UNIVERSITY OF CALGARY	Course Number : SENG 609.22	Course Name: Agent Based Software Engineering
	Session : Fall, 2005	Department: Electrical and Computer Engineering
		Document Type: Project SRS

Software Requirement Specification And Design Document

Global Parts Production e-System

(GPP e-System)

Submitted By:
Manpreet Hundal
mhundal@ece.ualberta.ca

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Manpreet Hundal	09/10/2005	Initial Draft	1.0 Draft 1
Manpreet Hundal	18/10/2005	Revision of System Features	1.0 Approved
Manpreet Hundal	23/10/2005	Integration of Document	1.0 Final Draft

Software Requirement Specification Document

Global Parts Production e-System

(GPP e-System)

1. Introduction

1.1 Purpose:

The purpose of this document is to identify unambiguously the user requirements and clearly define both functional and nonfunctional requirements of GPP e-SYSTEM. In addition, this document is intended to cover technical goals as well as objectives of the proposed System.

1.2 Project scope and product features:

The proposed solution focuses on developing the automation of production system (GPP e-SYSTEM). Businesses today are rapidly moving towards process automation, each achieving and benefiting from it to varying degrees. Process automation, the next evolutionary step after the definition of a suitable process itself, usually results in an extremely streamlined flow that is manageable and flexible. Workflow engine will be an integral part of this process, thus catering to the business process management needs.

GPP e-SYSTEM will maintain the databases of the following features:

