2.1.1.1.1 If NEW-MPS-PLAN-MONTH is entered

5.2.1.1.1.1.1 Then Set MPS-ORIG-MONTH = MPS-PLAN-MONTH
MPS-PLAN-MONTH = MPS-NEW-PLAN-MONTH

5.2.1.1.1.2 If NEW-MPS-DESTINY is entered

5.2.1.1.1.2.1 Then Set MPS-DESTINY = MPS-NEW-DESTINY

5.2.1.1.2 Write MPS-REC in MPS RECORDS using MPS-REC-NUMBER as key

6. Go To Process EXECUTE WORK LOAD.
PROCESS NARRATIVE

Process Name: DELETE MPS FROM PLAN

Process Number: 2-1-3-2-4.PN

1. Read THIS-MONTH in THIS MONTH

2. Set MONTH = THIS-MONTH

3. Select all MPS-REC in MPS RECORDS
   with MPS-PLAN-MONTH = MONTH
   and MPS-DESTINY = "Production"

3.1 For each MPS-REC

3.1.1 Read PROD-REC in PRODUCTS using
    TYPE = MPS-REC-TYPE
    and SUBCODE = MPS-REC-SUBCODE

3.1.1.1 If PROD-DESTINY = "Production"

3.1.1.1.1 Then Delete MPS-REC from selected group

3.1.1.1.2 Otherwise
    Set MPS-PRIORITY-TO-PURCHASE = PRIORITY-TO-PURCHASE
    using MPS-REC-NUMBER as index

3.2 Sort MPS-REC by MPS-PRIORITY-TO-PURCHASE

3.2.1 Select all RESOURCE-PLAN-REC in RESOURCE PLAN RECS
      with PLAN-MONTH = MONTH

3.2.1.1 Select one RESOURCE-PLAN-REC from selected group

3.2.1.1.1 If can’t find more RESOURCE-PLAN-REC

3.2.1.1.1.1 Then Set MONTH = MONTH + 1

3.2.1.1.1.1 If MONTH < THIS-MONTH + 4
### PROCESS NARRATIVE

**Process Name:** DELETE MPS FROM PLAN  
**Process Number:** 2-1-3-2-4.PN  
**Page:** 02

1. **Then Go Back to 3**  
   - (Do another Month)

2. **Otherwise Call Process**  
   - REPORT PLAN WORK LOAD

3. **Otherwise**  
   - (There still Resources in selected group)

4. **Set RATIO =**  
   - (WORK-LOAD)/(RES-EFFICIENCY * RES-HOURS-AVAILABLE)

5. **If RATIO > 1**

   1. **Then Select MPS-REC with highest priority to purchase**

   2. **If can't find more MPS-REC in sorted group**

   3. **Then Set MONTH = MONTH + 1**

   4. **If MONTH < THIS-MONTH + 4**

   5. **Then Go Back to 3**  
      - (Do another Month)

6. **Otherwise Call Process**  
   - REPORT PLAN WORK LOAD

7. **Otherwise**  
   - (There are still MPS that can be deleted from Plan)
### PROCESS NARRATIVE

**Process Name:** DELETE MPS FROM PLAN  
**Process Number:** 2-1-3-2-4.PN  
**Page:** 03

<table>
<thead>
<tr>
<th>3.2.1.1.1.2.1.1.1.2.1.1.2.1.1</th>
<th>Read PROD-ROUTE in ROUTES using TYPE = MPS-REC-TYPE and SUBCODE = MPS-REC-SUBCODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.2.1.1.1</td>
<td>Read OPERATION-REC in OPERATIONS using OPERATION-CODE as key</td>
</tr>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.1</td>
<td>If OPERATION-MACHINE = RESOURCE-NUMBER or OPERATION-WORKER = RESOURCE-NUMBER</td>
</tr>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.1.1</td>
<td>Then</td>
</tr>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.1.1.1.1.1</td>
<td>Set MPS-DESTINY = &quot;Purchase&quot;</td>
</tr>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1</td>
<td>Read PROD-ROUTE in ROUTES using TYPE = MPS-REC-TYPE and SUBCODE = MPS-REC-SUBCODE</td>
</tr>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.2.1.1</td>
<td>Read OPERATION-REC in OPERATIONS using OPERATION-CODE as key</td>
</tr>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.1</td>
<td>For each OPERATION-REC</td>
</tr>
<tr>
<td>3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1</td>
<td>Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS using PLAN-MONTH = MONTH and RESOURCE-NUMBER = OPERATION-MACHINE</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: DELETE MPS FROM PLAN

Process Number: 2-1-3-2-4.PN

3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.1.1 Set WORK-LOAD = WORK-LOAD - ( MPS-REC-UNITS * OP-TIME-PER-UNIT ) - SET-UP-TIME

3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.1.2 Write RESOURCE-PLAN-REC in RESOURCE PLAN RECS using RESOURCE-NUMBER as key

3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.1.2.1 Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS using PLAN-MONTH = MONTH and RESOURCE-NUMBER = OPERATION-MACHINE

3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.2 Set WORK-LOAD = WORK-LOAD - ( MPS-REC-UNITS * OP-TIME-PER-UNIT ) - SET-UP-TIME

3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.1.1.2.1.2.2 Write RESOURCE-PLAN-REC in RESOURCE PLAN RECS using RESOURCE-NUMBER as key

3.2.1.1.1.2.1.1.1.2.1.1.1.2.1.2 Delete MPS-REC from sorted group

3.2.1.1.1.2.1.1.1.2.2 Go Back to 3.2.1
( Select Resource Plan Records )

3.2.1.1.1.2.1.2 Otherwise ( Resource is not a Bottleneck )

3.2.1.1.1.2.1.2.1 Delete RESOURCE-PLAN-REC from selected group

3.2.1.1.1.2.1.2.2 Go Back to 3.2.1.1
( Select another Resource Plan Record from Selected Group)

4. Go To Process REPORT PLAN WORK LOAD.
PROCESS NARRATIVE

Process Name: APPROVE CAPACITY PLAN

1. Read User Option

1.1 If User Option is " Change Resources Hours Available "

1.1.1 Then Go To Process IDENTIFY AVAILABLE PLAN CAPACITY

1.2 If User Option is " Make Changes in Production Orders and MPS "

1.2.1 Then Go To Process CHANGE MPS AND PO

1.3 If User Option is " Let The System Find a Feasible Plan "

1.3.1 Then Go To Process DELETE MPS FROM PLAN

1.4 If User Option is " Approve Capacity Plan "

1.4.1 Then

1.4.1.1 Read THIS-MONTH from THIS-MONTH

1.4.1.2 Set MONTH = THIS-MONTH

1.4.1.2.1 Repeat the following 4 times

1.4.1.2.1.1 Read MPS-REC in MPS RECORDS

using MPS-PLAN-MONTH = MONTH

1.4.1.2.1.1.1 For each MPS-REC

1.4.1.2.1.1.1.1 If MPS-DESTINY = " Production "
**PROCESS NARRATIVE**

Process Name: APPROVE CAPACITY PLAN

Process Number: 2-1-3-3.PN

<table>
<thead>
<tr>
<th>1.4.1.2.1.1.1.1.1</th>
<th>Then Select PO-REC-NUMBER in PO RECORDS which has not been used</th>
</tr>
</thead>
</table>
| 1.4.1.2.1.1.1.1.2 | Set PO-REC-TYPE = MPS-REC-TYPE  
                   | PO-REC-SUBCODE = MPS-REC-SUBCODE  
                   | PO-REC-UNITS = MPS-REC-UNITS  
                   | PO-PLAN-MONTH = MPS-PLAN-MONTH  
                   | PO-ORIG-MONTH = MPS-ORIG-MONTH  
                   | PO-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE  
                   | UNITS-REQ-ADJUSTMENT in PO = UNITS-REQ-ADJUSTMENT in MPS |
| 1.4.1.2.1.1.1.1.3 | Write PO-REC in PO RECORDS using PO-REC-NUMBER as key |
| 1.4.1.2.1.1.1.1.4 | Read MPS-ORDER-REC in MPS ORDER using  
                   | MPS-ORDER-REC-MPS = MPS-REC-NUMBER |
| 1.4.1.2.1.1.1.1.4.1 | For each MPS-ORDER-REC |
| 1.4.1.2.1.1.1.1.1.4.1.2 | Set  
                   | PO-ORDER-REC-ORDER = MPS-ORDER-REC-ORDER  
                   | PO-ORDER-REC-ITEM = MPS-ORDER-REC-ITEM  
                   | PO-ORDER-REC-PO = MPS-ORDER-REC-PO  
                   | PO-ORDER-REC-UNITS = MPS-ORDER-REC-UNITS |
| 1.4.1.2.1.1.1.1.1.4.1.3 | Write PO-ORDER-REC in PO-ORDER |
| 1.4.1.2.1.1.1.1.1.4.1.4 | Delete MPS-ORDER-REC in MPS ORDER |
| 1.4.1.2.1.1.1.1.2 | Otherwise Select PR-REC-NUMBER in PR RECORDS which has not been used |
PROCESS NARRATIVE

Process Name: APPROVE CAPACITY PLAN
Process Number: 2-1-3-3.PN

1.4.1.2.1.1.2.1 Set PR-REC-TYPE = MPS-REC-TYPE
PR-REC-SUBCODE = MPS-REC-SUBCODE
PR-REC-UNITS = MPS-REC-UNITS
PR-PLAN-MONTH = MPS-PLAN-MONTH
PR-ORIG-MONTH = MPS-ORIG-MONTH
PR-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE
UNITS-REQ-ADJUSTMENT in PR = UNITS-REQ-ADJUSTMENT in MPS

1.4.1.2.1.1.2.3 Write PR-REC in PR RECORDS using
PR-REC-NUMBER as key

1.4.1.2.1.1.2.4 Read MPS-ORDER-REC in MPS ORDER using
MPS-ORDER-REC-MPS = MPS-REC-NUMBER

1.4.1.2.1.1.2.4.1 For each MPS-ORDER-REC

1.4.1.2.1.1.2.4.1.2 Set
PR-ORDER-REC-ORDER = MPS-ORDER-REC-ORDER
PR-ORDER-REC-ITEM = MPS-ORDER-REC-ITEM
PR-ORDER-REC-PR = MPS-ORDER-REC-PR
PR-ORDER-REC-UNITS = MPS-ORDER-REC-UNITS

1.4.1.2.1.1.2.4.1.3 Write PR-ORDER-REC in PR-ORDER

1.4.1.2.1.1.2.4.1.4 Delete MPS-ORDER-REC in MPS ORDER

1.4.1.2.1.1.2 Delete MPS-REC in MPS RECORDS
using MPS-REC-NUMBER as key

1.4.1.2.2 Set MONTH = MONTH + 1
1. Read FIRST-MONTH = THIS-MONTH in THIS MONTH

1.1 Read PO-REC in PO RECORDS
   with PO-PLAN-MONTH = ( MONTH )
   or PO-PLAN-MONTH = ( MONTH + 1 )
   or PO-PLAN-MONTH = ( MONTH + 2 )
   or PO-PLAN-MONTH = ( MONTH + 3 )

1.1.1 For each PO-REC

1.1.1.1 Read BOM-REC using
   PARENT-TYPE = PO-REC-TYPE
   and PARENT-SUBCODE = PO-REC-SUBCODE

1.1.1.1.1 For each BOM-REC

1.1.1.1.1.1 Set GROSS-REQ = GROSS-REQ
   + ( PO-REC-UNITS * SON-QUANT )
   using SON-TYPE, SON-SUBCODE and PO-PLAN-MONTH
   as index

2. For each group of GROSS-REQ using the same TYPE and
   SUBCODE

2.1 Read ON-HAND in INVENTORY SYSTEM
   using TYPE, SUBCODE and DISTRICT = " Sign Shop "
   as key

2.1.1 For each ON-HAND

2.1.1.1 Set PLANNED-INVENTORY( FIRST-MONTH ) =
   ON-HAND-QUANT
PROCESS NARRATIVE

Process Name: GENERATE RM-PR

Process Number: 2-2-1-1.PN

2.2 Read RM-PR in RAW MATERIAL PR
   with RM-PR-TYPE = TYPE
   and RM-PR-SUBCODE = SUBCODE

2.2.1 For each RM-PR

2.2.1.1 If RM-PR-PURCHASE-ORDER = " Null "

2.2.1.1.1 Then Delete RM-PR in RAW MATERIAL PR
   using RM-PR-NUMBER as key

2.2.1.1.2 Otherwise ( RM-PR purchased already )

2.2.1.1.2.1 Set SCHEDULE-RECEIPT = SCHEDULE-RECEIPT
   + RM-PR-UNITS
   using MONTH = EXPECTED-RECEIPT as index

2.3 Read PROD-REC in PRODUCTS
   using TYPE and SUBCODE as key

2.4 Set MONTH = FIRST-MONTH
   AVAIL-TO-PROM = 0

2.4.1 Repeat the following 4 times

2.4.1.1 Set PLANNED-INVENTORY( MONTH ) =
   PLANNED-INVENTORY( MONTH )
   + SCHEDULE-RECEIPT( MONTH )

   AVAILABLE( MONTH ) =
   PLANNED-INVENTORY( MONTH )
   - GROSS-REQ( MONTH )

2.4.1.1.1 If AVAILABLE( MONTH ) < LOW-LEVEL
PROCESS NARRATIVE

Process Name: GENERATE RM-PR

Process Number: 2-2-1-1.PN

Page: 03

2.4.1.1.1.1 Then
2.4.1.1.1.1.1 If ( AVAILABLE(MONTH) + AVAIL-TO-PROM ) < LOW-LEVEL
2.4.1.1.1.1.1.1 Then
2.4.1.1.1.1.1.1.1 Set NET-REQ(MONTH) = LOT-SIZE
AVAIL-TO-PROM = AVAIL-TO-PROM + AVAILABLE(MONTH) + NET-REQ(MONTH)
PLANNED-INVENTORY(MONTH + 1) = 0
2.4.1.1.1.1.1.2 Otherwise
2.4.1.1.1.1.1.2.1 Set PLANNED-INVENTORY(MONTH + 1) = 0
AVAIL-TO-PROM = AVAIL-TO-PROM + AVAILABLE(MONTH)
2.4.1.1.1.2 Otherwise
2.4.1.1.1.2.1 Set PLANNED-INVENTORY(MONTH + 1) =
AVAILABLE(MONTH)
2.4.2 Set MONTH = MONTH + 1
2.5 Set MONTH = FIRST-MONTH
2.5.1 Repeat the following 4 times
2.5.1.1 If NET-REQ(MONTH) > 0
2.5.1.1.1 Then Select RM-PR-NUMBER which has not been used in RAW MATERIAL PR

2.5.1.1.2 Set RM-PR-TYPE = TYPE
    RM-PR-SUBCODE = SUBCODE
    RM-PR-UNITS = NET-REQ(MONTH)
    RM-PR-PLAN-MONTH = MONTH

2.5.1.1.3 Write RM-PR in RAW MATERIAL PR using RM-PR-NUMBER as key.
1. Read NEW-RM-PR-DATA

1.1 For each NEW-RM-PR-DATA entered

1.1.1 Select RM-PR-NUMBER in RAW MATERIAL PR which has not been used

1.1.1.1 Set RM-PR-TYPE = NEW-RM-PR-TYPE
    RM-PR-SUBCODE = NEW-RM-PR-SUBCODE
    RM-PR-UNITS = NEW-RM-PR-UNITS
    RM-PR-PLAN-MONTH = NEW-RM-PR-MONTH

1.1.1.2 Write RM-PR in RAW MATERIAL PR using RM-PR-NUMBER as key.
PROCESS NARRATIVE

Process Name: GENERATE USER PR

Process Number: 2-2-1-3.PN

1. Read THIS-MONTH from THIS MONTH

2. Read NEW-PR-DATA

2.1 For each NEW-PR-DATA entered

2.1.1 Select PR-REC-NUMBER in PR RECORDS which has not been used

2.1.1.1 Set PR-REC-TYPE = NEW-PR-TYPE
    PR-REC-SUBCODE = NEW-PR-SUBCODE
    PR-REC-UNITS = NEW-PR-UNITS

2.1.1.2 Read PROD-REC using
    TYPE = NEW-PR-TYPE
    SUBCODE = NEW-PR-SUBCODE

2.1.1.2.1 Set EARLIEST-MONTH = ( THIS-MONTH + PURCHASE-LEAD-TIME )

2.1.1.2.1.1 If NEW-PR-PLAN-MONTH < EARLIEST-MONTH

2.1.1.2.1.1.1 Then Write " PR Plan Month is infeasible "
                       " Earliest Month = " EARLIEST-MONTH

2.1.1.2.1.2 Ask User Option

2.1.1.2.1.2.1 If Option is " Confirm Plan Month "

2.1.1.2.1.2.1.1 Then Set
    PR-REC-PLAN-MONTH = NEW-PR-MONTH

2.1.1.2.1.2.2 If Option is " Change Plan Month "
<table>
<thead>
<tr>
<th>PROCESS NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name: GENERATE USER PR</td>
</tr>
<tr>
<td>Process Number: 2-2-1-3.PN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.1.1.2.1.1.2.2.1</th>
<th>Read NEW-PR-PLAN-MONTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1.2.1.1.2.1.1</td>
<td>Go Back to 2.1.1.2.1.1</td>
</tr>
<tr>
<td>2.1.1.2.1.1.2.1.2</td>
<td>Otherwise (Plan Month OK)</td>
</tr>
<tr>
<td>2.1.1.2.1.1.2.1.1</td>
<td>Write PR-REC in PR RECORDS using PR-REC-NUMBER as key.</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: DELETE PR
Process Number: 2-2-1-4.PN

1. Read DELETED-PR
   1.1 For each DELETED-PR entered
      1.1.1 Read PR-REC in PR RECORDS
           with PR-REC-NUMBER = DELETED-PR-NUMBER
      1.1.1.1 If PR-AVAILABLE-TO-PROMISE = PR-REC-UNITS
      1.1.1.1.1 Then delete PR-REC in PR RECORDS
                   using DELETED-PR-NUMBER as key.
PROCESS NARRATIVE

Process Name: GENERATE PR AND RM-PR REPORT

Process Number: 2-2-2-1.PN

1. Read PR-REC in PR RECORDS

1.1 For each PR-REC

1.1.1 Set PR-REP-PR = PR-REC-NUMBER
    PR-REP-TYPE = PR-REC-TYPE
    PR-REP-SUBCODE = PR-REC-SUBCODE
    PR-REP-UNITS = UNITS-STILL-NOT-PURCHASED
    PR-REP-WHEN-PLANNED = PR-PLAN-MONTH
    PR-REP-WHEN-NEEDED = PR-ORIG-MONTH

