m-Power

SRS Document

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document should be used as a guideline by the students to design the Solution Architecture for the project. The document also describes the broad scope of the project and high level logical object model. But while developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation.

m-Power Request for Meter Change

Introduction

The purpose of this document is to define scope and requirements m-Power website that gives residential and business customers the convenience of 24x7 access to their accounts and usage history, as well as make online request for Meter Change in case the need arises.

This document should be used by the development team to architect the solution the project.

Management Summary

PowerHouse sought to enhance customer service by providing round-the-clock self-service capabilities as well as Outage related information on its Web site. Since the customers lead busy lives, the company wanted to give them the convenience of sorting out billing and payments issues by viewing information online by accessing their account information during and outside normal business hours.

The proposed solution will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

Power Utilities Operations

The power utility companies are faced with immense competition and they strive to reach make sure the customers are satisfied by keeping them informed of activities happening that can impact them. The project m-Power is focussed on making the grass root level resources such as electricians responsible of owning up schedules and test report. Any delay is escalated to their Reporting Engineer for that location.

These projects are inspired from the operations of Power Utility operations covering various aspects that are focussed on serving customers with information.

Key Assumptions

1. The project owner should have understanding of power utility company operations, how they strive to service customers.

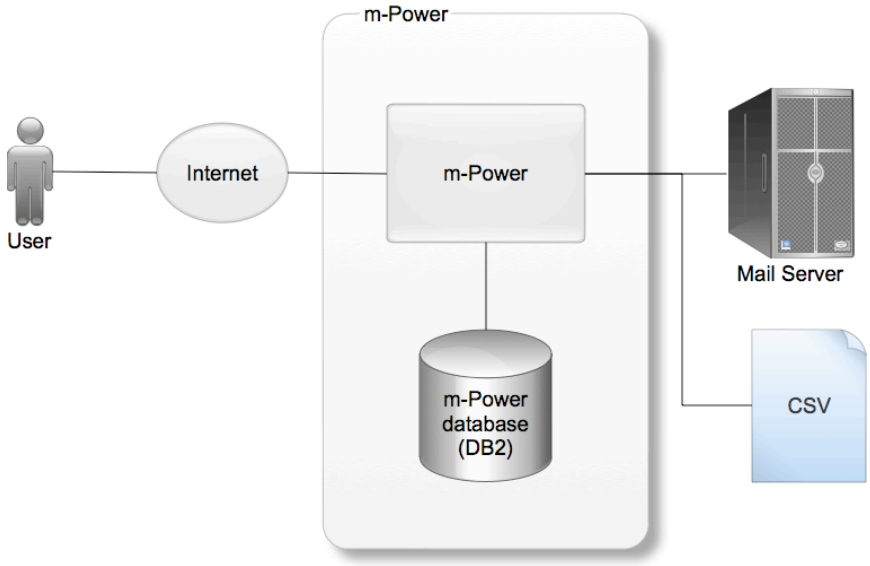
2. The customers are having connections and are registered on m-Power website. They have valid Account # with one or more connections.

3. The customer logs into the system with the user id and password provided by m-Power registration module.

High Level Architecture

m-Power's high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1. Customer
2. Engineer
3. Company Electrician
4. Administrator



m-Power Context Diagram

m-Power	m-Power allows the registered customers to make an online request for meter change. The process for Assigning Electrician, Scheduling visits, implementing and filing test report is fully automated. The m-Power engineers also publish and outage calendar for the benefit of customers.
m-Power Database	The database holds all customer, billing, request processing information
CSV	The master data like Locations, Engineer to location mapping, Electricians mapping to Engineers are uploaded via CSV
Mail Server	The notifications generated by system are sent via the Mail Server