- Work Order
- Customers
- Machines
- Parts

1.3 References Materials:

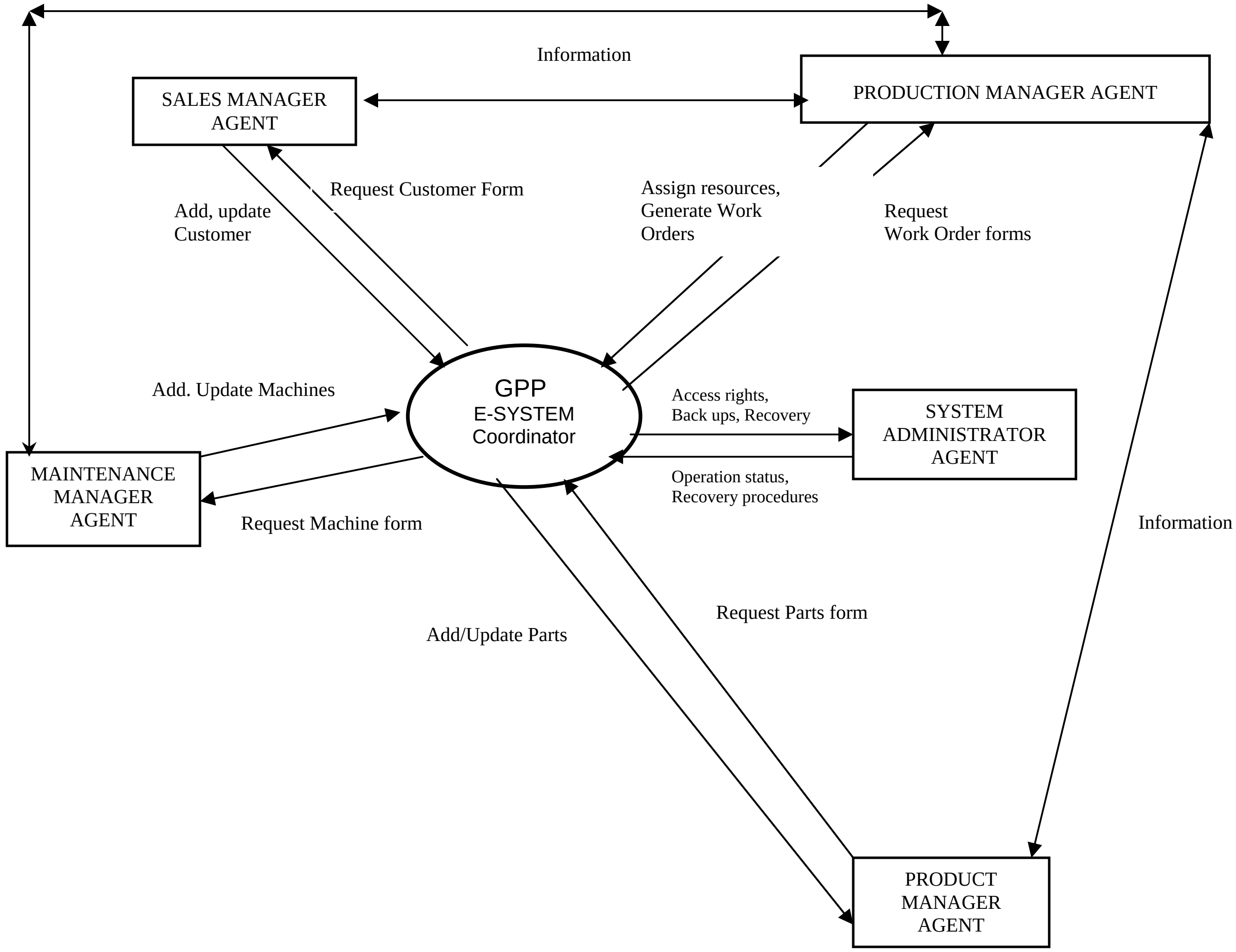
Demo Presentation

2. Overall Description

2.1 Project perspective:

This new production e-system replaces the current manual processes of the work flow. The Context diagram in Figure1 explains the entities and their interfaces.

Figure 1- Context Diagram for GPP e-System



2.2 User Classes and characteristics:

Sales manager:

- Responsible for managing customer's database by adding and updating customer's details.

Production Manager:

- Plans the work orders and is responsible for resource allocations and then generates Work Order and assign resources.

Maintenance Manager:

- Maintains machines database and update as required.

Product Manager:

- Manages the metallic parts catalogue.

System Administrator:

- Responsible for scheduling and performing backup at the complete database level as well as the individual objects level. It is also responsible for recovery of the database or objects in the event of hardware/software failure.

2.3 Operating environment:

- The production system shall operate on IE
- The Production System shall GUI environment
- The Production System shall maintain secure accessibility.

2.4 Design and implementation constraints:

- Application will be designed by using the multi agent methodology called GAIA.
- Application client shall be running on the Internet Explorer
- Application Data shall be stored in the SQL Database
- Application shall be running on the Windows 2000 / Windows NT operating system.
- Application shall be deployed on the IIS application server.

2.5 User documentation:

The first time a new user can request for a demo and documentation of the application from the IT support person.

3. Role Schemas of Agents

3.1 Product Manager Agent

Role Schema: ProductManager (PRM)
Description: Responsible for updating the parts database having the information of all the metallic parts manufactured by the company.
Protocols and Activities: GetPartDetails, AddPartDetails, UpdatePartDetails, PrintPartDetails
Permissions: Generates partDetails //creates new metallic part details Reads partDetails //reads metallic part details Update partDetails //updates metallic part details Print partDetails
Responsibilities Liveness: ProductManager= AddPartDetails |GetPartDetails| GetPartDetails. UpdatePartDetails | GetPartDetails. PrintPartDetails Safety: ■ true

3.2 Sales Manager Agent

Role Schema: SalesManager (SM)
Description: Responsible for updating the customers database
Protocols and Activities: GetCustomerDetails, AddCustomerDetails, UpdateCustomerDetails, PrintCustomerDetails
Permissions: Generates customerDetails //creates new customer details reads customerDetails //read customer details update customerDetails //update customer details Print customerDetails
Responsibilities Liveness: SalesManager= (AddCustomerDetails) | (GetCustomerDetails)| (GetCustomerDetails. UpdateCustomerDetails)| (GetCustomerDetails. PrintCustomerDetails) Safety: ■ true

3.3 Maintenance Manager

Role Schema: MaintenanceManager (MM)
Description: Responsible for updating the machines database
Protocols and Activities: GetMachineDetails, AddMachineDetails, UpdateMachineDetails, PrintMachineDetails
Permissions: Generates machineDetails //creates new machine details reads machineDetails //reads machine details update machineDetails //update machine details Print machineDetails
Responsibilities Liveness: MaintenanceManager= (AddMachineDetails) | (GetMachineDetails)| (GetMachineDetails. UpdateMachineDetails) | (GetMachineDetails. PrintMachineDetails) Safety: ■ true