1.1.2 Set PR-REP-BOOK-LEVEL =
    1 - (( PR-AVAILABLE-TO-PROMISE )/( PR-REC-UNITS ))

1.1.3 Read PROD-REC using
    TYPE = PR-REC-TYPE
    and SUBCODE = PR-REC-SUBCODE

1.1.3.1 Set PR-REP-WHEN-PURCHASE = PR-REC-PLAN-MONTH
    - PURCHASE-LEAD-TIME
    PR-REP-DESCRIPTION = DESCRIPTION in PRO-REC

1.1.4 Write PR-REPORT

2. Read RM-PR in RAW MATERIAL PR

2.1 For each RM-PR

2.1.1 Set RM-PR-REP-RM-PR = RM-PR-NUMBER
    RM-PR-REP-TYPE = RM-PR-TYPE
    RM-PR-REP-SUBCODE = RM-PR-SUBCODE
    RM-PR-REP-UNITS = RM-PR-UNITS
    RM-PR-REP-WHEN-PLANNED = RM-PR-PLAN-MONTH
<table>
<thead>
<tr>
<th>PROCESS NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name: GENERATE PR AND RM-PR REPORT</td>
</tr>
<tr>
<td>Process Number: 2-2-2-1.PN</td>
</tr>
</tbody>
</table>

| 2.1.2 Read PROD-REC using TYPE = RM-PR-TYPE and SUBCODE = RM-PR-SUBCODE |
| 2.1.2.1 Set RM-PR-REP-WHEN-PURCHASE = RM-PR-PLAN-MONTH - PURCHASE-LEAD-TIME |
| RM-PR-REP-DESCRIPTION = DESCRIPTION in PRO-REC |
| 2.1.3 Write RM-PR-REPORT. |
### Process Narrative

**Process Name:** PURCHASE RAW MATERIALS  
**Process Number:** 2-2-2-2.PN  
**Page:** 01

1. **Read RM-PR-PURCHASED**  
   1.1 For each RM-PR-PURCHASED entered  
      1.1.1 Read RM-PR in RAW MATERIAL PR  
           using RM-PR-NUMBER = PURCHASED-RM-PR  
      1.1.2 Read PURCHASE-ORDER in INVENTORY SYSTEM using  
           PURCHASE-ORDER-NUMBER = PURCHASED-PURCHASE-ORDER  
           PURCHASE-ORDER-ITEM = PURCHASED-PURCHASE-ITEM  
           PURCHASE-ORDER-TYPE = RM-PR-TYPE  
           PURCHASE-ORDER-SUBCODE = RM-PR-SUBCODE  
      1.1.2.1.1 Set RM-PR-UNITS = PURCHASE-ORDER-QUANT  
           RM-PR-PURCHASE-ORDER = PURCHASE-ORDER-NUMBER  
           RM-PR-PURCHASE-ITEM = PURCHASE-ORDER-ITEM  
      1.1.3 Set EXPECTED-RECEIPT in RM-PR =  
           PURCHASED-EXPECTING-RECEIPT
1. Read PR-PURCHASED
   1.1 For each PR-PURCHASED entered
      1.1.1 Read PR-REC in PR RECORDS
           using PR-REC-NUMBER = PURCHASED-PR
      1.1.2 Read PURCHASE-ORDER in INVENTORY SYSTEM using
           PURCHASE-ORDER-NUMBER = PURCHASED-PURCHASE-ORDER
           PURCHASE-ORDER-ITEM = PURCHASED-PURCHASE-ITEM
           PURCHASE-ORDER-TYPE = PR-REC-TYPE
           PURCHASE-ORDER-SUBCODE = PR-REC-SUBCODE
      1.1.2.1 If PURCHASE-ORDER-QUANT =
                  UNITS-STILL-NOT-PURCHASED
                 Then
                 1.1.2.1.1 Set UNITS-STILL-NOT-PURCHASED = 0
                 1.1.2.1.2 Otherwise
                 1.1.2.1.2.1 If PURCHASE-ORDER-QUANT >
                               UNITS-STILL-NOT-PURCHASED
                                Then
                                1.1.2.1.2.1.1 Set
                                   PR-REC-UNITS = PR-REC-UNITS
                                   + PURCHASE-ORDER-QUANT
                                   - UNITS-STILL-NOT-PURCHASED
                                UNITS-STILL-NOT-PURCHASED = 0
PROCESS NARRATIVE

Process Name: PURCHASE SIGNS AND MATERIALS

Process Number: 2-2-2-3.PN Page: 02

\[ \text{PR-AVAILABLE-TO-PROMISE} = \]
\[ \text{PR-AVAILABLE-TO-PROMISE} \]
\[ + \text{PURCHASE-ORDER-QUANT} \]
\[ - \text{UNITS-STILL-NOT-PURCHASED} \]

1.1.2.1.2.1.2 Otherwise

(Quantity Purchased less than Needed)

1.1.2.1.2.1.2.1 Set

\[ \text{UNITS-STILL-NOT-PURCHASED} = \]
\[ \text{UNITS-STILL-NOT-PURCHASED} - \text{PURCHASE-ORDER-QUANT} \]

1.1.3 Write PR-REC in PR RECORDS using PR-REC-NUMBER as key

1.1.4 Set PURCHASED-UNITS in PR-PURCHASE-ORDER

\[ \text{PURCHASE-ORDER-QUANT} = \]

1.1.4.1 Write PR-PURCHASE-ORDER in PR-PURCHASE ORDER using PR-REC-NUMBER

\[ \text{PURCHASE-ORDER-NUMBER} \]
\[ \text{PURCHASE-ORDER-ITEM} \] as key.
PROCESS NARRATIVE

Process Name: RECEIVE SIGNS AND MATERIALS

Process Number: 2-2-3-1.PN

1. Read THIS-WEEK in THIS WEEK

2. Read PR-REC in PR RECORDS
   with UNITS-STILL-NOT-PURCHASED < PR-REC-UNITS

2.1 For each PR-REC

2.1.1 Read PR-PURCHASED-ORDER in PR-PURCHASE ORDER
   using PR-REC-NUMBER as key

2.1.1.1 For each PR-PURCHASE-ORDER

2.1.1.1.1 Read PURCHASE-ORDER in INVENTORY SYSTEM
   using PURCHASE-ORDER-NUMBER
   and PURCHASE-ORDER-ITEM as key

2.1.1.1.1.1 If PURCHASE-ORDER-RECEIVED in PURCHASE-ORDER
   is greater than RECEIVED-UNITS in PR-PURCHASE-ORDER

2.1.1.1.1.1.1 Then ( New shipment arrived since last time
                              this application was runned )

2.1.1.1.1.1.1 Set NEW-SHIPMENT = PURCHASE-ORDER-RECEIVED
                              - RECEIVED-UNITS

2.1.1.1.1.1.2 If PURCHASE-ORDER-RECEIVED
   in PURCHASE-ORDER =
   PURCHASED-UNITS in PR-PURCHASE-ORDER

2.1.1.1.1.2.1 Then Delete PR-PURCHASE-ORDER
   in PR-PURCHASE ORDER using
   PR-REC-NUMBER,
   PURCHASE-ORDER-NUMBER
   and PURCHASE-ORDER-ITEM as key
PROCESS NARRATIVE

Process Name: RECEIVE SIGNS AND MATERIALS

Process Number: 2-2-3-1.PN

Page: 02

2.1.1.1.1.1.2.2 Otherwise

2.1.1.1.1.1.2.2.1 Set RECEIVED-UNITS

in PR-PURCHASE-ORDER = PURCHASE-ORDER-RECEIVED

in PURCHASE-ORDER

2.1.1.1.1.1.2.2.2 Write PR-PURCHASE-ORDER

in PR-PURCHASE ORDER using

PR-REC-NUMBER,

PURCHASE-ORDER-NUMBER

and PURCHASE-ORDER-ITEM as key

2.1.1.1.1.1.3 If NEW-SHIPMENT is greater than

(PR-REC-UNITS - PR-AVAILABLE-TO-PROMISE)

2.1.1.1.1.1.3.1 Then

2.1.1.1.1.1.3.1.1 Set

OLD-AVAIL-TO-PROMISE = PR-AVAILABLE-TO-PROMISE

PR-AVAILABLE-TO-PROMISE = PR-REC-UNITS

- NEW-SHIPMENT

NEW-SHIPMENT = PR-REC-UNITS

- OLD-AVAIL-TO-PROMISE

UNITS-RECEIVED in PR-REC = UNITS-RECEIVED +

+ OLD-AVAIL-TO-PROMISE - PR-AVAILABLE-TO-PROMISE

2.1.1.1.1.1.3.1.2 Read ON-HAND-REC in ON HAND RECORDS

using ON-HAND-REC-TYPE = PR-REC-TYPE

and ON-HAND-REC-SUBCODE = PR-REC-SUBCODE
PROCESS NARRATIVE

Process Name: RECEIVE SIGNS AND MATERIALS

Process Number: 2-2-3-1.PN

2.1.1.1.1.1.3.1.2.1 Set ON-HAND-AVAILABLE-TO-PROMISE = OLD-AVAIL-TO-PROMISE - PR-AVAILABLE-TO-PROMISE

2.1.1.1.1.1.3.1.2.2 Write ON-HAND-REC in ON HAND RECORDS using ON-HAND-REC-TYPE and ON-HAND-REC-SUBCODE as key

2.1.1.1.1.1.4 If NEW-SHIPMENT = 0 Then Go to 2.1.2

2.1.1.1.1.1.5 Read DELIVERY-REC in DELIVERY with WEEK-NUMBER > = THIS-WEEK

2.1.1.1.1.1.5.1 Sort DELIVERY-REC by WEEK-NUMBER (The earliest the week, the highest the priority)

2.1.1.1.1.1.5.1.1 Select first DELIVERY-REC from sorted group

2.1.1.1.1.1.5.1.2 Read PR-ORDER-REC in PR ORDER using PR-ORDER-REC-PR = PR-REC-NUMBER

2.1.1.1.1.1.5.1.2.1 For each PR-ORDER-REC and While NEW-SHIPMENT > 0

2.1.1.1.1.1.5.1.2.1.1 If DISTRICT in PR-ORDER-REC-ORDER = DISTRICT in DELIVERY-REC

2.1.1.1.1.1.5.1.2.1.1.1 Then

2.1.1.1.1.1.5.1.2.1.1.1.1 Read ORDER-REC in ORDER RECORDS using ORDER-REC-NUMBER = PR-ORDER-REC-ORDER and ORDER-REC-ITEM = PR-ORDER-REC-ITEM
2.1.1.1.1.5.1.2.1.1.1.2 Read
ON-HAND-ORDER-REC in ON HAND-ORDER
using ON-HAND-ORDER-REC-ORDER = PR-ORDER-REC-ORDER
and ON-HAND-ORDER-REC-ITEM = PR-ORDER-REC-ITEM

2.1.1.1.1.5.1.2.1.1.1.2.1 If can't find
ON-HAND-ORDER-REC in ON HAND-ORDER

2.1.1.1.1.5.1.2.1.1.2.1.1 Set
ON-HAND-ORDER-REC-ORDER = PR-ORDER-REC-ORDER
ON-HAND-ORDER-REC-ITEM = PR-ORDER-REC-ITEM
ON-HAND-ORDER-REC-UNITS = 0

2.1.1.1.1.5.1.2.1.1.1.3 If
NEW-SHIPMENT = PR-ORDER-REC-UNITS

2.1.1.1.1.5.1.2.1.1.1.3.1 Then

2.1.1.1.1.5.1.2.1.1.1.3.1.1 Set
ON-HAND-ORDER-REC-UNITS = ON-HAND-ORDER-REC-UNITS + NEW-SHIPMENT

PROMISED-BY-PR in ORDER-REC = PROMISED-BY-PR
- NEW-SHIPMENT

PROMISED-BY-ON-HAND in ORDER-REC = PROMISED-BY-ON-HAND
+ NEW-SHIPMENT

UNITS-RECEIVED in PR-REC = UNITS-RECEIVED + NEW-SHIPMENT

NEW-SHIPMENT = 0
PROCESS NARRATIVE

Process Name: RECEIVE SIGNS AND MATERIALS

Process Number: 2-2-3-1.PN Page: 05

2.1.1.1.1.1.1.5.1.2.1.1.1.3.1.2 Delete

PR-ORDER-REC in PR-ORDER
using PR-ORDER-REC-ORDER, PR-ORDER-REC-ITEM
and PR-ORDER-REC-PR as key

2.1.1.1.1.1.1.5.1.2.1.1.1.3.2 Otherwise

2.1.1.1.1.1.1.5.1.2.1.1.1.3.2.1 If
NEW-SHIPMENT < PR-ORDER-REC-UNITS

2.1.1.1.1.1.1.5.1.2.1.1.1.3.2.1.1 Then

2.1.1.1.1.1.1.5.1.2.1.1.1.3.2.1.1.1 Set

ON-HAND-ORDER-REC-UNITS = ON-HAND-ORDER-REC-UNITS + NEW-SHIPMENT

PROMISED-BY-PR in ORDER-REC = PROMISED-BY-PR - NEW-SHIPMENT

PROMISED-BY-ON-HAND in ORDER-REC = PROMISED-BY-ON-HAND + NEW-SHIPMENT

UNITS-RECEIVED in PR-REC = UNITS-RECEIVED + NEW-SHIPMENT

PR-ORDER-REC-UNITS = PR-ORDER-REC-UNITS - NEW-SHIPMENT

NEW-SHIPMENT = 0
### PROCESS NARRATIVE

Process Name: RECEIVE SIGNS AND MATERIALS

Process Number: 2-2-3-1.PN  
Page: 06

<table>
<thead>
<tr>
<th>2.1.1.1.1.5.1.2.1.1.1.3.2.1.1.2 Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR-ORDER-REC in PR-ORDER</td>
</tr>
<tr>
<td>using PR-ORDER-REC-ORDER, PR-ORDER-REC-ITEM</td>
</tr>
<tr>
<td>and PR-ORDER-REC-PR as key</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.1.1.1.1.1.5.1.2.1.1.1.3.2.1.2 Otherwise</th>
</tr>
</thead>
<tbody>
<tr>
<td>( New Shipment will satisfy more Orders )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.1.1.1.1.1.5.1.2.1.1.1.3.2.1.2.1 Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON-HAND-ORDER-REC-UNITS = ON-HAND-ORDER-REC-UNITS + PR-ORDER-REC-UNITS</td>
</tr>
<tr>
<td>PROMISED-BY-PR in ORDER-REC = PROMISED-BY-PR - PR-ORDER-REC-UNITS</td>
</tr>
<tr>
<td>PROMISED-BY-ON-HAND in ORDER-REC = PROMISED-BY-ON-HAND + PR-ORDER-REC-UNITS</td>
</tr>
</tbody>
</table>

| UNITS-RECEIVED in PR-REC = UNITS-RECEIVED + PR-ORDER-REC-UNITS |
| PR-ORDER-REC-UNITS = PR-ORDER-REC-UNITS - PR-ORDER-REC-UNITS |
| NEW-SHIPMENT = NEW-SHIPMENT - PR-ORDER-REC-UNITS |

<table>
<thead>
<tr>
<th>2.1.1.1.1.1.5.1.2.1.1.1.3.2.1.2.2 Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR-ORDER-REC in PR-ORDER</td>
</tr>
<tr>
<td>using PR-ORDER-REC-ORDER, PR-ORDER-REC-ITEM</td>
</tr>
<tr>
<td>and PR-ORDER-REC-PR as key</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: RECEIVE SIGNS AND MATERIALS

Process Number: 2-2-3-1.PN

Page: 07

2.1.1.1.1.1.5.1.2.1.1.4 Write

ON-HAND-ORDER-REC in ON HAND-ORDER
using ON-HAND-ORDER-REC-ORDER,
ON-HAND-ORDER-REC-TYPE
and ON-HAND-ORDER-REC-SUBCODE as key

ORDER-REC in ORDER RECORDS
using ORDER-REC-NUMBER
and ORDER-REC-ITEM as key

2.1.1.1.1.5.1.3 If NEW-SHIPMENT > 0

2.1.1.1.1.5.1.3.1 Then Unselect DELIVERY-REC from
sorted group

2.1.1.1.1.5.1.3.2 Go Back to 2.1.1.1.1.5.1.1
( Select a new Delivery Record )

2.1.2 If PR-REC-UNITS = UNITS-RECEIVED

2.1.2.1 Then Delete PR-REC in PR RECORDS using
PR-REC-NUMBER as key

2.1.2.2 Otherwise Write PR-REC in PR RECORDS using
PR-REC-NUMBER as key.
1. Read RM-PR in RAW MATERIAL PR with RM-PURCHASE-ORDER not "Null"

1.1 For each RM-PR

1.1.1 Read PURCHASE-ORDER in INVENTORY SYSTEM using PURCHASE-ORDER-NUMBER = RM-PURCHASE-ORDER and PURCHASE-ORDER-ITEM = RN-PURCHASE-ITEM

1.1.1.1 If PURCHASE-ORDER-RECEIVED in PURCHASE-ORDER is greater than UNITS-RECEIVED in RM-PR

1.1.1.1.1 Then (New shipment arrived since last time this application was runned)

1.1.1.1.1.1 Set NEW-SHIPMENT = PURCHASE-ORDER-RECEIVED - UNITS-RECEIVED

1.1.1.1.2 If RM-PR-UNITS = NEW-SHIPMENT

1.1.1.1.2.1 Then Delete RM-PR in RAW MATERIAL PR using RM-PR-NUMBER as key

1.1.1.1.2.2 Otherwise

1.1.1.1.2.2.1 Set UNITS-RECEIVED in RM-PR = PURCHASE-ORDER-RECEIVED in PURCHASE-ORDER

\[
\text{RM-PR-UNITS} = \text{RM-PR-UNITS} - \text{NEW-SHIPMENT}
\]

1.1.1.1.2.2.2 Write RM-PR in RAW MATERIAL PR using RM-PR-NUMBER as key.
PROCESS NARRATIVE

Process Name: IDENTIFY AVAILABLE SCHD CAPACITY

Process Number: 2-3-1-1.PN

Page: 01

1. Read DAYS-IN-WEEK

2. Read RESOURCE-PLAN-REC in RESOURCES PLAN RECS
   with PLAN-MONTH = MONTH

2.1 For each RESOURCE-PLAN-REC

2.1.1 Set
   RESOURCE-HOURS-AVAILABLE in RESOURCE-SCHD-CAPACITY =
   DAYS-IN-WEEK *
   ( RESOURCE-QUANT in RESOURCE-PLAN-REC ) *
   ( RESOURCE-WORK-HOURS in RESOURCE-PLAN-REC )