Functional Requirements

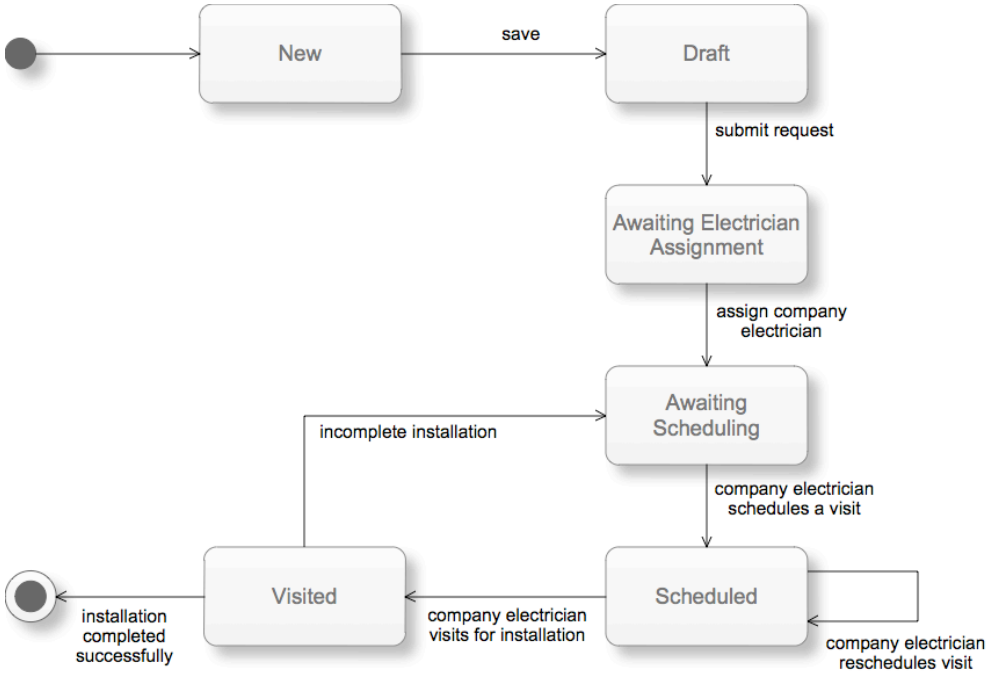
The high level functional requirements for the m-Power are outlined in the Use Case diagram described in this section.

m-Power will provide a secure user-id/password based secured login mechanism to access its services. The details of this are not outlined here. The development team is expected to create these keeping in mind the general practices followed by

the web applications. Login will be a prerequisite to use m-Power. Internal users will be provided user id/password pair separately.

The customer billing and payment information will come from an external source. The external source will generate a feed that will be uploaded by backend activity. The developer is expected to create this upload mechanism based on the fields provided in the View Bills and Payment use case.

State Diagram



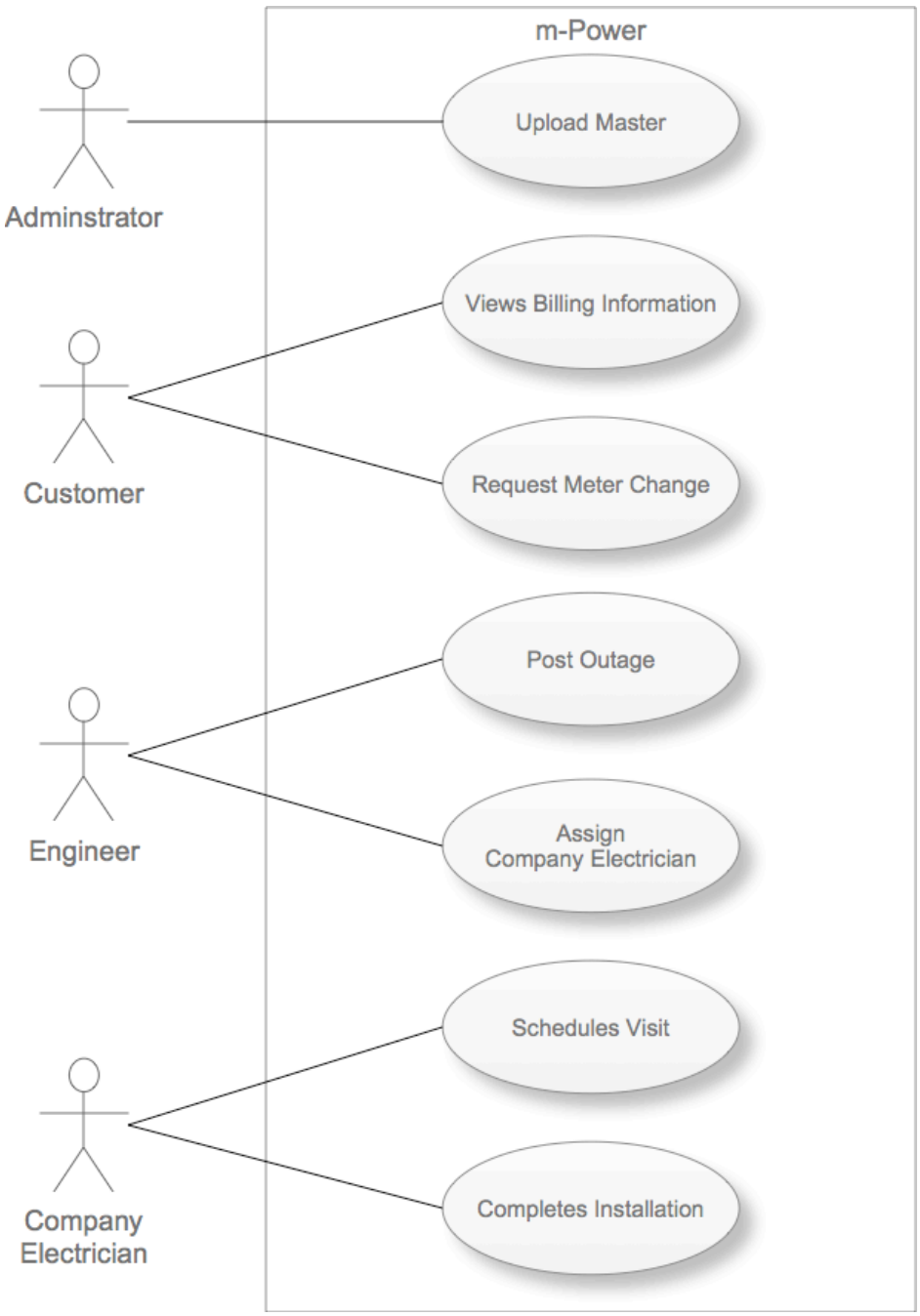
The state diagram depicts the process events that result in change of state for the Meter Change Request.