3.4 Production Manager

Role Schema: ProductionManager (PM)
Description: Responsible for creation of work orders and reads the information from the customer, machine and parts databases.
Protocols and Activities: GetMachineDetails, GetPartDetails, GetCustomerDetails, AddWorkOrderDetails, GetWorkOrderDetails ,UpdateWorkOrderDetails, PrintWorkOrderDetails
Permissions: Generates workOrderDetails //creates new workOrder details Reads part //read part details machine //read machine details customer //read customer details workOrderDetails //read workOrder details Update workOrderDetails //updates workOrder details Print workOrderDetails
Responsibilities Liveness: ProductionManager= GetPartDetails. GetCustomerDetails. GetMachineDetails .AddWorkOrderDetails | GetWorkOrderDetails| GetWorkOrderDetails.UpdateWorkOrderDetails | GetWorkOrderDetails. PrintWorkOrderDetails Safety: True

3.5 System Administrator

Role Schema: SystemAdministrator (SA)
Description: The role is responsible for scheduling and performing backup at the complete database level as well as the individual objects level. It is also responsible for recovery of the database or objects in the event of hardware/software failure.
Protocols and Activities: DBBackup, ObjectBackup, OfflineStorage, DBRecovery, ObjectRecovery, AcceptBackupSchedule, AcceptBackupRequest, UpdateCatalog
Permissions: reads BackUp Schedule update Catalog Database with updated backup/ recovery info
Responsibilities Liveness: SystemAdministrator= (AcceptBackupRequest. DBBackup) |(ObjectBackup. OfflineStorage) | DBRecovery |(ObjectRecovery. UpdateCatalog) Safety: ■ true

4. Interaction Protocol Definitions

4.1 Protocol: Accept / RejectMachineRequest

Purpose: This protocol allows the MaintenanceManager to accept or reject the request for use of machines for specific duration.
Initiator: ProductionManager
Responder: MaintenanceManager
Input: Request for use of machine for the specific time period
Output: Accept or Reject Request
Processing: The MaintenanceManager checks the machine status and accept or reject depending upon the availability and work load.

Requests for Machine Details and the availability for the specific duration.	
ProductionManager	MaintenanceManager
Sends the request for machine service for the specific duration and receive a response on the same from MaintenanceManager role.	

Machine to be required and duration

Accept or Reject Request

4.2 Protocol: CustomerRequest

Purpose: This protocol allows the ProductionManager to request the SalesManager for customer’s details.
Initiator: ProductionManager
Responder: SalesManager
Input: Request for customer’s details
Output: Give required information
Processing: The SalesManager checks the customer database and sends the required information.

Requests for Customer’s Details	
ProductionManager	SalesManager
Sends the request for customer’s details and receive a response from SalesManager role.	

Customer’s Name

Send Details of the customer

4.3 Protocol: PartRequest

Purpose: This protocol allows the ProductionManager to request the ProductManager for part’s details.
Initiator: ProductionManager
Responder: ProductManager
Input: Request for part’s details
Output: Give required information
Processing: The ProductManager checks the parts database and sends the required information.

Requests for Part’s Details	
ProductionManager	ProductManager
Sends the request for part’s details and receive a response from ProductManager role.	

Part’s Name

Send Details of the part

5 Use Cases

Functional Requirements

5.1 Work Order

Description and Priority

A user logged in as a Production Manager may create a new Work Order, or print/list/delete/update an existing Work Order. A work order can only be generated for a particular Sales Order, which already should exist in the system. Priority=High.

Stimulus/Response Sequences

Stimulus: User requests to generate a new Work Order record.
Response: System displays a wizard for creating a new Work Order.