   RESOURCE-NUMBER in RESOURCE-SCHD-CAPACITY =
   RESOURCE-NUMBER in RESOURCE-PLAN-REC

   RESOURCE-WORK-HOURS in RESOURCE-SCHD-CAPACITY =
   RESOURCE-WORK-HOURS in RESOURCE-PLAN-REC

   RESOURCE-QUANT in RESOURCE-SCHD-CAPACITY =
   RESOURCE-QUANT in RESOURCE-PLAN-REC

   RESOURCE-EFFICIENCY in RESOURCE-SCHD-CAPACITY =
   RESOURCE-EFFICIENCY in RESOURCE-PLAN-REC

2.1.2 Read RESOURCE-REC in RESOURCES
   using RESOURCE-NUMBER as key

2.1.2.1 Set
   RESOURCE-DESCRIPTION in RESOURCE-SCHD-CAPACITY =
   RESOURCE-DESCRIPTION in RESOURCE-REC

2.1.3 Write RESOURCE-SCHD-CAPACITY
### PROCESS NARRATIVE

**Process Name:** IDENTIFY AVAILABLE SCHD CAPACITY  
**Process Number:** 2-3-1-1.PN  
**Page:** 02

<table>
<thead>
<tr>
<th>2.1.4 Ask User Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.4.1 If User Option is</td>
</tr>
<tr>
<td>&quot; Change Resource Schedule Capacity &quot;</td>
</tr>
<tr>
<td>2.1.4.1.1 Then Read RES-CHANGE-WORK-HOURS</td>
</tr>
<tr>
<td>RES-CHANGE-QUANT</td>
</tr>
<tr>
<td>RES-CHANGE-EFFICIENCY</td>
</tr>
<tr>
<td>2.1.4.1.1.1 If RES-CHANGE-WORK-HOURS is entered</td>
</tr>
<tr>
<td>2.1.4.1.1.1.1 Then Set</td>
</tr>
<tr>
<td>RESOURCE-WORK-HOURS = RES-CHANGE-WORK-HOURS</td>
</tr>
<tr>
<td>2.1.4.1.1.2 If RES-CHANGE-QUANT is entered</td>
</tr>
<tr>
<td>2.1.4.1.1.2.1 Then Set</td>
</tr>
<tr>
<td>RESOURCE-QUANT = RES-CHANGE-QUANT</td>
</tr>
<tr>
<td>2.1.4.1.1.3 If RES-CHANGE-EFFICIENCY</td>
</tr>
<tr>
<td>2.1.4.1.1.3.1 Then Set</td>
</tr>
<tr>
<td>RES-EFFICIENCY in RESOURCE-SCHD-REC = RES-CHANGE-EFFICIENCY</td>
</tr>
<tr>
<td>2.1.4.1.2 Set RES-HOURS-AVAILABLE in RESOURCE-SCHD-REC =</td>
</tr>
<tr>
<td>DAYS-IN-WEEK * RESOURCE-QUANT * RESOURCE-WORK-HOURS</td>
</tr>
<tr>
<td>2.1.4.1.3 Write RESOURCE-SCHD-REC in RESOURCE SCHD RECS using RESOURCE-NUMBER as key</td>
</tr>
</tbody>
</table>

3. Go To Process SORT PO TO SCHEDULE.
1. Read THIS-MONTH from THIS MONTH
1.1 Set MONTH = THIS-MONTH

2. Read THIS-WEEK from THIS-WEEK
2.1 Read DELIVERY-REC in DELIVERY
   using WEEK-NUMBER = THIS-WEEK as key
2.1.1 For each DISTRICT
2.1.1.1 Set NEXT-DELIVERY-WEEK = MIN( WEEK-NUMBER )
       using DISTRICT as index
2.1.1.2 Set DELIVERY-PRIORITY = ( NEXT-DELIVERY-WEEK ) -
       ( THIS-WEEK ) + 1
       using DISTRICT as index
2.2 Sort DISTRICT by DELIVERY-PRIORITY
   ( The smaller the number, the higher is the priority )
2.2.1 For each DISTRICT
2.2.1.1 Set DISTRICT-DELIVER-LIST-NUMBER = DISTRICT

   DISTRICT-DELIVER-LIST-PRIORITY =
   DELIVERY-PRIORITY

2.3 Write DISTRICT-DELIVER-LIST

3. Read User Option
3.1 If user option is " Enter new district delivery list "

PROCESS NARRATIVE

Process Name: SORT PO TO SCHEDULE

Process Number: 2-3-1-2.PN
### PROCESS NARRATIVE

**Process Name:** SORT PO TO SCHEDULE  
**Process Number:** 2-3-1-2.PN  
**Page:** 02

#### 3.1.1 Then

1. **3.1.1.1** Read DELIVERY-DISTRICTS-LIST

2. **3.1.1.1.2** For each DISTRICT in DELIVERY-DISTRICTS-LIST entered

3. **3.1.1.1.2.1** Set DELIVERY-PRIORITY = USER-DELIVERY-PRIORITY using DISTRICT as index

#### 4. Read PO-REC in PO RECORDS

- with PO-REC-STATUS not equal "Scheduled"
- and PO-PLAN-MONTH < = MONTH

1. **4.1** Set PROMISED-TO-NEXT-DELIVERY = 0  
   - PROMISED-TO-SECOND-DELIVERY = 0  
   - PROMISED-TO-THIRD-DELIVERY = 0  
   - PROMISED-TO-FOURTH-DELIVERY = 0

2. **4.2** For each PO-REC

   1. **4.2.1** Read all PO-ORDER-REC in PO-ORDER
      - with PO-ORDER-REC-PO = PO-REC-NUMBER

2. **4.2.1.1** For each PO-ORDER-REC

   1. **4.2.1.1.1** Set PO-ORDER-REC-DELIVERY-PRIORITY = DELIVERY-PRIORITY(DISTRICT) using DISTRICT in PO-ORDER-REC-ORDER as index

2. **4.2.1.1.1.1** If PO-ORDER-REC-DELIVERY-PRIORITY = 1

   1. **4.2.1.1.1.1.1** Then
4.2.1.1.1.1.1 Set PROMISED-TO-NEXT-DELIVERY =
  PROMISED-TO-NEXT-DELIVERY + PO-ORDER-REC-UNITS
  using PO-REC-NUMBER as index

4.2.1.1.1.2 If PO-ORDER-REC-DELIVERY-PRIORITY = 2
4.2.1.1.1.2.1 Then
4.2.1.1.1.2.1.1 Set PROMISED-TO-SECOND-DELIVERY =
  PROMISED-TO-SECOND-DELIVERY + PO-ORDER-REC-UNITS
  using PO-REC-NUMBER as index

4.2.1.1.1.3 If PO-ORDER-REC-DELIVERY-PRIORITY = 3
4.2.1.1.1.3.1 Then
4.2.1.1.1.3.1.1 Set PROMISED-TO-THIRD-DELIVERY =
  PROMISED-TO-THIRD-DELIVERY + PO-ORDER-REC-UNITS
  using PO-REC-NUMBER as index

4.2.1.1.1.4 If PO-ORDER-REC-DELIVERY-PRIORITY = 4
4.2.1.1.1.4.1 Then
4.2.1.1.1.4.1.1 Set PROMISED-TO-FOURTH-DELIVERY =
  PROMISED-TO-FOURTH-DELIVERY + PO-ORDER-REC-UNITS
  using PO-REC-NUMBER as index
PROCESS NARRATIVE

Process Name: SORT PO TO SCHEDULE

Process Number: 2-3-1-2.PN

Page: 04

4.3 Set \( \text{SCHD-PRIORITY-RATIO-I} = \frac{(\text{PROMISED-TO-NEXT-DELIVERY})}{(\text{PO-REC-UNITS})} \)
using PO-REC-NUMBER as index

\( \text{SCHD-PRIORITY-RATIO-II} = \frac{(\text{PROMISED-TO-SECOND-DELIVERY})}{(\text{PO-REC-UNITS})} \)
using PO-REC-NUMBER as index

\( \text{SCHD-PRIORITY-RATIO-III} = \frac{(\text{PROMISED-TO-THIRD-DELIVERY})}{(\text{PO-REC-UNITS})} \)
using PO-REC-NUMBER as index

\( \text{SCHD-PRIORITY-RATIO-IV} = \frac{(\text{PROMISED-TO-FOURTH-DELIVERY})}{(\text{PO-REC-UNITS})} \)
using PO-REC-NUMBER as index

5. Sort PO-REC by \( \text{SCHD-PRIORITY-RATIO-I} \) (descendent)
   by \( \text{SCHD-PRIORITY-RATIO-II} \) (descendent)
   by \( \text{SCHD-PRIORITY-RATIO-III} \) (descendent)
   by \( \text{SCHD-PRIORITY-RATIO-IV} \) (descendent)
   by PO-ORIG-MONTH (ascendent)
in this order

5.1 Set \( \text{STANDARD-PRIORITY} = 1 \)
\( \text{SPECIAL-PRIORITY} = 1 \)
\( \text{INTERSTATE-PRIORITY} = 1 \)

5.2 Select PO-REC with higher priority in sorted group
5.2.1 For each PO-REC
5.2.1.1 Read PROD-REC in PRODUCTS
   using TYPE = PO-REC-TYPE
   and SUBCODE = PO-REC-SUBCODE
5.2.1.1.1 If PROD-TYPE = " Standard "

5.2.1.1.1.1 Then

5.2.1.1.1.1.1 Set PO-PRIORITY = STANDARD-PRIORITY
PO-CATEGORY = " Standard "
PO-NUMBER = PO-REC-NUMBER
STANDARD-PRIORITY = STANDARD-PRIORITY + 1

5.2.1.1.1.1.2 Write PO-TO-SCHEDULE in PO TO SCHEDULE

5.2.1.1.1.1.3 Unselect PO-REC from sorted group

5.2.1.1.2 If PROD-TYPE = " Special "

5.2.1.1.2.1 Then

5.2.1.1.2.1.1 If PO-AVAILABLE-TO-PROMISE = PO-REC-UNITS

5.2.1.1.2.1.1.1 Then ( PO is still not promised )

5.2.1.1.2.1.1.1.1 Unselect PO-REC from sorted group

5.2.1.1.2.1.1.2 Otherwise ( PO is promised to someone )

5.2.1.1.2.1.1.2.1 Set PO-PRIORITY = SPECIAL-PRIORITY
PO-CATEGORY = " Special "
PO-NUMBER = PO-REC-NUMBER
SPECIAL-PRIORITY = SPECIAL-PRIORITY + 1

5.2.1.1.2.1.1.2.2 Write PO-TO-SCHEDULE in PO TO SCHEDULE

5.2.1.1.2.1.1.2.3 Unselect PO-REC from sorted group
5.2.1.1.3 If PROD-TYPE = "Interstate"

5.2.1.1.3.1 Then

5.2.1.1.3.1.1 If PO-AVAILABLE-TO-PROMISE = PO-REC-UNITS

5.2.1.1.3.1.1 Then (PO is still not promised)

5.2.1.1.3.1.1.1 Unselect PO-REC from sorted group

5.2.1.1.3.1.1.2 Otherwise (PO is promised to someone)

5.2.1.1.3.1.1.2.1 Set PO-PRIORITY = INTERSTATE-PRIORITY

PO-CATEGORY = "Interstate"

PO-NUMBER = PO-REC-NUMBER

INTERSTATE-PRIORITY = INTERSTATE-PRIORITY + 1

5.2.1.1.3.1.1.2.2 Write PO-TO-SCHEDULE in PO TO SCHEDULE

5.2.1.1.3.1.1.3 Unselect PO-REC from sorted group

6. Read PO-TO-SCHEDULE

with PO-CATEGORY = "Standard"

6.1 For each PO-TO-SCHEDULE

6.1.1 Read PO-REC in PO RECORDS

with PO-REC-NUMBER = PO-NUMBER

6.1.2 Set STD-LIST-PO-NUMBER = PO-NUMBER

STD-LIST-TYPE = PO-REC-TYPE

STD-LIST-SUBCODE = PO-REC-SUBCODE

STD-LIST-UNITS = PO-REC-UNITS

STD-LIST-PLAN-MONTH = PO-PLAN-MONTH
STD-LIST-ORIG-MONTH = PO-ORIG-MONTH

STD-LIST-BOOK-LEVEL =
1 - ( PO-AVAILABLE-TO-PROMISE ) / ( PO-REC-UNITS )

6.2 Write STANDARD-PRIORITY-LIST

7. Read PO-TO-SCHEDULE
with PO-CATEGORY = " Special "

7.1 For each PO-TO-SCHEDULE

7.1.1 Read PO-REC in PO RECORDS
with PO-REC-NUMBER = PO-NUMBER

7.1.2 Set SPC-LIST-PO-NUMBER = PO-NUMBER
SPC-LIST-TYPE = PO-REC-TYPE
SPC-LIST-SUBCODE = PO-REC-SUBCODE
SPC-LIST-UNITS = PO-REC-UNITS
SPC-LIST-PLAN-MONTH = PO-PLAN-MONTH
SPC-LIST-ORIG-MONTH = PO-ORIG-MONTH

SPC-LIST-BOOK-LEVEL =
1 - ( PO-AVAILABLE-TO-PROMISE ) / ( PO-REC-UNITS )

7.2 Write SPECIAL-PRIORITY-LIST

8. Read PO-TO-SCHEDULE
with PO-CATEGORY = " Interstate "

8.1 For each PO-TO-SCHEDULE

8.1.1 Read PO-REC in PO RECORDS
with PO-REC-NUMBER = PO-NUMBER
PROCESS NARRATIVE

Process Name: SORT PO TO SCHEDULE

Process Number: 2-3-1-2.PN

8.1.2 Set INT-LIST-PO_NUMBER = PO_NUMBER
   INT-LIST-TYPE = PO-REC-TYPE
   INT-LIST-SUBCODE = PO-REC-SUBCODE
   INT-LIST-UNITS = PO-REC-UNITS
   INT-LIST-PLAN-MONTH = PO-PLAN-MONTH
   INT-LIST-ORIG-MONTH = PO-ORIG-MONTH
   INT-LIST-BOOK-LEVEL = 1 - (PO-AVAILABLE-TO-PROMISE) / (PO-REC-UNITS)

8.2 Write INTERSTATE-PRIORITY-LIST

9. Ask user option

9.1 If user option is "Change Priority to Schedule"

9.1.1 Read SCHD-PRIORITY

9.1.1.1 For each SCHD-PRIORITY entered

9.1.1.1.1 Read PO-TO-SCHEDULE in PO TO SCHEDULE
   with PO_NUMBER = SCHD-PRIORITY-PO_NUMBER

9.1.1.2 Set PO-PRIORITY = USER-SCHD-PRIORITY

9.1.1.3 Write PO-TO-SCHEDULE in PO-TO-SCHEDULE
   using PO-NUMBER as key

10. Go To Process CHECK MATERIAL AVAILABILITY
1. Read all PO-TO-SCHEDULE in PO TO SCHEDULE

2. Sort PO-TO-SCHEDULE by PO-PRIORITY

2.1 Select PO-TO-SCHEDULE with highest priority in sorted group

2.1.1 For each PO-TO-SCHEDULE

2.1.1.1 Read PO-REC in PO RECORDS
   with PO-REC-NUMBER = PO-NUMBER

2.1.1.2 Read BOM-REC in BILL OF MATERIALS
   using PARENT-TYPE = PO-REC-TYPE
   and PARENT-SUBCODE = PO-REC-SUBCODE

2.1.1.2.1 For each BOM-REC

2.1.1.2.1.1 Read RM-AVAILABLE in RM AVAILABLE
   using RM-TYPE = SON-TYPE
   and RM-SUBCODE = SON-SUBCODE

2.1.1.2.1.1.1 If can't find RM-AVAILABLE

2.1.1.2.1.1 Then

2.1.1.2.1.1.1 Read ON-HAND in INVENTORY SYSTEM
   using TYPE = SON-TYPE
   SUBCODE = SON-SUBCODE
   and DISTRICT = " Sign Shop "

2.1.1.2.1.1.2 Set RM-TYPE = TYPE
   RM-SUBCODE = SUBCODE
   RM-UNITS = ON-HAND-QUANT
PROCESS NARRATIVE

Process Name: CHECK RM AVAILABILITY

Process Number: 2-3-1-3.PN

2.1.1.2.1.2 Set RM-UNITS = RM-UNITS - (PO-REC-UNITS * SON-QUANT)

2.1.1.2.1.2.1 If RM-UNITS < 0

2.1.1.2.1.2.1.1 Then (No material is available for PO)

2.1.1.2.1.2.1.1.1 Delete PO-TO-SCHEDULE in PO TO SCHEDULE using PO-NUMBER as key

2.1.1.2.1.2.1.1.2 Delete PO-TO-SCHEDULE from sorted group

2.1.1.2.1.2.1.1.3 Go Back to 2.1 (Select a new PO)

2.1.1.2.1.2.1.2 Otherwise

2.1.1.2.1.2.1.2.1 Write RM-AVAILABLE in RM AVAILABLE using RM-TYPE and RM-SUBCODE as key

2.1.2 Set PO-STATUS = "Material Checked"

3. Delete all Records in RM AVAILABLE

4. Call Process LOAD PO AT NON-SHARED RESOURCE.
<table>
<thead>
<tr>
<th>Process Name: LOAD PO AT NON-SHARED RESOURCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Number: 2-3-1-4-1.PN Page: 01</td>
</tr>
</tbody>
</table>

1. Read PO-TO-SCHEDULE in PO TO PRODUCTION with PO-STATUS = "Material Checked"

1.1 Sort PO-TO-SCHEDULE by PO-PRIORITY

1.1.1 Select PO-TO-SCHEDULE with highest priority in sorted group

1.1.1.1 Read PO-REC in PRODUCTS RECORDS with PO-REC-NUMBER = PO-NUMBER

1.1.1.1.1 Read PROD-ROUTE in ROUTES using
    TYPE = PO-REC-TYPE
    SUBCODE = PO-REC-SUBCODE
    and ROUTE-TYPE = "Non-shared"

1.1.1.1.1 For each PROD-ROUTE

1.1.1.1.1.1 Read OPERATION-REC in OPERATIONS using OPERATION-CODE as key

1.1.1.1.1.1 Read RESOURCE-SCHD-REC in RESOURCE PLAN RECS using OPERATION-MACHINE = RESOURCE-NUMBER

1.1.1.1.1.1.1 If (WORK-LOAD / RES-HOURS-AVAILABLE) is less than RES-EFFICIENCY

1.1.1.1.1.1.1 Then (There is no capacity available)

1.1.1.1.1.1.1 Delete PO-TO-SCHEDULE in PO TO SCHEDULE using PO-NUMBER as key
PROCESS NARRATIVE

Process Name: LOAD PO AT NON-SHARED RESOURCES

Process Number: 2-3-1-4-1.PN

1.1.1.1.1.1.1.1.2 Unselect PO-TO-SCHEDULE from sorted group

1.1.1.1.1.1.1.1.3 Go back to 1.1.1
   (Select a new PO to Schedule)

1.1.1.1.1.1.2 Read RESOURCE-SCHD-REC in RESOURCE PLAN RECS
   using OPERATION-WORKER = RESOURCE-NUMBER

1.1.1.1.1.1.2.1 If (WORK-LOAD / RES-HOURS-AVAILABLE)
   is less than RES-EFFICIENCY

1.1.1.1.1.1.2.1.1 Then (There is no capacity available)

1.1.1.1.1.1.2.1.1 Delete PO-TO-SCHEDULE
   in PO TO SCHEDULE
   using PO-NUMBER as key

1.1.1.1.1.1.2.1.2 Unselect PO-TO-SCHEDULE from sorted group

1.1.1.1.1.1.2.1.3 Go back to 1.1.1
   (Select a new PO to Schedule)

1.1.1.2 Read PO-REC in PRODUCTS RECORDS
   with PO-REC-NUMBER = PO-NUMBER

1.1.1.2.1 Read PROD-ROUTE in ROUTES using
   TYPE = PO-REC-TYPE
   SUBCODE = PO-REC-SUBCODE
   and ROUTE-TYPE = "Non-shared"