1. The request is created by the customer who is a registered user on the m-Power Website.
2. The new request can be Saved and worked upon later. Simply saving will move the request from New to the Draft state.
3. On Submission of the request, the state of request becomes Awaiting Electrician Assignment.
4. The Engineer assigns the request to Electrician, the status of the request is now Awaiting Scheduling.
5. The Electrician schedules visit with the customer after discussing offline on phone about each others convenience.
6. The Electrician updates the scheduled date of visit. The request moves into Scheduled state.
7. The Electrician updates the Test Report once meter is installed. The Test report is updated on line. A sample of test report columns is provided at the end of this section for the developer's convenience. The status of the request is moved to Visited. This is a closed state of the request for meter change.
8. In case of need to revisit the customer to complete the installation activity, the request moves to Awaiting Scheduling as the again the fixing of date happens offline.
9. The Electrician updates the system for Scheduled visit date.

10. Business rules apply for the call closure. Electrician can at best get 2 visits to get the job completed, in case of 3rd visit, the system sends escalation email to Engineer who is the reporting manager for Electrician to resolve the issue being faced by Electrician for the request.

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system. The MiS use cases are not detailed here.



Use Case Diagram

Use Cases

Upload Master

Use Case Element	Description
Number	UC.01
Application	Masters in the application are uploaded as CSV files. The power company operates at various locations, thus locations for engineers, electricians are uploaded into the system using CSV. All the electricians have engineers as their Reporting manager who assigns them work on daily basis. Locations master contains Location id, Location Engineer -location master contains Engineer id, location id (Multiple location records) Company Electrician master contains Electrician id, Electrician Name, Engineer id, Location id
Use Case Name	Upload Masters
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Upload Master** on the landing page.
Basic Flow	System prompts for the file name to be uploaded. Standard file upload dialog is presented to select a file from the local system. • The selected file data is uploaded in the related masters; if an existing record is encountered, the old details are replaced with the new details.
Alternate Flow	• In event of incorrect CSV format, system gives an error and NO data is uploaded. • Operation is cancelled
Output	System displays the number of records uploaded. It also highlights the number of records updated (i.e. already existing ones being replaced)

View Billing Information

Use Case Element	Description
Number	UC.02
Application	Customer can view previous and current bills for the account number registered in the m-Power application. The information is uploaded as a backend activity daily to ensure the customers get to view bills and payment updates at the earliest
Use Case Name	View Billing Information
Primary Actor	Customer
Secondary Actor	None
Pre-condition	Customer is a valid registered user for m-Power application.