Stimulus: User requests to print a given Work Order
Response: System prints the current Work Order

Stimulus: User requests to update information related to a Work Order
Response: System allows the information to be updated

Stimulus: User requests to list all the Work Orders in the system.
Response: System lists all the Work Orders

Stimulus: User requests to remove a Work Order record from the system
Response: The Work Order is removed from the system

Functional Requirements

WorkOrder.Create: User can generate a new Work Order
WorkOrder.Create.SalesOrderID: System prompts user for an ID of a Sales Order for which the Work Order is to be generated
WorkOrder.Create.SalesOrderID.Select: User specify the ID manually
WorkOrder.Create.StartDate: At creation the system automatically set the start date to current date
WorkOrder.Create.DueDate: At creation user can specify the due date by which the Work Order is to be completed
WorkOrder.Create.Cancel: User should be able to cancel creation of the Work Order at any time.
WorkOrder.Create.Cancel.Confirm: System confirms the cancellation request
WorkOrder.Create.Cancel.No: If user does not want to cancel, system does nothing
WorkOrder.Create.Cancel.Yes: If cancellation is confirmed, no record is created
WorkOrder.Create.Commit: If done creating this Work Order, commit it to database and return to the Work Order Management Tool interface

WorkOrder.Create.Status **:** System automatically sets the Status to In-Progress
WorkOrder.Update: User may request to update a Work Order. System will display the Work order. WorkOrder.Update.Status: Update the Status of the work order, which may be In-Progress/Suspended/Cancelled/Complete WorkOrder.Update.Part: A user may change information about parts listed in this Work Order (See the functional requirements for Parts and Machines). WorkOrder.Update.Part.Add: User can add a Part which is to be produced as part of a Work Order WorkOrder.Update.Part.Remove: User can remove a part from the Work Order WorkOrder.Update.Commit: User may at any time commit the changes done to Work Order (changes to parts, their operations, assignments of employees and machines). WorkOrder.Update.Commit.Confirm: System shall confirm that the changes are to be made permanent
WorkOrder.View: All details of a Work Order transaction will appear in a view which user can browse WorkOrder.View.Part: User can view details of parts WorkOrder.View.Part.Operations **:** View the operations which are required to be performed to produce any Part in the Work Order . WorkOrder.View.Part.Operation.Machines **:** View information about machines assigned to any Operation in the Work Order Operation. WorkOrder.View.Part.Operation.Employees: View information about the employees scheduled to work on any particular Operation WorkOrder.View.Employees: System will list all employees currently assigned to Operations in this Work Order WorkOrder.View.Machines **:** System will list all machines assigned to Operations in this Work Order WorkOrder.View.Operations: System will list all Operations which need to be performed in Order to complete the Work Order
WorkOrder.List **:** User can see the list of all Work Orders in the system WorkOrder.List.ByCustomerName**:** User can see the list of all Work Orders in the system sorted by Customer Name alphabetically WorkOrder.List.ByDate: User can see the list of all Work Orders in the system sorted by most recent Start Date
WorkOrder.Delete: User can delete a Work Order if it has been completed and there is an Invoice record in the system with this Work Order's ID WorkOrder.Delete.Confirm: System should confirm deletion of a Work Order
WorkOrder.Print **:** User can print all information pertaining to this Work Order

5.2 Create, view, Modify and Delete Customer

Description and Priority

The Customer *Management tool* must contain a set of functions, which must allow Sales Person to create, edit and remove information on customers. Priority=Moderate

Stimulus/Response Sequence

Stimulus: Sales person request to the system to add a new customer.
Response: System queries Sales person for information on customer such as customer ID, customer's name, telephone, fax and address.

Stimulus: Sale's person request to search for information on a customer.
Response: If the customer exists in the database, the system must allow user to retrieve information about the customer.

Stimulus: Sale's person request to modify one or more fields on customer's information.
Response: If the customer exists in the database, the system must allow user to retrieve and edit information about the customer (the only field that is not editable is customer's ID).

Stimulus: Sales Manager request to delete a customer.
Response: If the customer is without transactions associated whose status is active, system must allow user to delete the customer. Otherwise, the system must not allow delete the customer.