1.1.1.2.1.1 For each PROD-ROUTE
**PROCESS NARRATIVE**

Process Name: LOAD PO AT NON-SHARED RESOURCES

Process Number: 2-3-1-4-1.PN

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.2.1.1.1</td>
<td>Read OPERATION-REC in OPERATIONS using OPERATION-CODE as key</td>
</tr>
<tr>
<td>1.1.1.2.1.1.1</td>
<td>Read RESOURCE-SCHD-REC in RESOURCE PLAN RECS using OPERATION-MACHINE = RESOURCE-NUMBER</td>
</tr>
<tr>
<td>1.1.1.2.1.1.1.1</td>
<td>Set WORK-LOAD = WORK-LOAD + SET-UP-TIME + ( OP-TIME-PER-UNIT * PO-REC-UNITS )</td>
</tr>
<tr>
<td>1.1.1.2.1.1.1.2</td>
<td>Write RESOURCE-SCHD-REC in RESOURCE SCHD RECS using RESOURCE-NUMBER as key</td>
</tr>
<tr>
<td>1.1.1.2.1.1.2</td>
<td>Read RESOURCE-SCHD-REC in RESOURCE SCHD RECS using OPERATION-WORKER = RESOURCE-NUMBER</td>
</tr>
<tr>
<td>1.1.1.2.1.1.2.1</td>
<td>Set WORK-LOAD = WORK-LOAD + SET-UP-TIME + ( OP-TIME-PER-UNIT * PO-REC-UNITS )</td>
</tr>
<tr>
<td>1.1.1.2.1.1.2.2</td>
<td>Write RESOURCE-SCHD-REC in RESOURCE SCHD RECS using RESOURCE-NUMBER as key</td>
</tr>
<tr>
<td>1.1.2</td>
<td>Set PO-STATUS = &quot; Loaded at Non-Shared &quot;</td>
</tr>
<tr>
<td>1.1.3</td>
<td>Write PO-TO-SCHEDULE in PO TO SCHEDULE using PO-NUMBER as key</td>
</tr>
<tr>
<td>1.1.4</td>
<td>Unselect PO-TO-SCHEDULE from sorted group</td>
</tr>
<tr>
<td>1.1.5</td>
<td>Go back to !.1.1 ( Select new PO to Schedule )</td>
</tr>
<tr>
<td>2.</td>
<td>Go To Process LOAD PO AT SHARED RESOURCES</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: LOAD PO AT SHARED RESOURCES

Process Number: 2-3-1-4-2.PN

Page: 01

1. Read PO-TO-SCHEDULE in PO TO PRODUCTION with PO-STATUS = "Loaded at Non-Shared"

1.1 For each PO-TO-SCHEDULE

1.1.1 Read PO-REC in PRODUCTS RECORDS with PO-REC-NUMBER = PO-NUMBER

1.1.1.1 Read PROD-ROUTE in ROUTES using
   TYPE = PO-REC-TYPE
   SUBCODE = PO-REC-SUBCODE
   and ROUTE-TYPE = "Shared"

1.1.1.1.1 For each PROD-ROUTE

1.1.1.1.1.1 Read OPERATION-REC in OPERATIONS using OPERATION-CODE as key

1.1.1.1.1.1 Read RESOURCE-SCHD-REC in RESOURCE PLAN RECS using OPERATION-MACHINE = RESOURCE-NUMBER

1.1.1.1.1.1.1 Set WORK-LOAD = WORK-LOAD +
   SET-UP-TIME + ( OP-TIME-PER-UNIT * PO-REC-UNITS )

1.1.1.1.1.1.2 Write RESOURCE-SCHD-REC in RESOURCE SCHD RECS using RESOURCE-NUMBER as key

1.1.1.1.1.2 Read RESOURCE-SCHD-REC in RESOURCE SCHD RECS using OPERATION-WORKER = RESOURCE-NUMBER

1.1.1.1.1.2.1 Set WORK-LOAD = WORK-LOAD +
   SET-UP-TIME + ( OP-TIME-PER-UNIT * PO-REC-UNITS )
PROCESS NARRATIVE

Process Name: LOAD PO AT SHARED RESOURCES

Process Number: 2-3-1-4-2.PN

1.1.1.1.1.1.2.2 Write RESOURCE-SCHD-REC in RESOURCE SCHD RECS using RESOURCE-NUMBER as key

1.2 Set PO-STATUS = "Loaded at Shared"

1.3 Write PO-TO-SCHEDULE in PO TO SCHEDULE using PO-NUMBER as key

2. Go To Process REPORT SCHED WORK LOAD
1. Read RESOURCE-SCHD-REC in RESOURCE SCHD RECS
   1.1 For each RESOURCE-SCHD-REC
      1.1.1 Set RATIO = (WORK-LOAD) / (RES-HOURS-AVAILABLE)
      1.1.1.1 If RATIO > RES-EFFICIENCY Then
         1.1.1.1.1 Set RES-STATUS = "Bottleneck"
      1.1.1.2 Otherwise
         1.1.1.2.1 Set RES-STATUS = "Non-Bottleneck"
      1.1.1.2 Set REPORT-RES-NUMBER = RESOURCE-NUMBER
         REPORT-RES-HOURS-AVAILABLE = RESOURCE-HOURS-AVAILABLE
         REPORT-RES-EFFICIENCY = RES-EFFICIENCY
         REPORT-LOAD-RATIO = RATIO
         REPORT-RES-SATUS = RES-STATUS
      1.1.1.3 Read RESOURCE-REC in RESOURCES using RESOURCE-NUMBER as key
      1.1.1.3.1 Set REPORT-RES-DESCRIPTION = RESOURCE-DESCRIPTION
      1.1.1.4 Read THIS-MONTH from THIS MONTH
PROCESS NARRATIVE

Process Name: REPORT SCHD WORK LOAD

Process Number: 2-3-1-4-1.PN

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.4.1</td>
<td>Read RESOURCE-PLAN-REC in RESOURCE PLAN RECS using MONTH = THIS-MONTH and RESOURCE-NUMBER as key</td>
</tr>
<tr>
<td>1.1.1.4.1.1</td>
<td>Set REPORT-RES-QUANT = RESOURCE-QUANT</td>
</tr>
<tr>
<td>1.1.1.5</td>
<td>Write RESOURCE-LOAD-REPORT</td>
</tr>
<tr>
<td>2.</td>
<td>Read PO-TO-SCHEDULE in PO TO SCHEDULE with PO-STATUS = &quot;Loaded at Shared&quot; and PO-CATEGORY = &quot;Standard&quot;</td>
</tr>
<tr>
<td>2.1</td>
<td>For each PO-TO-SCHEDULE</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Set REPORT-PO-NUMBER = PO-NUMBER</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Read PO-REC in PO RECORDS with PO-REC-NUMBER = PO-NUMBER</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Set REPORT-PO-TYPE = PO-REC-TYPE REPORT-PO-SUBCODE = PO-REC-SUBCODE REPORT-PO-UNITS = PO-REC-UNITS REPORT-PO-PLAN-MONTH = PO-PLAN-MONTH REPORT-PO-ORIG-MONTH = PO-ORIG-MONTH REPORT-PO-BOOK-LEVEL = 1 - (( PO-AVAILABLE-TO-PROMISE )/( PO-REC-UNITS ))</td>
</tr>
<tr>
<td>2.1.4</td>
<td>Write STD-PO-REPORT</td>
</tr>
<tr>
<td>3.</td>
<td>Read PO-TO-SCHEDULE in PO TO SCHEDULE with PO-STATUS = &quot;Loaded at Shared&quot; and PO-CATEGORY = &quot;Special&quot;</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: REPORT SCHD WORK LOAD

Process Number: 2-3-1-4-1.PN

Page: 03

3.1 For each PO-TO-SCHEDULE

3.1.1 Set REPORT-PO-NUMBER = PO-NUMBER

3.1.2 Read PO-REC in PO RECORDS
   with PO-REC-NUMBER = PO-NUMBER

3.1.3 Set REPORT-PO-TYPE = PO-REC-TYPE
   REPORT-PO-SUBCODE = PO-REC-SUBCODE
   REPORT-PO-UNITS = PO-REC-UNITS
   REPORT-PO-PLAN-MONTH = PO-PLAN-MONTH
   REPORT-PO-ORIG-MONTH = PO-ORIG-MONTH

   REPORT-PO-BOOK-LEVEL =
   1 - (( PO-AVAILABLE-TO-PROMISE )/( PO-REC-UNITS ))

3.1.4 Write SPC-PO-REPORT

4. Read PO-TO-SCHEDULE in PO TO SCHEDULE
   with PO-STATUS = " Loaded at Shared "
   and PO-CATEGORY = " Interstate "

4.1 For each PO-TO-SCHEDULE

4.1.1 Set REPORT-PO-NUMBER = PO-NUMBER

4.1.2 Read PO-REC in PO RECORDS
   with PO-REC-NUMBER = PO-NUMBER

4.1.3 Set REPORT-PO-TYPE = PO-REC-TYPE
   REPORT-PO-SUBCODE = PO-REC-SUBCODE
   REPORT-PO-UNITS = PO-REC-UNITS
   REPORT-PO-PLAN-MONTH = PO-PLAN-MONTH
   REPORT-PO-ORIG-MONTH = PO-ORIG-MONTH
REPORT-PO-BOOK-LEVEL =
1 - ((PO-AVAILABLE-TO-PROMISE)/(PO-REC-UNITS))

4.1.4 Write SPC-PO-REPORT

5. Ask User Option

5.1 If Option is "Make Changes in Schedule"

5.1.1 Then Go To Process DISCHARGE PO FROM LIST

5.2 If Option is "Approve Schedule"

5.2.1 Then Go To Process APPROVE SCHEDULE.
PROCESS NARRATIVE

Process Name: DISCHARGE PO FROM LIST

Process Number: 2-3-1-4-4.PN Page: 01

1. Ask User Option
   1.1 If Option is "Discharge Orders from Schd List"
      1.1.1 Then
         1.1.1.1 Read PO-DISCHARGED
         1.1.1.1.1 For each PO-DISCHARGED
            1.1.1.1.1.1 Read PO-TO-SCHEDULE in PO TO SCHEDULE
                        using PO-NUMBER as key
            1.1.1.1.1.1.1 Set PO-STATUS = "Discharged"
            1.1.1.1.1.2 Write PO-TO-SCHEDULE in PO TO SCHEDULE
                        using PO-NUMBER as key
   1.2 If Option is "Reconsider PO discharged"
      1.2.1 Ask User Option
      1.2.1.1 If Option is "Generate Discharged Orders Report"
         1.2.1.1.1 Then
            1.2.1.1.1.1 Read PO-TO-SCHEDULE in PO TO SCHEDULE
                        with PO-STATUS = "Discharged"
            1.2.1.1.1.1.1 For each PO-TO-SCHEDULE
            1.2.1.1.1.1.1.1 Set PO-DISCHARGED-REPORT = PO-NUMBER
PROCESS NARRATIVE

Process Name: DISCHARGE PO FROM LIST

Process Number: 2-3-1-4-4.PN Page: 02

1.2.1.1.1.1.2 Write PO-DISCHARGED-REPORT

1.2.1.2 If Option is "Enter PO to reconsider"

1.2.1.2.1 Read PO-RECONSIDERED

1.2.1.2.1.1 For each PO-RECONSIDERED

1.2.1.2.1.1.1 Read PO-TO-SCHEDULE in PO TO SCHEDULE using PO-NUMBER as key

1.2.1.2.1.1.1.1 Set PO-STATUS = "Loaded at Shared"

1.2.1.2.1.1.2 Write PO-TO-SCHEDULE in PO TO SCHEDULE using PO-NUMBER as key

2. Go To Process REPORT SCHD WORK LOAD.
PROCESS NARRATIVE

Process Name: APPROVE SCHEDULE

Process Number: 2-3-1-4-5.PN

1. Read PO-TO-SCHEDULE in PO TO SCHEDULE
   with PO-STATUS = "Loaded at Shared"
   and PO-CATEGORY = "Standard"

   1.1 For each PO-REC

   1.1.1 Read PO-REC in PO RECORDS
       with PO-REC-NUMBER = PO-NUMBER

   1.1.1.1 Read PROD-REC in PRODUCTS
       with TYPE = PO-REC-TYPE
       and SUBCODE = PO-REC-SUBCODE

   1.1.1.1.1 Set PO-FAMILY = PROD-FAMILY
       using PO-NUMBER as index

   1.1.1.1.2 Write PO-TO-SCHEDULE in PO-TO-SCHEDULE
       using PO-NUMBER as key

2. Read PO-TO-SCHEDULE in PO TO SCHEDULE
   with PO-STATUS = "Loaded at Shared"
   and PO-CATEGORY = "Standard"

   2.1 Sort PO-TO-SCHEDULE by PO-PRIORITY

   2.1.1 Select PO-TO-SCHEDULE with highest priority in
       sorted group

   2.1.1.1 Set FAMILY = PO-FAMILY
       PRIORITY = PO-PRIORITY

   2.1.1.2 Delete PO-TO-SCHEDULE from sorted group
PROCESS NARRATIVE

Process Name: APPROVE SCHEDULE

Process Number: 2-3-1-4-5.PN

2.1.1.3 Select PO-TO-SCHEDULE from sorted group with PO-FAMILY = FAMILY

2.1.1.3.1 If can't find PO-TO-SELECT
2.1.1.3.1.1 Then
2.1.1.3.1.1.1 Go Back to 2.1.1 (Select a new Family of PO)
2.1.1.3.1.2 Otherwise
2.1.1.3.1.2.1 For each PO-TO-SCHEDULE
2.1.1.3.1.2.1.1 Set PO-PRIORITY = PRIORITY
2.1.1.3.1.2.1.2 Write PO-TO-SCHEDULE in PO-TO-SCHEDULE using PO-NUMBER as key
2.1.1.3.1.2.1.3 Delete PO-TO-SCHEDULE from sorted group
2.1.1.3.1.2.1.4 Go Back to 2.1.1.3 (Select another PO of the same family)

3. Read PO-TO-SCHEDULE in PO TO SCHEDULE with PO-STATUS = "Loaded at Shared" and PO-CATEGORY = "Standard"

3.1 Sort PO-TO-SCHEDULE by PO-PRIORITY
3.1.1 Select PO-TO-SCHEDULE with highest priority in sorted group
3.1.1.1 Set LIST-PO-NUMBER = PO-NUMBER
### Process Narrative

**Process Name:** APPROVE SCHEDULE  
**Process Number:** 2-3-1-4-5.PN  
**Page:** 03

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3.1.1.2 | **Read PO-REC in PO RECORDS**  
with **PO-REC-NUMBER** = **PO-NUMBER** |
| 3.1.1.2.1 | **Set**  
**LIST-PO-TYPE** = **PO-REC-TYPE**  
**LIST-PO-SUBCODE** = **PO-REC-SUBCODE**  
**LIST-PO-UNITS** = **PO-REC-UNITS**  
**LIST-PO-BOOK-LEVEL** =  
1 - (( **PO-AVAILABLE-TO-PROMISE** )/ ( **PO-REC-UNITS** )) |
| 3.1.1.2.1.1 | **Read PRO-REC in PRODUCTS**  
with **TYPE** = **PO-REC-TYPE**  
and **SUBCODE** = **PO-REC-SUBCODE** |
| 3.1.1.2.1.1.1 | **Set**  
**LIST-PO-DESCRIPTION** =  
**DESCRIPTION in PROD-REC** |
| 3.1.2 | **Delete PO-TO-SCHEDULE in sorted group** |
| 3.1.2 | **Delete PO-TO-SCHEDULE in PO TO SCHEDULE**  
using **PO-NUMBER** as key |
| 3.1.3 | **Set PO-REC-STATUS** = "Scheduled" |
| 3.1.4 | **Write PO-REC in PO RECORDS**  
using **PO-REC** as key |
| 3.1.5 | **Write STD-SCHD-LIST** |
| 4. | **Read PO-TO-SCHEDULE in PO TO SCHEDULE**  
with **PO-STATUS** = "Loaded at Shared"  
and **PO-CATEGORY** = "Special" |
| 4.1 | **Sort PO-TO-SCHEDULE by PO-PRIORITY** |
PROCESS NARRATIVE

Process Name: APPROVE SCHEDULE

Process Number: 2-3-1-4-5.PN

4.1.1 Select PO-TO-SCHEDULE with highest priority in sorted group

4.1.1.1 Set LIST-PO-NUMBER = PO-NUMBER

4.1.1.2 Read PO-REC in PO RECORDS
   with PO-REC-NUMBER = PO-NUMBER

4.1.1.2.1 Set LIST-PO-TYPE = PO-REC-TYPE
   LIST-PO-SUBCODE = PO-REC-SUBCODE
   LIST-PO-UNITS = PO-REC-UNITS

   LIST-PO-BOOK-LEVEL =
   1 - (( PO-AVAILABLE-TO-PROMISE )/ ( PO-REC-UNITS ))

4.1.1.2.1.1 Read PRO-REC in PRODUCTS
   with TYPE = PO-REC-TYPE
   and SUBCODE = PO-REC-SUBCODE

4.1.1.2.1.1.1 Set LIST-PO-DESCRIPTION =
   DESCRIPTION in PROD-REC

4.1.2 Delete PO-TO-SCHEDULE in sorted group

4.1.2 Delete PO-TO-SCHEDULE in PO TO SCHEDULE
   using PO-NUMBER as key

4.1.3 Set PO-REC-STATUS = " Scheduled "

4.1.4 Write PO-REC in PO RECORDS
   using PO-REC as key

4.1.5 Write SPC-SCHD-LIST
5. Read PO-TO-SCHEDULE in PO TO SCHEDULE with PO-STATUS = " Loaded at Shared " and PO-CATEGORY = " Interstate "

5.1 Sort PO-TO-SCHEDULE by PO-PRIORITY

5.1.1 Select PO-TO-SCHEDULE with highest priority in sorted group

5.1.1.1 Set LIST-PO-NUMBER = PO-NUMBER

5.1.1.2 Read PO-REC in PO RECORDS with PO-REC-NUMBER = PO-NUMBER

5.1.1.2.1 Set LIST-PO-TYPE = PO-REC-TYPE
LIST-PO-SUBCODE = PO-REC-SUBCODE
LIST-PO-UNITS = PO-REC-UNITS

LIST-PO-BOOK-LEVEL =
1 - (( PO-AVAILABLE-TO-PROMISE )/( PO-REC-UNITS )

5.1.1.2.1.1 Read PRO-REC in PRODUCTS with TYPE = PO-REC-TYPE and SUBCODE = PO-REC-SUBCODE

5.1.1.2.1.1.1 Set LIST-PO-DESCRIPTION = DESCRIPTION in PROD-REC

5.1.2 Delete PO-TO-SCHEDULE in sorted group

5.1.2 Delete PO-TO-SCHEDULE in PO TO SCHEDULE using PO-NUMBER as key

5.1.3 Set PO-REC-STATUS = " Scheduled "
| 5.1.4 Write PO-REC in PO RECORDS  
using PO-REC as key |
| 5.1.5 Write INT-SCHD-LIST |
Process Name: COMPLETE PO

1. Read PO-COMPLETED

1.1 For each PO-COMPLETED

1.1.1 Read PO-REC in PO RECORDS
    with PO-REC-NUMBER = PO-COMPLETED-NUMBER

1.1.1.1 If PO-REC-UNITS = PO-COMPLETED-UNITS

1.1.1.1.1 Then

1.1.1.1.1.1 If PO-AVAILABLE-TO-PROMISE < PO-REC-UNITS

1.1.1.1.1.1.1 Then

1.1.1.1.1.1.1.1 Read PO-ORDER-REC in PO-ORDER
    with PO-ORDER-REC-PO = PO-COMPLETED-NUMBER

1.1.1.1.1.1.1.1 For each PO-ORDER-REC

1.1.1.1.1.1.1.1.1 Read ORDER-REC in ORDER RECORDS
    with PO-ORDER-REC-ORDER = ORDER-REC-ORDER
    and PO-ORDER-REC-ITEM = ORDER-REC-ITEM

1.1.1.1.1.1.1.1.2 Set PROMISED-BY-PO in ORDER =
    PROMISED-BY-PO - PO-ORDER-REC-UNITS

    PROMISED-BY-ON-HAND in ORDER =
    PROMISED-BY-ON-HAND + PO-ORDER-REC-UNITS

1.1.1.1.1.1.1.1.3 Read ON-HAND-ORDER-REC
    in ON-HAND-ORDER
    with ON-HAND-ORDER-REC-ORDER = ORDER-REC-ORDER
    and ON-HAND-ORDER-REC-ITEM = ORDER-REC-ITEM
1.1.1.1.1.1.1.1.3.1 If can’t find ON-HAND-ORDER-REC
1.1.1.1.1.1.1.1.3.1.1 Then Set

ON-HAND-ORDER-REC-ORDER = ORDER-REC-NUMBER
ON-HAND-ORDER-REC-ITEM = ORDER-REC-ITEM
ON-HAND-ORDER-REC-UNITS = 0

1.1.1.1.1.1.1.1.4 Set ON-HAND-ORDER-REC-UNITS =
    ON-HAND-ORDER-REC-UNITS + PO-ORDER-REC-UNITS

1.1.1.1.1.1.1.1.5 Write ON-HAND-ORDER-REC
    in ON-HAND-ORDER
    using ON-HAND-ORDER-REC-ORDER
    and ON-HAND-ORDER-REC-ITEM as key

1.1.1.1.1.1.1.1.6 Write ORDER-REC in ORDER RECORDS
    using ORDER-REC-NUMBER
    and ORDER-REC-ITEM as key

1.1.1.1.1.1.1.1.6 Delete PO-ORDER-REC in PO-ORDER
    using PO-ORDER-REC-ORDER
    PO-ORDER-REC-ITEM as key

1.1.1.1.1.2 If PO-AVAILABLE-TO-PROMISE < PO-REC-UNITS
    in ON-HAND-ORDER
    using ON-HAND-ORDER-REC-ORDER
    and ON-HAND-ORDER-REC-ITEM as key

1.1.1.1.1.2.1 If PO-AVAILABLE-TO-PROMISE > 0
1.1.1.1.1.2.1.1 Then
PROCESS NARRATIVE

Process Name: COMPLETE PO

1.1.1.1.1.2.1.1.1 If UNITS-REQ-ADJUSTMENT in PO-REC > 0
1.1.1.1.1.2.1.1.1 Then
1.1.1.1.1.2.1.1.1.1 If PO-AVAILABLE-TO-PROMISE > = UNITS-REQ-ADJUSTMENT
1.1.1.1.1.2.1.1.1.1 Then
1.1.1.1.1.2.1.1.1.1.1.1 Set PO-AVAILABLE-TO-PROMISE = PO-AVAILABLE-TO-PROMISE - UNITS-REQ-ADJUSTMENT
   UNITS-REQ-ADJUSTMENT = 0
1.1.1.1.1.2.1.1.1.1.2 Otherwise
   ( Available to Promise < Units Req Adjustment )
1.1.1.1.1.2.1.1.1.1.2.1 Set UNITS-REQ-ADJUSTMENT - PO-AVAILABLE-TO-PROMISE
   PO-AVAILABLE-TO-PROMISE = 0
1.1.1.1.1.3 If PO-AVAILABLE-TO-PROMISE > 0
1.1.1.1.1.3.1 Read ON-HAND-REC in ON HAND RECORDS using ON-HAND-REC-TYPE = PO-REC-TYPE and ON-HAND-REC-SUBCODE = PO-REC-SUBCODE
1.1.1.1.1.3.1.1 If UNITS-REQ-ADJUSTMENT in ON-HAND-REC > 0
1.1.1.1.1.3.1.1 Then
1.1.1.1.1.3.1.1.1 If PO-AVAILABLE-TO-PROMISE > = UNITS-REQ-ADJUSTMENT
PROCESS NARRATIVE

Process Name: COMPLETE PO
Process Number: 2-3-2.PN

1.1.1.1.3.1.1.1.1.1 Then
1.1.1.1.3.1.1.1.1 Set PO-AVAILABLE-TO-PROMISE = PO-AVAILABLE-TO-PROMISE - UNITS-REQ-ADJUSTMENT

UNITS-REQ-ADJUSTMENT = 0

1.1.1.1.3.1.1.1.2 Otherwise
( Available to Promise < Units Req Adjustment )
1.1.1.1.3.1.1.2.1 Set UNITS-REQ-ADJUSTMENT - PO-AVAILABLE-TO-PROMISE
PO-AVAILABLE-TO-PROMISE = 0

1.1.1.1.3.2 If PO-AVAILABLE-TO-PROMISE > 0
1.1.1.1.3.2.1 Then
1.1.1.1.3.2.1.1 Set ON-HAND-AVAILABLE-TO-PROMISE = ON-HAND-AVAILABLE-TO-PROMISE + PO-AVAILABLE-TO-PROMISE

1.1.1.1.3.3 If UNITS-REQ-ADJUSTMENT in PO-REC > 0
1.1.1.1.3.3.1 Then
1.1.1.1.3.3.1.1 Set UNITS-REQ-ADJUSTMENT in ON-HAND-REC = UNITS-REQ-ADJUSTMENT in ON-HAND-REC + UNITS-REQ-ADJUSTMENT in PO-REC
PROCESS NARRATIVE

Process Name: COMPLETE PO
Process Number: 2-3-2.PN

1.1.1.1.3.4 Write ON-HAND-REC in ON HAND RECORDS
    using ON-HAND-REC-TYPE
    and ON-HAND-REC-SUBCODE as key

1.1.1.2 If PO-REC-UNITS < PO-COMPLETED-UNITS
  1.1.1.2.1 Then
    1.1.1.2.1.1 Set PO-AVAILABLE-TO-PROMISE =
                  PO-AVAILABLE-TO-PROMISE + ( PO-REC-UNITS - PO-COMPLETED-UNITS )
    PO-REC-UNITS = PO-COMPLETED-UNITS
  1.1.1.2.1.2 Go Back to 1.1.1.1
               ( Test again Units in Order against Units completed )

1.1.1.3 If PO-REC-UNITS > PO-COMPLETED-UNITS
  1.1.1.3.1 Then
    1.1.1.3.1.1 Set DIFFERENCE = PO-REC-UNITS - PO-COMPLETED-UNITS
  1.1.1.3.1.2 If PO-AVAILABLE-TO-PROMISE > 0
                 1.1.1.3.1.2.1 Then
                 1.1.1.3.1.2.1.1 If PO-AVAILABLE-TO-PROMISE >= DIFFERENCE
                 1.1.1.3.1.2.1.1.1 Then
PROCESS NARRATIVE

Process Name: COMPLETE PO

Process Number: 2-3-2.PN

1.1.1.3.1.2.1.1.1.1 Set \( \text{PO-AVAILABLE-TO-PROMISE} = \text{PO-AVAILABLE-TO-PROMISE} - \text{DIFFERENCE} \)

1.1.1.3.1.2.1.1.2 Otherwise (Available to Promise < Difference)

1.1.1.3.1.2.1.1.2.1 Set \( \text{UNITS-REQ-ADJUSTMENT in PO-REC} = \text{UNITS-REQ-ADJUSTMENT} + \text{DIFFERENCE} - \text{PO-AVAILABLE-TO-PROMISE} \)

\( \text{PO-AVAILABLE-TO-PROMISE} = 0 \)

1.1.1.3.1.2.2 Otherwise (Available to Promise in PO = 0)

1.1.1.3.1.2.2.1 Set \( \text{UNITS-REQ-ADJUSTMENT in PO-REC} = \text{UNITS-REQ-ADJUSTMENT} + \text{DIFFERENCE} \)

1.1.1.3.1.3 Set \( \text{PO-COMPLETED} = \text{PO-REC-UNITS} \)

1.1.1.3.1.4 Go Back to 1.1.1.1 (Test again Units in Order against Units completed)

1.1.2 Delete \( \text{PO-REC in PO RECORDS using \text{PO-REC-NUMBER as key.}} \)
<table>
<thead>
<tr>
<th>PROCESS NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name: END WEEK</td>
</tr>
<tr>
<td>Process Number: 2-3-3.PN</td>
</tr>
</tbody>
</table>

1. Read PO-REC in PO RECORDS with PO-REC-STATUS = 'Scheduled'
   1.1 For each PO-REC
   1.1.1 Set PO-NUMBER in NOT-EXECUTED-REPORT = PO-REC-NUMBER
   1.1.2 Write PO-NOT-EXECUTED-REPORT

2. Ask User Option
   2.1 If Option is "Do not Unschedule some PO"
      2.1.1 Then Read all PO-TO-KEEP-SCHEDULED
      2.1.2 Read PO-REC in PO RECORDS with PO-REC-STATUS = 'Scheduled'
         2.1.2.1 For each PO-REC
         2.1.2.1.1 If PO-REC-NUMBER not equal to any PO-TO-KEEP-SCHEDULED
         2.1.2.1.1.1 Then
         2.1.2.1.1.1.1 Set PO-REC-STATUS = 'Planned'
         2.1.2.1.1.2 Write PO-REC in PO RECORDS using PO-REC-NUMBER as key
   2.2 If Option is "Unschedule all PO"
2.2.1 Read PO-REC in PO RECORDS with PO-REC-STATUS = "Scheduled"

2.2.1.1 For each PO-REC

2.2.1.1.1 Set PO-REC-STATUS = "Planned"

2.2.1.1.1 Write PO-REC in PO RECORDS using PO-REC-NUMBER as key.
PROCESS NARRATIVE

Process Name: DELETE PO

Process Number: 2-3-4.PN

Page: 01

1. Read DELETED-PO
   1.1 For each DELETED-PO entered
      1.1.1 Read PO-REC in PO RECORDS
           with PO-REC-NUMBER = DELETED-PO-NUMBER
      1.1.1.1 If PO-AVAILABLE-TO-PROMISE = PO-REC-UNITS
      1.1.1.1.1 Then delete PO-REC in PO RECORDS
                       using DELETED-PO-NUMBER as key.
PROCESS NARRATIVE

Process Name: REPORT ORDER

Process Number: 3-1.PN

Page: 01

1. Read ORDER-INFO

1.1 For each ORDER-INFO entered

1.1.1 Read ORDER-REC in ORDER RECORDS
    using ORDER-REC-NUMBER
    and ORDER-REC-ITEM as key

1.1.1.1 Set
REP-ORDER-REC-NUMBER = ORDER-REC-NUMBER
REP-ORDER-REC-ITEM = ORDER-REC-ITEM
REP-ORDER-REC-TYPE = ORDER-REC-TYPE
REP-ORDER-REC-SUBCODE = ORDER-REC-SUBCODE
REP-ORDER-REC-QUANT = ORDER-REC-QUANT
REP-PROMISED-BY-ON-HAND = PROMISED-BY-ON-HAND
REP-PROMISED-BY-MPS = PROMISED-BY-MPS
REP-PROMISED-BY-PO = PROMISED-BY-PO
REP-PROMISED-BY-PR = PROMISED-BY-PR
REP-UNITS-DELIVERED = UNITS-DELIVERED
REP-STILL-NOT-PROMISED = UNITS-STILL-NOT-PROMISED
REP-REQ-ADJUSTMENT = UNITS-REQ-ADJUSTMENT

1.1.1.2 Write REP-ORDER-REC

1.1.1.3 If PROMISED-BY-MPS > 0

1.1.1.3.1 Then Read MPS-ORDER-REC in MPS-ORDER
    using MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER
    and MPS-ORDER-REC-ITEM = ORDER-REC-ITEM as key

1.1.1.3.1.1 For each MPS-REC

1.1.1.3.1.1.1 Set
REP-MPS-NUMBER = MPS-ORDER-REC-MPS
REP-UNITS = MPS-REC-ORDER-REC-UNITS
PROCESS NARRATIVE

Process Name: REPORT ORDER

Process Number: 3-1.PN

Page: 02

1.1.1.3.1.1.2 Write REP-MPS-ORDER-REC

1.1.1.4 If PROMISED-BY-PO > 0

1.1.1.4.1 Then Read PO-ORDER-REC in PO-ORDER
   using PO-ORDER-REC-ORDER = ORDER-REC-NUMBER
   and PO-ORDER-REC-ITEM = ORDER-REC-ITEM as key

1.1.1.4.1.1 For each PO-REC

1.1.1.4.1.1.1 Set REP-PO-NUMBER = PO-ORDER-REC-PO
              REP-UNITS = PO-REC-ORDER-REC-UNITS

1.1.1.4.1.2 Write REP-PO-ORDER-REC

1.1.1.5 If PROMISED-BY-PR > 0

1.1.1.5.1 Then Read PR-ORDER-REC in PR-ORDER
   using PR-ORDER-REC-ORDER = ORDER-REC-NUMBER
   and PR-ORDER-REC-ITEM = ORDER-REC-ITEM as key

1.1.1.5.1.1 For each PR-REC

1.1.1.5.1.1.1 Set REP-PR-NUMBER = PR-ORDER-REC-PR
              REP-UNITS = PR-REC-ORDER-REC-UNITS

1.1.1.5.1.2 Write REP-PR-ORDER-REC.
<table>
<thead>
<tr>
<th>1. Select User Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 If Option is &quot;Report On Hand Requiring Adjustment&quot;</td>
</tr>
<tr>
<td>1.1.1 Then</td>
</tr>
<tr>
<td>1.1.1.1.1 Read ON-HAND-REC in ON HAND RECORDS with UNITS-REQ-ADJUSTMENT &gt; 0</td>
</tr>
<tr>
<td>1.1.1.1.1.1 For each ON-HAND-REC</td>
</tr>
<tr>
<td>1.1.1.1.1.1.1 Set ADJ-REP-TYPE = ON-HAND-REC-TYPE</td>
</tr>
<tr>
<td>ADJ-REP-SUBCODE = ON-HAND-REC-SUBCODE</td>
</tr>
<tr>
<td>ADJ-REP-UNITS = ON-HAND-REC-UNITS</td>
</tr>
<tr>
<td>ADJ-REP-ADJ-UNITS = UNITS-REQ-ADJUSTMENT</td>
</tr>
<tr>
<td>1.1.1.1.1.2 Write ON-HAND-REQ-ADJUSTMENT-REPORT</td>
</tr>
<tr>
<td>1.2 If Option is &quot;Report On Hand Available to Promise&quot;</td>
</tr>
<tr>
<td>1.1.1 Then</td>
</tr>
<tr>
<td>1.1.1.1 Read ON-HAND-REC in ON HAND RECORDS with ON-HAND-AVAILABLE-TO-PROMISE &gt; 0</td>
</tr>
<tr>
<td>1.1.1.1.1 For each ON-HAND-REC</td>
</tr>
<tr>
<td>1.1.1.1.1.1 Set AVAIL-REP-TYPE = ON-HAND-REC-TYPE</td>
</tr>
<tr>
<td>AVAIL-REP-SUBCODE = ON-HAND-REC-SUBCODE</td>
</tr>
<tr>
<td>AVAIL-REP-UNITS = ON-HAND-REC-UNITS</td>
</tr>
<tr>
<td>AVAIL-REP-AVAIL-UNITS = ON-HAND-AVAILABLE-TO-PROMISE</td>
</tr>
<tr>
<td>1.1.1.1.2 Write ON-HAND-AVAILABLE-REPORT</td>
</tr>
</tbody>
</table>
1.3 If Option is "Enter Order New Promised By On Hand"

1.3.1 Then

1.3.1.1.1.1 For each ORDER-ON-HAND-INFO entered

1.3.1.1.1.1.1 Read ORDER-REC
using ORDER-REC-NUMBER
and ORDER-REC-ITEM as key

1.3.1.1.1.1.1 Read ON-HAND-ORDER-REC in ON-HAND-ORDER
using ON-HAND-ORDER-REC-ORDER = ORDER-REC-NUMBER
ON-HAND-ORDER-REC-ITEM = ORDER-REC-ITEM

1.3.1.1.1.1.1.1 If can't find ON-HAND-ORDER-REC

1.3.1.1.1.1.1.1.1 Then

1.3.1.1.1.1.1.1.1.1 Set
ON-HAND-ORDER-REC-ORDER = ORDER-REC-NUMBER
ON-HAND-ORDER-REC-ITEM = ORDER-REC-ITEM
ON-HAND-ORDER-REC-UNITS = 0