Functional Requirements

Customer.Create : The system must let the sale's person to add a new customer.
Customer.Create.Id : The system must create automatically this number in an incremental order beginning from 1.
Customer.Create.Name: The system must prompt the sale's person for the Customer's name.
Customer.Create.telephone: The system must prompt sale's person for entering the Customer's phone.
Customer.Create.e-mail: The system must prompt the sale's person for the Customer's email.
Customer.Create.address: The system must prompt the sale's person for the Customer's address.
Customer.Create.bill_information: The system must prompt sale's person for entering the Customer's bill information.

Customer.Search: The system must let the system user to find Customer's information based on several filters such as: customer ID, name, address and e-mail.
Customer.Search.NO: If the customer in question is not in the database, the system must show a suitable message.
Customer.Modify: The system must let the system user to edit Customer's information. Also, the system must allow user to make visible/invisible to any customer.
Customer.Modify.NO: The system must not let the system user to edit customer's ID.

Customer.Purge: The system must let the system user to remove a customer, only if there is no transaction associated with him or her.
Customer.Purge.No: The system must not allow the user to remove a client that has active transactions.

5.3 Create, View, Modify and Delete Parts

Description and Priority
The Fabricated Parts Management tool will help Product Manager to maintain (create, modify or delete) manufactured parts. Every part may be an individual part or a multiple-part consisting of *n* number of sub-parts, which may be assemblies or sub assemblies.
Priority=Moderate.

Stimulus/Response Sequence
Stimulus: Product Manager requests to create a part.
Response: System queries Product Manager for details of part(s) such as part description, operations, discounts, part drawing as well as items or parts provided from supplier.

Stimulus: Product Manager requests to change the information of a part.
Response: If the part status is *in process*, system must allow user to edit and modify the information related to the part.

Stimulus: Product Manager requests to remove a part.
Response: If the part status is *not in process*, system must allow user to delete the part or any sub part. Otherwise, the system must not allow delete any manufactured part.

Functional Requirements

Part.Create: The system must let the maintenance manager place a new part.
Part.Create.Description : The system shall prompt the product manager for the part description.(This field may be empty).
Part.Create.price: The system shall prompt the product manager for entering the part’s price.
Part.Create.price.NO: If the product manager does not provide a price for the part, the system must not let manager finish this step. (The system must display a suitable message).
Part.Create.associateparts: The system shall prompt the product manager for entering all associated parts ID (if any) as well as the number of these.
Part.Create.associateparts.NO: If the product manager does not commit associate parts, the system must let continue.
Part.Create.associatedrawing: The system must prompt the product manager for entering the associated drawing to the part.
Part.Create.associatedrawing.NO: If the product manager does not provide at least one drawing, the system must not let manager finish this step. (The system must display a suitable message).
Part.Create.operations: The system shall prompt the product manager for entering the operations in order to make the part. The order of operations must be specified.

Part.Create.operations.NO: The system must not exit of this menu without capturing at least one operation. Part.Create.operations.operationID: The system shall prompt the product manager for the operations ID. (The system must create automatically in ascendant sequence, the natural numbers corresponding to operation ID beginning from 1). Part.Create.operations.setuptime: The system shall prompt the product manager for the operation setup time (only business hours). Part.Create.operations.runtime: The system shall prompt the product manager for the operation runtime. (only business hours) Part.Create.operations.totaltime: The system shall prompt the product manager for the operation total time. (only business hours). Part.Create.operations.labourSetupTime: The system shall prompt the product manager for the operation setup time. (only business hours). Part.Create.operations.labourRunTime: The system shall prompt the product manager for the operation total time. (only business hours). Part.Create.NO: If the product manager does not commit a new part, the system must let exit from this step.
Part.Modify: The system must let the system user to modify a part whose status is *in process*. Part.Modify.Description : The system must allow users to edit a part's description. Part.Modify.price: The system shall allow the product manager for modifying the part's price. Part.Modify.associateparts: The system must allow users to edit all associated parts to the part in question. Part.Modify.associateparts.NO: If the user(s) does not commit a change in any of the associate parts, the system must let continue. Part.Modify.associatedrawing: The system must allow the product manager to edit the associated drawing to the part in case of modifying. Part.Modify.associatedrawing.NO: If the product manager does not make any modify, the system must let continue. Part.Modify.operations: The system must allow user to edit the part operations. Part.Modify.operations.NO: If the product manager does not commit any modification, the system must finish this step without any database's change. Part.Modify.operations.setuptime: The system must allow user both to edit and modify operation setup time (only business hours). Part.Modify.operations.runtime: The system must allow user both to edit and modify operation runtime. (only business hours). Part.Modify.operations.totaltime: The system must allow user both to edit and modify operation total time. (only business hours are valid data). Part.Modify.operations.labourSetupTime: The system must allow user both to edit and modify operation labour setup time. (only business hours are valid data). Part.Modify.operations.labourRunTime: The system must allow product manager for editing and modifying of operation labor run time. (only business hours). Part.Modify.NO: If the product manager does not commit a change part, the system must let exit from this step without any database's change.
Part. Purge: The system must let the system user to remove a part (or a subpart), only if it's status is *not active*. Part.Purge.No: The system must allow the user to exit from this function.