1.3.1.1.1.2 If NEW-PROMISED-BY-ON-HAND is less than
PROMISED-BY-ON-HAND in ORDER-REC

1.3.1.1.2.1 Then

1.3.1.1.2.1.1 Set DIFFERENCE =
( PROMISED-BY-ON-HAND in ORDER-REC )
- ( NEW-PROMISED-BY-ON-HAND )
1.3.1.1.1.2.1.2 Read ON-HAND-REC in ON HAND RECORDS
  using ON-HAND-TYPE = ORDER-REC-TYPE
  ON-HAND-SUBCODE = ORDER-REC-SUBCODE

1.3.1.1.1.2.1.3 If UNITS-REQ-ADJUSTMENT in ON-HAND-REC > 0

1.3.1.1.1.2.1.3.1 Then

1.3.1.1.1.2.1.3.1.1 If UNITS-REQ-ADJUSTMENT in ON-HAND-REC
  is less than DIFFERENCE

1.3.1.1.1.2.1.3.1.1.1 Then

1.3.1.1.1.2.1.3.1.1.1.1 Set

  DIFFERENCE = DIFFERENCE
  - UNITS-REQ-ADJUSTMENT in ON-HAND-REC

  UNITS-REQ-ADJUSTMENT in ON-HAND-REC = 0

1.3.1.1.1.2.1.3.1.1.2 Otherwise
  ( Units Req Adjust is Greater )

1.3.1.1.1.2.1.3.1.1.2.1 Set

  UNITS-REQ-ADJUSTMENT in ON-HAND-REC =
  UNITS-REQ-ADJUSTMENT - DIFFERENCE

  DIFFERENCE = 0

1.3.1.1.1.2.1.4 If DIFFERENCE > 0

1.3.1.1.1.2.1.4.1 Then
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY ON HAND

Process Number: 3-2-1.PN

Page: 04

1.3.1.1.1.2.1.4.1.1 Set ON-HAND-AVAILABLE-TO-PROMISE =
    ON-HAND-AVAILABLE-TO-PROMISE
    + DIFFERENCE

1.3.1.1.1.2.1.5 If NEW-PROMISED-BY-ON-HAND = 0

1.3.1.1.1.2.1.5.1 Then

1.3.1.1.1.2.1.5.1.1 Delete ON-HAND-ORDER-REC
    in ON HAND-ORDER
    using ON-HAND-ORDER-REC-ORDER
    and ON-HAND-ORDER-REC-ITEM as key

1.3.1.1.1.2.1.5.2 Otherwise

1.3.1.1.1.2.1.5.2.1 Set ON-HAND-ORDER-REC-UNITS =
    NEW-PROMISED-BY-ON-HAND

1.3.1.1.1.2.1.5.2.2 Write ON-HAND-ORDER-REC
    in ON HAND-ORDER
    using ON-HAND-ORDER-REC-ORDER
    and ON-HAND-ORDER-REC-ITEM as key

1.3.1.1.2.1.6 Write ON-HAND-REC in ON HAND RECORDS
    using ON-HAND-REC-TYPE
    and ON-HAND-REC-SUBCODE as key

1.3.1.1.1.3 If NEW-PROMISED-BY-ON-HAND is greater than
    PROMISED-BY-ON-HAND in ORDER-REC

1.3.1.1.1.3.1 Then
1.3.1.1.1.3.1.1 Set DIFFERENCE =
   ( NEW-PROMISED-BY-ON-HAND )
   - ( PROMISED-BY-ON-HAND in ORDER-REC )

1.3.1.1.3.1.2 Read ON-HAND-REC in ON HAND RECORDS
   using ON-HAND-TYPE = ORDER-REC-TYPE
   ON-HAND-SUBCODE = ORDER-REC-SUBCODE

1.3.1.1.3.1.3 If ON-HAND-AVAILABLE-TO-PROMISE > 0
   1.3.1.1.3.1.3.1 Then
   1.3.1.1.3.1.3.1.1 If ON-HAND-AVAILABLE-TO-PROMISE
      is greater than DIFFERENCE
      1.3.1.1.3.1.3.1.1.1 Then
      1.3.1.1.3.1.3.1.1.1.1 Set ON-HAND-AVAILABLE-TO-PROMISE =
         ON-HAND-AVAILABLE-TO-PROMISE
         - DIFFERENCE
         DIFFERENCE = 0
      1.3.1.1.3.1.3.1.1.2 Otherwise
         ( Available to Prom is Smaller )
      1.3.1.1.3.1.3.1.1.2.1 Set
         DIFFERENCE = DIFFERENCE - ON-HAND-AVAILABLE-TO-PROMISE
         ON-HAND-AVAILABLE-TO-PROMISE = 0
   1.3.1.1.3.1.4 If DIFFERENCE > 0
### PROCESS NARRATIVE

**Process Name:** CHANGE PROMISED BY ON HAND  
**Process Number:** 3-2-1.PN  
**Page:** 06

| 1.3.1.1.1.3.1.4.1 | Set UNITS-REQ-ADJUSTMENT in ON-HAND-REC  
|                  | = UNITS-REQ-ADJUSTMENT + DIFFERENCE |
| 1.3.1.1.1.3.1.5  | Write ON-HAND-ORDER-REC in ON HAND-ORDER  
|                  | using ON-HAND-ORDER-REC-ORDER  
|                  | and ON-HAND-ORDER-REC-ITEM as key |
| 1.3.1.1.1.3.1.6  | Write ON-HAND-REC in ON HAND RECORDS  
|                  | using ON-HAND-REC-TYPE  
|                  | and ON-HAND-REC-SUBCODE as key |
| 1.3.1.1.1.4      | If NEW-PROMISED-BY-ON-HAND is less than  
|                  | PROMISED-BY-ON-HAND in ORDER-REC |
| 1.3.1.1.1.4.1    | Then  
| 1.3.1.1.1.4.1.1 | Set DIFFERENCE =  
|                  | ( PROMISED-BY-ON-HAND in ORDER-REC )  
|                  | - ( NEW-PROMISED-BY-ON-HAND ) |
|                  | PROMISED-BY-ON-HAND in ORDER-REC =  
|                  | PROMISED-BY-ON-HAND - DIFFERENCE |
| 1.3.1.1.1.4.1.2  | If UNITS-REQ-ADJUSTMENT in ORDER-REC > 0  
| 1.3.1.1.1.4.1.2.1 | Then  
| 1.3.1.1.1.4.1.2.1.1 | If UNITS-REQ-ADJUSTMENT <= DIFFERENCE  
| 1.3.1.1.1.4.1.2.1.1.1 | Then  
| 1.3.1.1.1.4.1.2.1.1.1.1 | Set DIFFERENCE =  
|                  | DIFFERENCE - UNITS-REQ-ADJUSTMENT |
**PROCESS NARRATIVE**

**Process Name:** CHANGE PROMISED BY ON HAND

**Process Number:** 3-2-1.PN  
**Page:** 07

\[
\text{UNITS-REQ-ADJUSTMENT} = 0
\]

1.3.1.1.4.1.2.1.1.2 Otherwise  
( Units Req Adjust is Greater )

1.3.1.1.4.1.2.1.1.2.1 Then

1.3.1.1.4.1.2.1.1.2.1.1 Set \[ \text{UNITS-REQ-ADJUSTMENT} = \text{UNITS-REQ-ADJUSTMENT} - \text{DIFFERENCE} \]

\[ \text{DIFFERENCE} = 0 \]

1.3.1.1.4.1.3 If \[ \text{DIFFERENCE} > 0 \]

1.3.1.1.4.1.3.1 Then

1.3.1.1.4.1.3.1.1 Set \[ \text{UNITS-STILL-NOT-PROMISED in ORDER-REC} = \text{UNITS-STILL-NOT-PROMISED} + \text{DIFFERENCE} \]

1.3.1.1.4.2 Otherwise  
( New Promised By Hand is Greater )

1.3.1.1.4.2.1 Set \[ \text{DIFFERENCE} = \]

( NEW-PROMISED-BY-ON-HAND )
- ( PROMISED-BY-ON-HAND in ORDER-REC )

PROMISED-BY-ON-HAND in ORDER-REC =
PROMISED-BY-ON-HAND + DIFFERENCE

1.3.1.1.4.2.2 If  
UNITS-STILL-NOT-PROMISED in ORDER-REC > 0

1.3.1.1.4.2.2.1 Then
1.3.1.1.1.4.2.2.1.1 If UNITS-STILL-NOT-PROMISED is greater than DIFFERENCE

1.3.1.1.1.4.2.2.1.1.1 Then

1.3.1.1.1.4.2.2.1.1.1.1 Set UNITS-STILL-NOT-PROMISED = UNITS-STILL-NOT-PROMISED - DIFFERENCE

DIFFERENCE = 0

1.3.1.1.1.4.2.2.1.1.2 Otherwise (Difference is Greater)

1.3.1.1.1.4.2.2.1.1.2.1 Set DIFFERENCE = DIFFERENCE - UNITS-STILL-NOT-PROMISED

UNITS-STILL-NOT-PROMISED = 0

1.3.1.1.1.4.2.3 If DIFFERENCE > 0

1.3.1.1.1.4.2.3.1 Then

1.3.1.1.1.4.2.3.1.1 Set UNITS-REQ-ADJUSTMENT in ORDER-REC = UNITS-REQ-ADJUSTMENT + DIFFERENCE

1.3.1.1.1.5 Write ORDER-REC using ORDER-REC-NUMBER and ORDER-REC-ITEM as key.
<table>
<thead>
<tr>
<th>PROCESS NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name: CHANGE PROMISED BY MPS</td>
</tr>
<tr>
<td>Process Number: 3-2-2.PN</td>
</tr>
</tbody>
</table>

1. Select User Option

1.1 If Option is "Report MPS Available to Promise"

1.1.1 Then

1.1.1.1 Read MPS-REC in MPS RECORDS with MPS-AVAILABLE-TO-PROMISE > 0

1.1.1.1.1 For each MPS-REC

1.1.1.1.1.1 Set AVAIL-REP-TYPE = MPS-REC-TYPE
AVAIL-REP-SUBCODE = MPS-REC-SUBCODE
AVAIL-REP-UNITS = MPS-REC-UNITS
AVAIL-REP-AVAIL-UNITS = MPS-AVAILABLE-TO-PROMISE

1.1.1.1.2 Write MPS-AVAILABLE-REPORT

1.2 If Option is "Enter Order New Promised By MPS"

1.2.1 Then

1.2.1.1 Read ORDER-MPS-INFO

1.2.1.1.1 For each ORDER-MPS-INFO entered

1.2.1.1.1.1 Read ORDER-REC using ORDER-REC-NUMBER
ORDER-REC-ITEM
and MPS-REC-NUMBER as key
## PROCESS NARRATIVE

**Process Name:** CHANGE PROMISED BY MPS  
**Process Number:** 3-2-2.PN

| 1.2.1.1.1.1.1.1 | Read MPS-ORDER-REC in MPS-ORDER  
using MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER  
MPS-ORDER-REC-ITEM = ORDER-REC-ITEM  
and MPS-ORDER-REC-MPS = MPS-REC-NUMBER |
|-----------------|-----------------------------------|
| 1.2.1.1.1.1.1.1.1.1 | If can’t find MPS-ORDER-REC  
1.2.1.1.1.1.1.1.1.1.1 | Then  
1.2.1.1.1.1.1.1.1.1.1.1 | Set  
MPS-ORDER-REC-ORDER = ORDER-REC-NUMBER  
MPS-ORDER-REC-ITEM = ORDER-REC-ITEM  
MPS-ORDER-REC-MPS = MPS-REC-NUMBER  
MPS-ORDER-REC-UNITS = 0 |
|-----------------|-----------------------------------|
| 1.2.1.1.1.1.2 | If NEW-PROMISED-BY-MPS is less than  
PROMISED-BY-MPS in ORDER-REC  
1.2.1.1.1.1.2.1 | Then  
1.2.1.1.1.1.2.1.1 | Set DIFFERENCE =  
( PROMISED-BY-MPS in ORDER-REC )  
- ( NEW-PROMISED-BY-MPS ) |
|-----------------|-----------------------------------|
| 1.2.1.1.1.1.2.1.2 | Read MPS-REC in MPS RECORDS  
using MPS-REC-NUMBER as key |
|-----------------|-----------------------------------|
| 1.2.1.1.1.1.2.1.3 | Set MPS-AVAILABLE-TO-PROMISE =  
MPS-AVAILABLE-TO-PROMISE + DIFFERENCE |
|-----------------|-----------------------------------|
| 1.2.1.1.1.1.2.1.4 | If NEW-PROMISED-BY-MPS = 0  
1.2.1.1.1.1.2.1.4.1 | Then |
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY MPS

Process Number: 3-2-2.PN

1.2.1.1.1.2.1.4.1.1 Delete MPS-ORDER-REC in MPS-ORDER
    using MPS-ORDER-REC-ORDER
    MPS-ORDER-REC-ITEM
    and MPS-ORDER-REC-MPS as key

1.2.1.1.1.2.1.4.2 Otherwise

1.2.1.1.1.2.1.4.2.1 Set MPS-ORDER-REC-UNITS = NEW-PROMISED-BY-MPS

1.2.1.1.1.2.1.4.2.2 Write MPS-ORDER-REC in MPS-ORDER
    using MPS-ORDER-REC-ORDER
    MPS-ORDER-REC-ITEM
    and MPS-ORDER-REC-MPS as key

1.2.1.1.1.2.1.5 Write MPS-REC in MPS RECORDS
    using MPS-REC-TYPE
    and MPS-REC-SUBCODE as key

1.2.1.1.1.3 If NEW-PROMISED-BY-MPS is greater than
    PROMISED-BY-MPS in ORDER-REC

1.2.1.1.1.3.1 Then

1.2.1.1.1.3.1.1 Set DIFFERENCE =
    ( NEW-PROMISED-BY-MPS )
    - ( PROMISED-BY-MPS in ORDER-REC )

1.2.1.1.1.3.1.2 Read MPS-REC in MPS RECORDS
    using MPS-REC-NUMBER as key

1.2.1.1.1.3.1.3 If MPS-AVAILABLE-TO-PROMISE > 0

1.2.1.1.1.3.1.3.1 Then
1.2.1.1.1.3.1.3.1.1.1 If MPS-AVAILABLE-TO-PROMISE is greater than DIFFERENCE

1.2.1.1.1.3.1.3.1.1.1 Then

1.2.1.1.1.3.1.3.1.1.1.1 Set MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE - DIFFERENCE

   DIFFERENCE = 0

1.2.1.1.1.3.1.3.1.1.2 Otherwise (Available to Prom is Smaller)

   1.2.1.1.1.3.1.3.1.1.2.1 Set
   
   DIFFERENCE = DIFFERENCE - MPS-AVAILABLE-TO-PROMISE

   MPS-AVAILABLE-TO-PROMISE = 0

1.2.1.1.1.3.1.4 If DIFFERENCE > 0

   1.2.1.1.1.3.1.4.1 Set MPS-REC-UNITS =
   
   = MPS-REC-UNITS + DIFFERENCE

   1.2.1.1.1.3.1.5 Write MPS-ORDER-REC in MPS ORDER

   using MPS-ORDER-REC-ORDER

   MPS-ORDER-REC-ITEM

   and MPS-ORDER-REC-MPS as key

   1.2.1.1.1.3.1.6 Write MPS-REC in MPS RECORDS

   using MPS-REC-NUMBER as key

1.2.1.1.1.4 If NEW-PROMISED-BY-MPS is less than PROMISED-BY-MPS in ORDER-REC
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY MPS

Process Number: 3-2-2.PN

Page: 05

1.2.1.1.4.1 Then

1.2.1.1.4.1.1 Set DIFFERENCE =

(PROMISED-BY-MPS in ORDER-REC )

- ( NEW-PROMISED-BY-MPS )

PROMISED-BY-MPS in ORDER-REC =

PROMISED-BY-MPS - DIFFERENCE

1.2.1.1.4.1.2 If UNITS-REQ-ADJUSTMENT in ORDER-REC > 0

1.2.1.1.4.1.2.1 Then

1.2.1.1.4.1.2.1.1 If UNITS-REQ-ADJUSTMENT < = DIFFERENCE

1.2.1.1.4.1.2.1.1.1 Then

1.2.1.1.4.1.2.1.1.1.1 Set DIFFERENCE =

DIFFERENCE - UNITS-REQ-ADJUSTMENT

UNITS-REQ-ADJUSTMENT = 0

1.2.1.1.4.1.2.1.1.2 Otherwise

( Units Req Adjust is Greater )

1.2.1.1.4.1.2.1.2.1 Then

1.2.1.1.4.1.2.1.2.1.1 Set UNITS-REQ-ADJUSTMENT =

UNITS-REQ-ADJUSTMENT - DIFFERENCE

DIFFERENCE = 0

1.2.1.1.4.1.3 If DIFFERENCE > 0
1.2.1.1.4.1.3.1 Then

1.2.1.1.4.1.3.1.1 Set UNITS-STILL-NOT-PROMISED in ORDER-REC =
UNITS-STILL-NOT-PROMISED + DIFFERENCE

1.2.1.1.4.2 Otherwise

( New Promised By Hand is Greater )

1.2.1.1.4.2.1 Set DIFFERENCE =
( NEW-PROMISED-BY-MPS )
- ( PROMISED-BY-MPS in ORDER-REC )
PROMISED-BY-MPS in ORDER-REC =
PROMISED-BY-MPS + DIFFERENCE

1.2.1.1.4.2.2 If
UNITS-STILL-NOT-PROMISED in ORDER-REC > 0

1.2.1.1.4.2.2.1 Then

1.2.1.1.4.2.2.1.1 If UNITS-STILL-NOT-PROMISED is greater than DIFFERENCE

1.2.1.1.4.2.2.1.1.1 Then

1.2.1.1.4.2.2.1.1.1.1 Set UNITS-STILL-NOT-PROMISED =
UNITS-STILL-NOT-PROMISED - DIFFERENCE

DIFFERENCE = 0

1.2.1.1.4.2.2.1.1.2 Otherwise

( Difference is Greater )
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY MPS

Process Number: 3-2-2.PN

1.2.1.1.4.2.2.1.1.2.1 Set DIFFERENCE = DIFFERENCE
- UNITS-STILL-NOT-PROMISED

UNITS-STILL-NOT-PROMISED = 0

1.2.1.1.4.2.3 If DIFFERENCE > 0

1.2.1.1.4.2.3.1 Then

1.2.1.1.4.2.3.1.1 Set UNITS-REQ-ADJUSTMENT in ORDER-REC
= UNITS-REQ-ADJUSTMENT + DIFFERENCE

1.2.1.1.5 Write ORDER-REC
using ORDER-REC-NUMBER
and ORDER-REC-ITEM as key.
# PROCESS NARRATIVE

**Process Name:** MANAGE PO-REC  
**Process Number:** 3-2-3.PN  
**Page:** 01

<table>
<thead>
<tr>
<th>1. Select User Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 If Option is &quot;Report PO Requiring Adjustment&quot;</td>
</tr>
<tr>
<td>1.1.1 Then</td>
</tr>
</tbody>
</table>
| 1.1.1.1 Read PO-REC in PO RECORDS  
with UNITS-REQ-ADJUSTMENT > 0 |
| 1.1.1.1.1 For each PO-REC |
| 1.1.1.1.1.1 Set ADJ-REP-TYPE = PO-REC-TYPE  
ADJ-REP-SUBCODE = PO-REC-SUBCODE  
ADJ-REP-UNITS = PO-REC-UNITS  
ADJ-REP-ADJ-UNITS = UNITS-REQ-ADJUSTMENT |
| 1.1.1.1.1.2 Write PO-REQ-ADJUSTMENT-REPORT |
| 1.2 If Option is "Report PO Available to Promise" |
| 1.1.1 Then |
| 1.1.1.1 Read PO-REC in PO RECORDS  
with PO-AVAILABLE-TO-PROMISE > 0 |
| 1.1.1.1.1 For each PO-REC |
| 1.1.1.1.1.1 Set AVAIL-REP-TYPE = PO-REC-TYPE  
AVAIL-REP-SUBCODE = PO-REC-SUBCODE  
AVAIL-REP-UNITS = PO-REC-UNITS  
AVAIL-REP-AVAIL-UNITS = PO-AVAILABLE-TO-PROMISE |
| 1.1.1.1.2 Write PO-AVAILABLE-REPORT |
1.3 If Option is "Enter Order New Promised By PO"

1.3.1 Then

1.3.1.1. Read ORDER-PO-INFO

1.3.1.1.1 For each ORDER-PO-INFO entered

1.3.1.1.1.1 Read ORDER-REC
    using ORDER-REC-NUMBER
    and ORDER-REC-ITEM as key

1.3.1.1.1.1 Read PO-ORDER-REC in PO-ORDER
    using PO-ORDER-REC-ORDER = ORDER-REC-NUMBER
    PO-ORDER-REC-ITEM = ORDER-REC-ITEM
    and PO-ORDER-REC-PO = PO-REC-NUMBER as index

1.3.1.1.1.1.1 If can’t find PO-ORDER-REC

1.3.1.1.1.1.1.1 Then

1.3.1.1.1.1.1.1.1 Set
    PO-ORDER-REC-ORDER = ORDER-REC-NUMBER
    PO-ORDER-REC-ITEM = ORDER-REC-ITEM
    PO-ORDER-REC-PO = PO-REC-NUMBER
    PO-ORDER-REC-UNITS = 0

1.3.1.1.2 If NEW-PROMISED-BY-PO is less than
PROMISED-BY-PO in ORDER-REC

1.3.1.1.2.1 Then
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY PO

Process Number: 3-2-3.PN

Page: 03

1.3.1.1.1.2.1.1 Set DIFFERENCE =
   ( PROMISED-BY-PO in ORDER-REC )
   - ( NEW-PROMISED-BY-PO )

1.3.1.1.1.2.1.2 Read PO-REC in PO RECORDS
   using PO-REC-NUMBER as index

1.3.1.1.1.2.1.3 If UNITS-REQ-ADJUSTMENT in PO-REC > 0
1.3.1.1.1.2.1.3.1 Then
1.3.1.1.1.2.1.3.1.1 If UNITS-REQ-ADJUSTMENT in PO-REC
   is less than DIFFERENCE
1.3.1.1.1.2.1.3.1.1.1 Then
1.3.1.1.1.2.1.3.1.1.1.1 Set
   DIFFERENCE = DIFFERENCE
   - UNITS-REQ-ADJUSTMENT in PO-REC

   UNITS-REQ-ADJUSTMENT in PO-REC = 0

1.3.1.1.1.2.1.3.1.1.2 Otherwise
   ( Units Req Adjust is Greater )

1.3.1.1.1.2.1.3.1.2.1 Set
   UNITS-REQ-ADJUSTMENT in PO-REC =
   UNITS-REQ-ADJUSTMENT - DIFFERENCE

   DIFFERENCE = 0
<table>
<thead>
<tr>
<th>1.3.1.1.1.2.1.4 If DIFFERENCE &gt; 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.1.1.1.2.1.4.1 Then</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.4.1.1 Set PO-AVAILABLE-TO-PROMISE =</td>
</tr>
<tr>
<td>PO-AVAILABLE-TO-PROMISE + DIFFERENCE</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.5 If NEW-PROMISED-BY-PO = 0</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.5.1 Then</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.5.1.1 Delete PO-ORDER-REC in PO-ORDER</td>
</tr>
<tr>
<td>using PO-ORDER-REC-ORDER</td>
</tr>
<tr>
<td>and PO-ORDER-REC-PO as key</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.5.2 Otherwise</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.5.2.1 Set PO-ORDER-REC-UNITS =</td>
</tr>
<tr>
<td>NEW-PROMISED-BY-PO</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.5.2.2 Write PO-ORDER-REC in PO-ORDER</td>
</tr>
<tr>
<td>using PO-ORDER-REC-ORDER</td>
</tr>
<tr>
<td>and PO-ORDER-REC-PO as key</td>
</tr>
<tr>
<td>1.3.1.1.2.1.6 Write PO-REC in PO RECORDS</td>
</tr>
<tr>
<td>using PO-REC-NUMBER as key</td>
</tr>
<tr>
<td>1.3.1.1.1.3 If NEW-PROMISED-BY-PO is greater than</td>
</tr>
<tr>
<td>PROMISED-BY-PO in ORDER-REC</td>
</tr>
<tr>
<td>1.3.1.1.1.3.1 Then</td>
</tr>
</tbody>
</table>
1.3.1.1.1.3.1.1 Set DIFFERENCE =
   ( NEW-PROMISED-BY-PO )
   - ( PROMISED-BY-PO in ORDER-REC )

1.3.1.1.1.3.1.2 Read PO-REC in PO RECORDS
   using PO-REC-NUMBER as key

1.3.1.1.1.3.1.3 If PO-AVAILABLE-TO-PROMISE > 0

1.3.1.1.1.3.1.1 Then

1.3.1.1.1.3.1.1.1 If PO-AVAILABLE-TO-PROMISE
   is greater than DIFFERENCE

1.3.1.1.1.3.1.1.1.1 Then

1.3.1.1.1.3.1.1.1.1.1 Set PO-AVAILABLE-TO-PROMISE =
   PO-AVAILABLE-TO-PROMISE
   - DIFFERENCE
   DIFFERENCE = 0

1.3.1.1.1.3.1.1.1.2 Otherwise
   ( Available to Prom is Smaller )

1.3.1.1.1.3.1.1.2.1 Set

   DIFFERENCE = DIFFERENCE - PO-AVAILABLE-TO-PROMISE
   PO-AVAILABLE-TO-PROMISE = 0

1.3.1.1.1.3.1.4 If DIFFERENCE > 0
<table>
<thead>
<tr>
<th>Process Name: CHANGE PROMISED BY PO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Number: 3-2-3.PN</td>
</tr>
</tbody>
</table>

1.3.1.1.1.3.1.4.1 Set UNITS-REQ-ADJUSTMENT in PO-REC = UNITS-REQ-ADJUSTMENT + DIFFERENCE

1.3.1.1.1.3.1.5 Write PO-ORDER-REC in PO-ORDER using PO-ORDER-REC-ORDER, PO-ORDER-REC-ITEM, and PO-ORDER-REC-PO as key

1.3.1.1.1.3.1.6 Write PO-REC in PO RECORDS using PO-REC-NUMBER as key

1.3.1.1.1.4 If NEW-PROMISED-BY-PO is less than PROMISED-BY-PO in ORDER-REC

1.3.1.1.1.4.1 Then

1.3.1.1.1.4.1.1 Set DIFFERENCE = (PROMISED-BY-PO in ORDER-REC) - (NEW-PROMISED-BY-PO)

   PROMISED-BY-PO in ORDER-REC = PROMISED-BY-PO - DIFFERENCE

1.3.1.1.1.4.1.2 If UNITS-REQ-ADJUSTMENT in ORDER-REC > 0

1.3.1.1.1.4.1.2.1 Then

1.3.1.1.1.4.1.2.1.1 If UNITS-REQ-ADJUSTMENT = DIFFERENCE

1.3.1.1.1.4.1.2.1.1.1 Then

1.3.1.1.4.1.2.1.1.1.1 Set DIFFERENCE = DIFFERENCE - UNITS-REQ-ADJUSTMENT
### PROCESS NARRATIVE

**Process Name:** CHANGE PROMISED BY PO  
**Process Number:** 3-2-3.PN  
**Page:** 07

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UNITS-REQ-ADJUSTMENT</strong> = 0</td>
<td></td>
</tr>
<tr>
<td>1.3.1.1.1.4.1.2.1.1.2 Otherwise</td>
<td>( Units Req Adjust is Greater )</td>
</tr>
<tr>
<td>1.3.1.1.1.4.1.2.1.1.2.1 Then</td>
<td></td>
</tr>
<tr>
<td>1.3.1.1.1.4.1.2.1.1.2.1.1 Set <strong>UNITS-REQ-ADJUSTMENT</strong> = <strong>UNITS-REQ-ADJUSTMENT</strong> - <strong>DIFFERENCE</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DIFFERENCE</strong> = 0</td>
<td></td>
</tr>
<tr>
<td>1.3.1.1.1.4.1.3 If <strong>DIFFERENCE</strong> &gt; 0</td>
<td></td>
</tr>
<tr>
<td>1.3.1.1.1.4.1.3.1 Then</td>
<td></td>
</tr>
<tr>
<td>1.3.1.1.1.4.1.3.1.1 Set <strong>UNITS-STILL-NOT-PROMISED</strong> in <strong>ORDER-REC</strong> = <strong>UNITS-STILL-NOT-PROMISED</strong> + <strong>DIFFERENCE</strong></td>
<td></td>
</tr>
<tr>
<td>1.3.1.1.1.4.2 Otherwise</td>
<td>( New Promised By Hand is Greater )</td>
</tr>
<tr>
<td>1.3.1.1.1.4.2.1 Set <strong>DIFFERENCE</strong> =</td>
<td></td>
</tr>
<tr>
<td>( <strong>NEW-PROMISED-BY-PO</strong> )</td>
<td></td>
</tr>
<tr>
<td>- ( <strong>PROMISED-BY-PO</strong> in <strong>ORDER-REC</strong> )</td>
<td></td>
</tr>
<tr>
<td><strong>PROMISED-BY-PO</strong> in <strong>ORDER-REC</strong> = <strong>PROMISED-BY-PO</strong> + <strong>DIFFERENCE</strong></td>
<td></td>
</tr>
<tr>
<td>1.3.1.1.1.4.2.2 If</td>
<td><strong>UNITS-STILL-NOT-PROMISED</strong> in <strong>ORDER-REC</strong> &gt; 0</td>
</tr>
<tr>
<td>1.3.1.1.1.4.2.2.1 Then</td>
<td></td>
</tr>
</tbody>
</table>
1.3.1.1.4.2.2.1.1 If UNITS-STILL-NOT-PROMISED is greater than DIFFERENCE
1.3.1.1.4.2.2.1.1.1 Then
1.3.1.1.4.2.2.1.1.1.1 Set UNITS-STILL-NOT-PROMISED = UNITS-STILL-NOT-PROMISED - DIFFERENCE
DIFFERENCE = 0
1.3.1.1.4.2.2.1.1.2 Otherwise (Difference is Greater)
1.3.1.1.4.2.2.1.1.2.1 Set DIFFERENCE = DIFFERENCE - UNITS-STILL-NOT-PROMISED
UNITS-STILL-NOT-PROMISED = 0
1.3.1.1.4.2.3 If DIFFERENCE > 0
1.3.1.1.4.2.3.1 Then
1.3.1.1.4.2.3.1.1 Set UNITS-REQ-ADJUSTMENT in ORDER-REC = UNITS-REQ-ADJUSTMENT + DIFFERENCE
1.3.1.1.5 Write ORDER-REC using ORDER-REC-NUMBER and ORDER-REC-ITEM as key.
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY PR
Process Number: 3-2-4.PN

1. Select User Option

1.1 If Option is "Report PR Still Not Purchased"

1.1.1 Then

1.1.1.1 Read PR-REC in PR RECORDS
  with UNITS-STILL-NOT-PURCHASED > 0

1.1.1.1.1 For each PR-REC

1.1.1.1.1.1 Set NOT-PURCH-REP-TYPE = PR-REC-TYPE
  NOT-PURCH-REP-SUBCODE = PR-REC-SUBCODE
  NOT-PURCH-REP-PR-UNITS = PR-REC-UNITS
  NOT-PURCH-REP-UNITS = UNITS-STILL-NOT-PURCHASED

1.1.1.1.1.2 Write PR-STILL-NOT-PURCHASED-REPORT

1.2 If Option is "Report PR Available to Promise"

1.2.1 Then

1.2.1.1 Read PR-REC in PR RECORDS
  with PR-AVAILABLE-TO-PROMISE > 0

1.2.1.1 For each PR-REC

1.2.1.1.1 Set AVAIL-REP-TYPE = PR-REC-TYPE
  AVAIL-REP-SUBCODE = PR-REC-SUBCODE
  AVAIL-REP-UNITS = PR-REC-UNITS
  AVAIL-REP-AVAIL-UNITS = PR-AVAILABLE-TO-PROMISE

1.2.1.1.2 Write PR-AVAILABLE-REPORT
1.3 If Option is "Enter Order New Promised By PR"

1.3.1 Then

1.3.1.1.1. Read ORDER-PR-INFO

1.3.1.1.1.1 For each ORDER-PR-INFO entered

1.3.1.1.1.1.1 Read ORDER-REC
    using ORDER-REC-NUMBER
    and ORDER-REC-ITEM as key

1.3.1.1.1.1.1. Read PR-ORDER-REC in PR-ORDER
    using PR-ORDER-REC-ORDER = ORDER-REC-NUMBER
    PR-ORDER-REC-ITEM = ORDER-REC-ITEM
    and PR-ORDER-REC-PR = PR-REC-NUMBER as index

1.3.1.1.1.1.1.1 If can’t find PR-ORDER-REC

1.3.1.1.1.1.1.1.1 Set
    PR-ORDER-REC-ORDER = ORDER-REC-NUMBER
    PR-ORDER-REC-ITEM = ORDER-REC-ITEM
    PR-ORDER-REC-PR = PR-REC-NUMBER
    PR-ORDER-REC-UNITS = 0

1.3.1.1.1.2 If NEW-PROMISED-BY-PR is less than
PROMISED-BY-PR in ORDER-REC

1.3.1.1.2.1 Then
**PROCESS NARRATIVE**

Process Name: CHANGE PROMISED BY PR

Process Number: 3-2-4.PN

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.1.1.1.2.1.1</td>
<td>Set DIFFERENCE = (PROMISED-BY-PR in ORDER-REC) - (NEW-PROMISED-BY-PR)</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.2</td>
<td>Read PR-REC in PR RECORDS using PR-REC-NUMBER as index</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.3</td>
<td>If UNITS-STILL-NOT-PURCHASED &gt; 0</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.3.1</td>
<td>Then</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.3.1.1</td>
<td>If UNITS-STILL-NOT-PURCHASED is less than DIFFERENCE</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.3.1.1.1</td>
<td>Then</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.3.1.1.1.1</td>
<td>Set DIFFERENCE = DIFFERENCE - UNITS-STILL-NOT-PURCHASED</td>
</tr>
<tr>
<td></td>
<td>UNITS-STILL-NOT-PURCHASED = 0</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.3.1.1.2</td>
<td>Otherwise (Units Req Adjust is Greater)</td>
</tr>
<tr>
<td>1.3.1.1.1.2.1.3.1.1.2.1</td>
<td>Set UNITS-STILL-NOT-PURCHASED = UNITS-STILL-NOT-PURCHASED - DIFFERENCE</td>
</tr>
<tr>
<td></td>
<td>DIFFERENCE = 0</td>
</tr>
</tbody>
</table>
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY PR

Process Number: 3-2-4.PN

1.3.1.1.1.2.1.4 If DIFFERENCE > 0

1.3.1.1.1.2.1.4.1 Then

1.3.1.1.1.2.1.4.1.1 Set PR-AVAILABLE-TO-PROMISE = PR-AVAILABLE-TO-PROMISE + DIFFERENCE

1.3.1.1.1.2.1.5 If NEW-PROMISED-BY-PR = 0

1.3.1.1.1.2.1.5.1 Then

1.3.1.1.1.2.1.5.1.1 Delete PR-ORDER-REC in PR-ORDER using PR-ORDER-REC-ORDER, PR-ORDER-REC-ITEM, and PR-ORDER-REC-PR as key

1.3.1.1.1.2.1.5.2 Otherwise

1.3.1.1.1.2.1.5.2.1 Set PR-ORDER-REC-UNITS = NEW-PROMISED-BY-PR

1.3.1.1.1.2.1.5.2.2 Write PR-ORDER-REC in PR-ORDER using PR-ORDER-REC-ORDER, PR-ORDER-REC-ITEM, and PR-ORDER-REC-PR as key

1.3.1.1.1.2.1.6 Write PR-REC in PR RECORDS using PR-REC-NUMBER as key

1.3.1.1.1.3 If NEW-PROMISED-BY-PR is greater than PROMISED-BY-PR in ORDER-REC

1.3.1.1.1.3.1 Then
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY PR
Process Number: 3-2-4.PN

1.3.1.1.1.3.1.1 Set DIFFERENCE =
( NEW-PROMISED-BY-PR )
- ( PROMISED-BY-PR in ORDER-REC )

1.3.1.1.1.3.1.2 Read PR-REC in PR RECORDS
using PR-REC-NUMBER as key

1.3.1.1.1.3.1.3 If PR-AVAILABLE-TO-PROMISE > 0
1.3.1.1.1.3.1.3.1 Then
1.3.1.1.1.3.1.3.1.1 If PR-AVAILABLE-TO-PROMISE
is greater than DIFFERENCE
1.3.1.1.1.3.1.3.1.1.1 Set
PR-AVAILABLE-TO-PROMISE =
PR-AVAILABLE-TO-PROMISE
- DIFFERENCE
DIFFERENCE = 0