5.4 Create, View, Modify and Delete Machines

Description and Priority

A Maintenance manager whose identity has been verified may create a new machine in to the list with a list of tasks a machine can perform. He/She can update machine tasks, list the machine tasks and delete a machine, which is out of order. Priority=Moderate

Stimulus/Response Sequence

Stimulus: Maintenance Manager requests to create a new machine for adding operation tasks.
Response: System queries manager for details of the ID, arrived date, operation tasks, price, etc of a new machine.

Stimulus: Maintenance Manager requests to print a particular machine record.
Response: System queries manager for machine ID number.

Stimulus: Maintenance Manager requests to cancel or edit a particular machine.
Response: System queries manager for machine ID number.

Functional Requirements

Machines.Create: The system shall let a manager who is logged into the E-System create a new machine for entering machine details. Machines.Create.ID: The system shall prompt the manager for the Machine Id. Machines.Create.Date: The system shall ask the manager for the machine arrived Date. Machine.Create. Price: The system shall prompt for the price of a machine. Machine.Create.Create.Task: The system shall show the task pull down menu. Machine.Create.Select: The manager shall select the task from the task menu. Machine.Create.Status: The manager shall specify the status of machine is in active state or not. Machine.Create.Operation Code: The system shall prompt for operation code as the information will be exported to scheduling list of the E-System.
Machines.View: The system shall let a manager who is logged into the E-system can view a machine record. Machines.View:ID: The system shall prompt the manager for the Machine Id.
Machines.Modify: The system shall let a manager who is logged into the E-system can edit a machine record. Machines.Modify:ID: The system shall prompt the manager for the Machine Id. Machines.Modify:Status: The manager can edit the machine status depending on the machine state. Machines.Modify:Status: The manager can edit the operations task of machine using the task menu. Machines.Modify:Operation Code: The manager can edit the operations code of a machine which needs to interact with the schedule list.

Machines.Delete: The system shall let a manager who is logged into the E-system can delete a machine record.
Machines.Delete:ID: The system shall prompt the manager for the Machine Id.
Machines.Delete:Status: The system will validate the status and prompt the reconfirmation for delete.

6. External Interface Requirements

6.1 User Interfaces

- The GPP E-System interface can be viewed over the intranet in an internet browser.
- The interface will vary depending on whether the user is logged in as a system administrator/s, sales manager/s, maintenance manager/s, and production manager/s will have different options displayed to them.
- A general Help index will be available for consulting to all users, even when not logged in, and another help index will be displayed to logged in users particular to their status.
- All content will be displayed using HTML and can be browsed using keyboard and mouse.

6.2 Hardware interfaces

- The E-System is expected to be installed on office PCs with a sufficient processing capacity (E.g. Pentium III and above) and at least 2GB HDD and 512 MB RAM.

6.3 Software Interfaces

- In the current phase of the system, there is no external software interface.

7. Other Non-Functional Requirements

7.1 Performance Requirements

The system shall accommodate 50 users during the business hours.
All reports generated by the system shall be fully downloadable in no more than 15 seconds over a 50Kbps modem connection.
Responses to search queries shall take no longer than 10 seconds to load onto the screen after the user submits the query.
The system will display customized messages after the user submits information to the system.

7.2 Safety Requirements (Backup Utility)

- A user logged in as a System Administrator may perform backup procedures to backup data currently stored in the databases for safety. User may also view log files generated by the backup. A backup can be generated on all or any subset of all the records in the system.

- User may manually trigger the backup routine, to run immediately.
- Alternatively, user may schedule backups – daily, weekly, biweekly, monthly – in which case the backup procedure will run automatically. For automatic backups, user may specify the hours during which the backup should take place.

7.3 Security Requirements

- Users shall be required to log in to the E-system for all operations.
- The system shall permit only authorized users to update the databases.

8 Software Quality Attributes

8.1 Availability

The E-system shall be available to users on the Internet 95% of time during the business hours.

8.2 Robustness

If the connection between the system and the user is broken prior to a transaction being either confirmed or cancelled, the E-System shall enable the user recover the incomplete transaction.

If the user switches between the different modules with in a system then the data entered or searched by him in the previous module remains there on screen.

If the user selects the particular record in the result set then he should able to view the details of selected record and result set will remain there on the screen. No need to make the same search again.

Appendix A:

Data Dictionary

Customer = Person who buys metallic parts.

Customer ID = Client ID number who requested a metallic part.

E-system = The software system on the Internet/Intranet.

Machine = A device with several moving parts which uses power to do a part.

Machine ID number = Company ID number of machine which makes a certain part.

Machine’s status = [assigned |not assigned| act up]

Manager = Responsible for controlling and organizing the workflow process.

Part = A separate metallic piece which combines with other metallic pieces to form the whole of a metallic part.

Part descriptions = Piece of paper that tells you what part is like.

Part drawing = Picture of a part design

Part price = The price that it cost to make a metallic part, without a profit being added

Part status = [in process |finished| sent]

Part unit price = Cost of each unit part.

Process automation = Programmable series of actions that a machine takes in order to achieve a result

Work Order = Piece of paper which proves that information on making a part have been created.

Workflow = Series of actions that a machine takes in order to achieve a result

Data model:

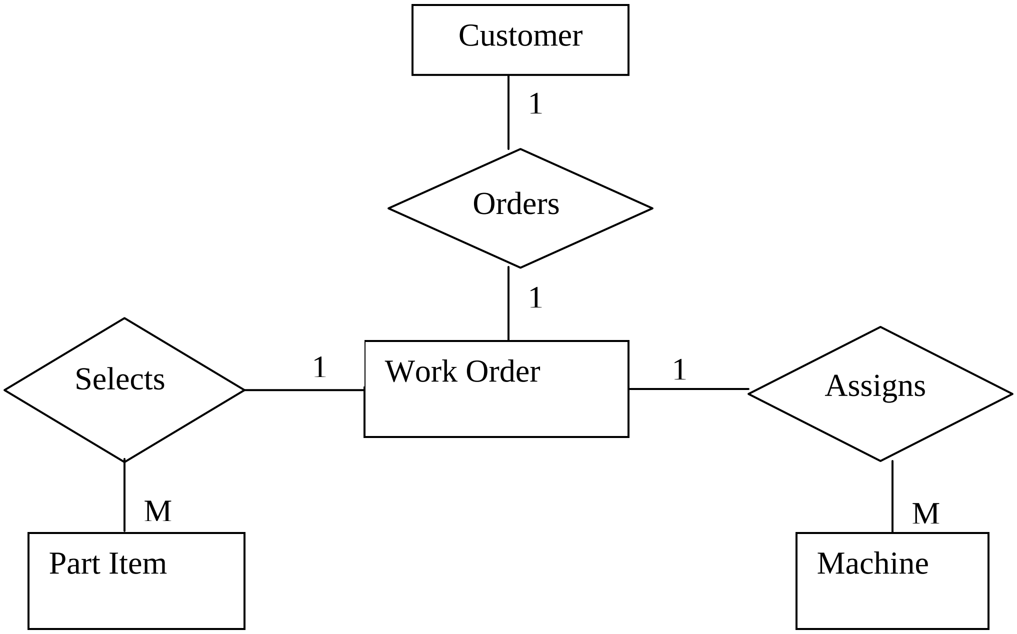


Figure 2-High-level Data model for the parts production System

Analysis model:

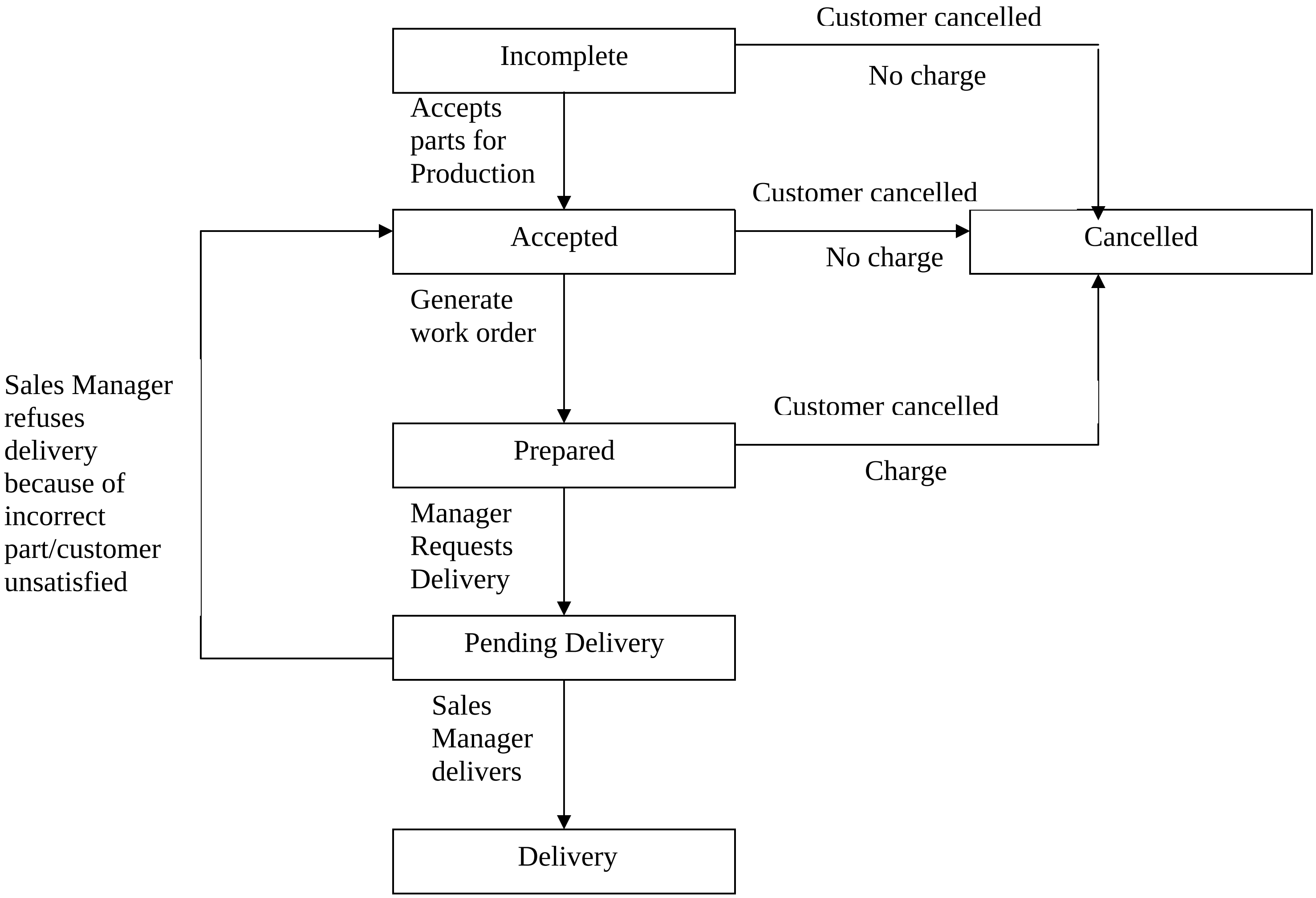


Figure 3-State diagram for Part Production status