1.3.1.1.1.3.1.3.1.1.2 Otherwise
( Available to Prom is Smaller )

1.3.1.1.1.3.1.1.2.1 Set
DIFFERENCE = DIFFERENCE - PR-AVAILABLE-TO-PROMISE
PR-AVAILABLE-TO-PROMISE = 0

1.3.1.1.1.3.1.4 If DIFFERENCE > 0
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY PR

Process Number: 3-2-4.PN

1.3.1.1.3.1.4.1 Set UNITS-STILL-NOT-PURCHASED = UNITS-STILL-NOT-PURCHASED + DIFFERENCE

1.3.1.1.3.1.5 Write PR-ORDER-REC in PR-ORDER using PR-ORDER-REC-ORDER PR-ORDER-REC-ITEM and PR-ORDER-REC-PR as key

1.3.1.1.3.1.6 Write PR-REC in PR RECORDS using PR-REC-NUMBER as key

1.3.1.1.4 If NEW-PROMISED-BY-PR is less than PROMISED-BY-PR in ORDER-REC

1.3.1.1.4.1 Then

1.3.1.1.4.1.1 Set DIFFERENCE = ( PROMISED-BY-PR in ORDER-REC ) - ( NEW-PROMISED-BY-PR )

PROMISED-BY-PR in ORDER-REC = PROMISED-BY-PR - DIFFERENCE

1.3.1.1.4.1.2 If UNITS-REQ-ADJUSTMENT in ORDER-REC > 0

1.3.1.1.4.1.2.1 Then

1.3.1.1.4.1.2.1.1 If UNITS-REQ-ADJUSTMENT <= DIFFERENCE

1.3.1.1.4.1.2.1.1.1 Then

1.3.1.1.4.1.2.1.1.1.1 Set DIFFERENCE = DIFFERENCE - UNITS-REQ-ADJUSTMENT
PROCESS NARRATIVE

Process Name: CHANGE PROMISED BY PR

Process Number: 3-2-4.PN

Page: 07

UNITS-REQ-ADJUSTMENT = 0

1.3.1.1.1.4.1.2.1.1.2 Otherwise (Units Req Adjust is Greater)

1.3.1.1.1.4.1.2.1.1.2.1 Then

1.3.1.1.1.4.1.2.1.1.2.1.1 Set UNITS-REQ-ADJUSTMENT = UNITS-REQ-ADJUSTMENT - DIFFERENCE

DIFFERENCE = 0

1.3.1.1.1.4.1.3 If DIFFERENCE > 0

1.3.1.1.1.4.1.3.1 Then

1.3.1.1.1.4.1.3.1.1 Set UNITS-STILL-NOT-PROMISED in ORDER-REC = UNITS-STILL-NOT-PROMISED + DIFFERENCE

1.3.1.1.4.2 Otherwise (New Promised By Hand is Greater)

1.3.1.1.4.2.1 Set DIFFERENCE = (NEW-PROMISED-BY-PR)

- (PROMISED-BY-PR in ORDER-REC)

PROMISED-BY-PR in ORDER-REC = PROMISED-BY-PR + DIFFERENCE

1.3.1.1.4.2.2 If UNITS-STILL-NOT-PROMISED in ORDER-REC > 0

1.3.1.1.4.2.2.1 Then
1.3.1.1.4.2.2.1.1 If UNITS-STILL-NOT-PROMISED is greater than DIFFERENCE

1.3.1.1.4.2.2.1.1.1 Then

1.3.1.1.4.2.2.1.1.1.1 Set UNITS-STILL-NOT-PROMISED = UNITS-STILL-NOT-PROMISED - DIFFERENCE

DIFERENCE = 0

1.3.1.1.4.2.2.1.1.2 Otherwise (Difference is Greater)

1.3.1.1.4.2.2.1.1.2.1 Set DIFFERENCE = DIFFERENCE - UNITS-STILL-NOT-PROMISED

UNITS-STILL-NOT-PROMISED = 0

1.3.1.1.4.2.3 If DIFFERENCE > 0

1.3.1.1.4.2.3.1 Then

1.3.1.1.4.2.3.1.1 Set UNITS-REQ-ADJUSTMENT in ORDER-REC = UNITS-REQ-ADJUSTMENT + DIFFERENCE

1.3.1.1.5 Write ORDER-REC using ORDER-REC-NUMBER and ORDER-REC-ITEM as key.
PROCESS NARRATIVE

Process Name: TRANSFER INFO

Process Number: 3-2-5.PN Page: 01

1. Read User Option
1.1 If Option is "Transfer From PO to MPS"
1.1.1 Then Read PO-MPS-INFO
1.1.1.1 Read PO-REC in PO RECORDS using PO-REC-NUMBER = PO-MPS-INFO-PO
1.1.1.2 Set UNITS-REQUIRED = PO-REC-UNITS
   - PO-AVAILABLE-TO-PROMISE
   + UNITS-REQ-ADJUSTMENT in PO-REC
1.1.1.3 Read MPS-REC in MPS-REC using MPS-REC-NUMBER = PO-MPS-INFO-MPS
1.1.1.3.1 If MPS-AVAILABLE-TO-PROMISE < UNITS-REQUIRED
   or MPS-REC-TYPE not equal PO-REC-TYPE
   or MPS-REC-SUBCODE not equal PO-REC-SUBCODE
1.1.1.3.1.1 Then
1.1.1.3.1.1.1 Set OLD-UNITS = MPS-REC-UNITS
   MPS-REC-UNITS = MPS-REC-UNITS
   + UNITS-REQUIRED
   - MPS-AVAILABLE-TO-PROMISE
   MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE
   + MPS-REC-UNITS
   - OLD-UNITS
1.1.1.3.1.1.2 Go Back to 1.1.1.3.1
### PROCESS NARRATIVE

**Process Name:** TRANSFER INFO

**Process Number:** 3-2-5.PN

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.3.1.2 Otherwise (Units available for transfer)</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.1.2.1 If MPS-REC-AVAILABLE-TO-PROMISE ( \geq ) (PO-REC-UNITS + UNITS-REQ-ADJUSTMENT in PO-REC)</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.1.2.1.1 Then Set MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE - PO-REC-UNITS - UNITS-REQ-ADJUSTMENT in PO-REC</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.1.2.1.1.1 If MPS-AVAILABLE-TO-PROMISE ( \geq ) PO-AVAILABLE-TO-PROMISE</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.1.2.1.1.1.1 Then Set MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE - PO-AVAILABLE-TO-PROMISE</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.1.2.1.1.1.2 Otherwise Set MPS-AVAILABLE-TO-PROMISE = 0</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.1.2.1.2 Otherwise Set MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE - UNITS-REQUIRED</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.2 Set MPS-DESTINY = PO-MPS-INFO-DESTINY</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.3 If PO-REC-AVAILABLE-TO-PROMISE &lt; MPS-REC-UNITS</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.3.1 Then</td>
<td></td>
</tr>
<tr>
<td>1.1.1.3.3.1.1 Read PO-ORDER-REC in PO-ORDER using PO-ORDER-REC-PO = PO-REC-NUMBER</td>
<td></td>
</tr>
</tbody>
</table>
**PROCESS NARRATIVE**

Process Name: TRANSFER INFO  
Process Number: 3-2-5.PN  
Page: 03

<table>
<thead>
<tr>
<th>1.1.1.3.3.1.1.1</th>
<th>For each PO-ORDER-REC</th>
</tr>
</thead>
</table>
| 1.1.1.3.3.1.1.1.1 | Read MPS-ORDER-REC in MPS-ORDER  
using MPS-ORDER-REC-ORDER = PO-ORDER-REC-ORDER  
MPS-ORDER-REC-ITEM = PO-ORDER-REC-ITEM  
and MPS-ORDER-REC-MPS = MPS-REC-NUMBER |
| 1.1.1.3.3.1.1.1.1.1 | If can’t find MPS-ORDER-REC |
| 1.1.1.3.3.1.1.1.1.1.1 | Then Set  
MPS-ORDER-REC-ORDER = PO-ORDER-REC-ORDER  
MPS-ORDER-REC-ITEM = PO-ORDER-REC-ITEM  
MPS-ORDER-REC-MPS = MPS-REC-NUMBER |
| 1.1.1.3.3.1.1.1.2 | Set MPS-ORDER-REC-UNITS = PO-ORDER-REC-UNITS |
| 1.1.1.3.3.1.1.1.3 | Delete PO-ORDER-REC in PO-ORDER  
using PO-ORDER-REC-ORDER  
PO-ORDER-REC-ITEM  
and PO-ORDER-REC-PO as key |
| 1.1.1.3.3.1.1.1.4 | Write MPS-ORDER-REC in MPS-ORDER  
using MPS-ORDER-REC-ORDER  
MPS-ORDER-REC-ITEM  
and MPS-ORDER-REC-MPS as key |
| 1.1.1.4 | Write MPS-REC in MPS RECORDS  
using MPS-REC-NUMBER as key |
<p>| 1.1.2 | Ask User Option |
| 1.1.2.1 | If Option is &quot;Delete PO-REC&quot; |</p>
<table>
<thead>
<tr>
<th>PROCESS NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name: TRANSFER INFO</td>
</tr>
<tr>
<td>Process Number: 3-2-5.PN</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.2.1.1 Then Delete PO-REC in PO-RECORDS using PO-REC-NUMBER as key</td>
<td></td>
</tr>
<tr>
<td>1.1.2.1.2 Otherwise ( Keep the record )</td>
<td></td>
</tr>
<tr>
<td>1.1.2.1.2.1 Set PO-REC-AVAILABLE-TO-PROMISE = PO-REC-UNITS UNITS-REQ-ADJUSTMENT in PO-REC = 0</td>
<td></td>
</tr>
<tr>
<td>1.1.2.1.2.2 Write PO-REC in PO RECORDS using PO-REC-NUMBER as key</td>
<td></td>
</tr>
<tr>
<td>1.2 If Option is &quot; Transfer From PR to MPS &quot;</td>
<td></td>
</tr>
<tr>
<td>1.2.1 Then Read PR-MPS-INFO</td>
<td></td>
</tr>
<tr>
<td>1.2.1.1 Read PR-REC in PR RECORDS using PR-REC-NUMBER = PR-MPS-INFO-PR</td>
<td></td>
</tr>
<tr>
<td>1.2.1.2 Set UNITS-REQUIRED = UNITS-STILL-NOT-PURCHASED - PR-AVAILABLE-TO-PROMISE</td>
<td></td>
</tr>
<tr>
<td>1.2.1.3 Read MPS-REC in MPS-REC using MPS-REC-NUMBER = PR-MPS-INFO-MPS</td>
<td></td>
</tr>
<tr>
<td>1.2.1.3.1 If MPS-AVAILABLE-TO-PROMISE &lt; UNITS-REQUIRED or MPS-REC-TYPE not equal PR-REC-TYPE or MPS-REC-SUBCODE not equal PR-REC-SUBCODE or UNITS-STILL-NOT-PURCHASED in PR-REC = 0</td>
<td></td>
</tr>
<tr>
<td>1.2.1.3.1.1 Then</td>
<td></td>
</tr>
<tr>
<td>1.2.1.3.1.1.1 Set OLD-UNITS = MPS-REC-UNITS</td>
<td></td>
</tr>
<tr>
<td>PROCESS NARRATIVE</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>Process Name: TRANSFER INFO</td>
<td></td>
</tr>
<tr>
<td>Process Number: 3-2-5.PN</td>
<td></td>
</tr>
</tbody>
</table>

**MPS-REC-UNITS**

\[
MPS-REC-UNITS = MPS-REC-UNITS + UNITS-REQUIRED - MPS-AVAILABLE-TO-PROMISE
\]

**MPS-AVAILABLE-TO-PROMISE**

\[
MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE + MPS-REC-UNITS - OLD-UNITS
\]

1.2.1.3.1.1.2 Go Back to 1.1.1.3.1

1.2.1.3.1.2 Otherwise (Units available for transfer)

1.2.1.3.1.2.1 If MPS-REC-AVAILABLE-TO-PROMISE > = UNITS-STILL-NOT-PURCHASED

1.2.1.3.1.2.1.1 Then Set MPS-AVAILABLE-TO-PROMISE =

\[
MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE - UNITS-STILL-NOT-PURCHASED
\]

1.2.1.3.1.2.1.2 Otherwise Set MPS-AVAILABLE-TO-PROMISE =

\[
MPS-AVAILABLE-TO-PROMISE = MPS-AVAILABLE-TO-PROMISE - UNITS-REQUIRED
\]

1.2.1.3.2 Set MPS-DESTINY = PR-MPS-INFO-DESTINY

\[
UNITS-ALLOCATED = PR-REC-UNITS - PR-AVAILABLE-TO-PROMISE
\]

\[
UNITS-PURCHASED = PR-REC-UNITS - UNITS-STILL-NOT-PURCHASED
\]

1.2.1.3.3 If UNITS-ALLOCATED > UNITS-PURCHASED
1.2.1.3.3.1 Then
1.2.1.3.3.1.1 Set UNITS-TO-RELOCATE = UNITS-ALLOCATED - UNITS-PURCHASED
1.2.1.3.3.1.2 While UNITS-TO-RELOCATE > 0
1.2.1.3.3.1.2.1 Read PR-ORDER-REC in PR-ORDER using PR-ORDER-REC-PR = PR-REC-NUMBER
1.2.1.3.3.1.2.1.1 For each PR-ORDER-REC
1.2.1.3.3.1.2.1.1.1 Read MPS-ORDER-REC in MPS-ORDER using MPS-ORDER-REC-ORDER = PR-ORDER-REC-ORDER MPS-ORDER-REC-ITEM = PR-ORDER-REC-ITEM and MPS-ORDER-REC-MPS = MPS-REC-NUMBER
1.2.1.3.3.1.2.1.1.1 If can’t find MPS-ORDER-REC Then Set
1.2.1.3.3.1.2.1.1.1.1 MPS-ORDER-REC-ORDER = PR-ORDER-REC-ORDER MPS-ORDER-REC-ITEM = PR-ORDER-REC-ITEM MPS-ORDER-REC-MPS = MPS-REC-NUMBER
1.2.1.3.3.1.2.1.1.2 Set MPS-ORDER-REC-UNITS = PR-ORDER-REC-UNITS
1.2.1.3.3.1.2.1.1.2.1 UNITS-TO-RELOCATE = UNITS-TO-RELOCATE - PR-ORDER-REC-UNITS
PROCESS NARRATIVE

Process Name: TRANSFER INFO

Process Number: 3-2-5.PN

1.2.1.3.3.1.2.1.1.3 Delete PR-ORDER-REC in PR-ORDER
    using PR-ORDER-REC-ORDER
    PR-ORDER-REC-ITEM
    and PR-ORDER-REC-PR as key

1.2.1.3.3.1.2.1.1.4 Write MPS-ORDER-REC in MPS-ORDER
    using MPS-ORDER-REC-ORDER
    MPS-ORDER-REC-ITEM
    and MPS-ORDER-REC-MPS as key

1.2.1.4 Write MPS-REC in MPS RECORDS
    using MPS-REC-NUMBER as key

1.2.2 If UNITS-STILL-NOT-PURCHASED in PR-REC =
    PR-REC-UNITS

1.2.2.1 Then

1.2.2.1.1 Ask User Option

1.2.2.1.1.1 If Option is "Delete PR-REC"

1.2.2.1.1.1 Then Delete PR-REC in PR-RECORDS
    using PR-REC-NUMBER as key

1.2.2.1.1.2 Otherwise Set PR-AVAILABLE-TO-PROMISE =
    UNITS-STILL-NOT-PURCHASED

1.2.2.2 Otherwise Set UNITS-STILL-NOT-PURCHASED = 0
    PR-AVAILABLE-TO-PROMISE = 0

1.2.3 Write PR-REC in PR RECORDS
    using PR-REC-NUMBER AS KEY.
APPENDIX B

FRAMEWORKS
Allocate Order Framework - (2 of 2)

1. ORDER
   - ANY AVAILABLE ON HAND?
     - YES: ALLOCATE ON HAND
     - NO: ANY AVAILABLE PO?
   - NO: ANY AVAILABLE PR?
     - YES: ALLOCATE PURCHASE REQUIREMENT
     - NO: ANY AVAILABLE MPS?
   - NO: GENERATE REQ MPS
   - ORDER TOTALLY ALLOCATED?
     - YES: ORDER
     - NO: ANY AVAILABLE ON HAND?

2. ORDERS
   - ANY AVAILABLE MPS?
     - YES: ALLOCATE MPS
     - NO: ANY AVAILABLE MPS?

3. ALLOCATE ORDER
   - ORDER TOTALLY ALLOCATED?
     - YES: ORDER
     - NO: ANY AVAILABLE PO?
Control Delivery Framework

On Hand Inventory at the Sign Shop

ON HAND PROMISED TO DISTRICT?

YES

GENERATE DELIVERY LIST

UPDATE ORDER DELIVERED

USER

ORDERS
Manage Demand, MPS & Capacity Framework

Demand Forecast

Inventory at Districts

USER

GENERATE NET DEMAND

Net Demand

GENERATE MPS

Inventory at Sign Shop

FIND FEASIBLE PLAN

Available Capacity

Purchase Requirements

Inventory at Sign Shop

R.M. Purchase Requirements

GENERATE R.M. PR

Production Orders
Find Feasible Capacity Framework

MPS

Capacity Bills

EXECUTE WORK LOAD

REPORT WORK LOAD

Resource Records

CHANGE MPS AND PO

DELETE MPS FROM PLAN

APPROVE PLAN

USER

PO

PR

Production Orders
Purchase PR and RM-PR
& Receiving Purchase Orders

- Generate PR Report
- User
- Purchase PR
  - Purchase Requirements
  - R.M. Purchase Requirements
- Receive Purchase Order
  - Purchase Orders
    - Orders at Inventory System
Control Production Framework

District to Deliver

Inventory at Sign Shop

Production Orders

SORT PO TO SCHEDULE

CHECK R.M. AVAILABILITY

FIND FEASIBLE SCHEDULE

Available Capacity

IDENTIFY AVAILABLE CAPACITY

UPDATE PO COMPLETED

ORDERS

Schedule Lists
Find Feasible Schedule Framework

PO

LOAD PO AT NON-SHARED RESOURCES

LOAD PO AT SHARED RESOURCES

REPORT SCHD WORK LOAD

USER

APPROVE SCHEDULE

DISCHARGE PO FROM LIST

Schedule Lists

Capacity Bills

Resource Records
The Design of an Integrated Production and Inventory Control System for a Traffic Sign Shop

ABSTRACT

The West Virginia Department of Highways (WVDOH) is the agency responsible for maintaining the state's roadway system. One of the activities involved in this responsibility is to supply traffic signs for newly constructed highways and to replace those signs that are either deteriorated, stolen or damaged. To continually provide these signs for installation in the field, the WVDOH maintains ten warehouses (districts) in different locations in the state, and a central sign shop which produces and distributes signs to these districts.

This thesis describes the logical design for an integrated system to control the inventory and the production of the WVDOH central sign shop. The objective of this proposed system is to provide management with a decision support tool for 1) controlling the orders issued by the district, 2) planning, scheduling and controlling the production at the sign shop, and 3) planning and controlling the raw materials and subcontracted parts inventory.

The logic for this system was developed using structured architecture. The proposed design is documented in the form of data flows diagrams, a data dictionary, and process narratives.
This integrated system will provide WVDOH management information that will enable them to reduce inventory levels at the districts and central sign shop, improve production lead times, and reduce signs shortage. The implementation of such system will also reduce manual work in planning and controlling the inventory and improve information accuracy.