Use Case Element	Description
Trigger	User clicks on the **Billing Information** link on the customer landing page
Basic Flow	The system displays the list of bills generated for the customer account number in a chronological order (Descending i.e. the latest bills are on top) The display is row wise Bill Date, Bill #, Amount, Payment Status The user can highlight any one bill and click on it to get a view of complete bill. The system shall pull up the detailed data as follows • Customer Account # • Connection # • Bill Date • Payment Due Date • Status (Paid / Unpaid) • Billing Cycle Start Date • Billing Cycle End Date • Taxes • Payable Break up of Net Payable comes from the following items in a bill.
Alternate Flow	If no bills are available, the system shall display message, “Billing information is not available for the <account #>
Output	None

Request Meter Change

Use Case Element	Description
Number	UC.03
Application	The customer can switch to smart meters from existing manual ones. New customer case is not included in the scope of this project
Use Case Name	Request Meter Change
Primary Actor	Customer
Secondary Actor	None
Pre-condition	Valid users of the m-Power system
Trigger	User clicks on the **Meter Change** link on the landing page

Use Case Element	Description
Basic Flow	The header of the form displays the logged in users Name, Contact Details, Account Number (If existing customer) Display multiple Connection numbers and their installation address, the user selects one of the connection number to proceed with the request. The details of the connection i.e. Existing meter number, make, model, load, date of installation are displayed. A form is displayed that allows the user to select installation of various types of meter (Smart Meter, Digital) Select Reason from a given list (Move to digital, faulty meter, damaged etc) Brief description of the reason selected. User can save the request as draft. The status becomes 'Draft' User confirms the processing of request, the record is saved and SLA of 5 days is assigned to track the next step. The state of the request is Awaiting Electrician Assignment
Alternate Flow	None
Output	Email is sent to Engineer for request processing.

Assign Company Electrician

Use Case Element	Description
Number	UC.04
Application	The request for meter change queue keeps increasing, the prime responsibility of the Engineer in that location is to ensure quick response time by assigning the request to Company Electrician
Use Case Name	Assign Electrician
Primary Actor	Engineer
Secondary Actor	None
Pre-condition	Engineer is a registered user for m-Power application.
Trigger	User clicks on the **Assign Electrician** link on the landing page
Basic Flow	The system displays the list of Requests waiting to be assigned in a chronological order Ascending (with oldest request on top) Click on any one request to view details in a read only form. Display the list of Electricians in the location of the customer. Selects one Electrician from the list. Click on Assigned will move the request to Awaiting Scheduling. The system sends notification to Electrician for the new request in their queue. Refer to State Diagram section at the bottom of Use Cases.
Alternate Flow	None
Output	Notification to Company Electrician

Schedules Visit

Use Case Element	Description
Number	UC.05
Application	Electrician views the the request and calls up the contact number of the customer. Fixes and updates the scheduled visit date in the request
Use Case Name	Schedule Visit
Primary Actor	Electrician
Secondary Actor	None
Pre-condition	Electrician is a registered user for m-Power application.
Trigger	User clicks on the **Schedule Visit** link on the landing page
Basic Flow	The system displays the list of Requests at are awaiting visit to be scheduled. The user selects the entry that has been confirmed offline for the visit. Updates the Visit Date and clicks on Confirm. The Scheduled Visit rule gets executed on save to check if the number of visits have exceeded 2. In case it has, the system sends email to the Reporting Engineer for looking to the delay in installation. The system updates the request and its status to Scheduled.
Alternate Flow	None
Output	Email to the Reporting Engineer for looking to the delay in installation.

Updates Request Status

Use Case Element	Description
Number	UC.06
Application	Post the meter installation the Test is done for all the circuits and a report is filled by the Electrician
Use Case Name	Updates Request Status
Primary Actor	Company Electrician
Secondary Actor	None
Pre-condition	Electrician is a registered user for m-Power application.
Trigger	User clicks on the **Update Request Status** link on the landing page

Use Case Element	Description

Basic Flow	The system displays the list of scheduled requests for the logged in Electrician
	
	Select the request that has to be updated.
	
	The status is changed to Completed or Incomplete
	
	If the status is Completed, then system displays a form opens giving customer details and input fields for the user to enter the test summary.
	
	User enters the following information:
	
	• Date of Testing:
	• Installation Resistance : _____ mega ohms
	• Earth leakage protection device installed Yes/No If yes,
	  • Make :
	  • Model :
	  • Capacity:
	• Testing Report Summary :
	The report is filled and saved with the customer connection # for future reference.
	
	The Use Case follows the state diagram in either case of Status marked as Completed or Incomplete.

Alternate Flow	None
Output	None

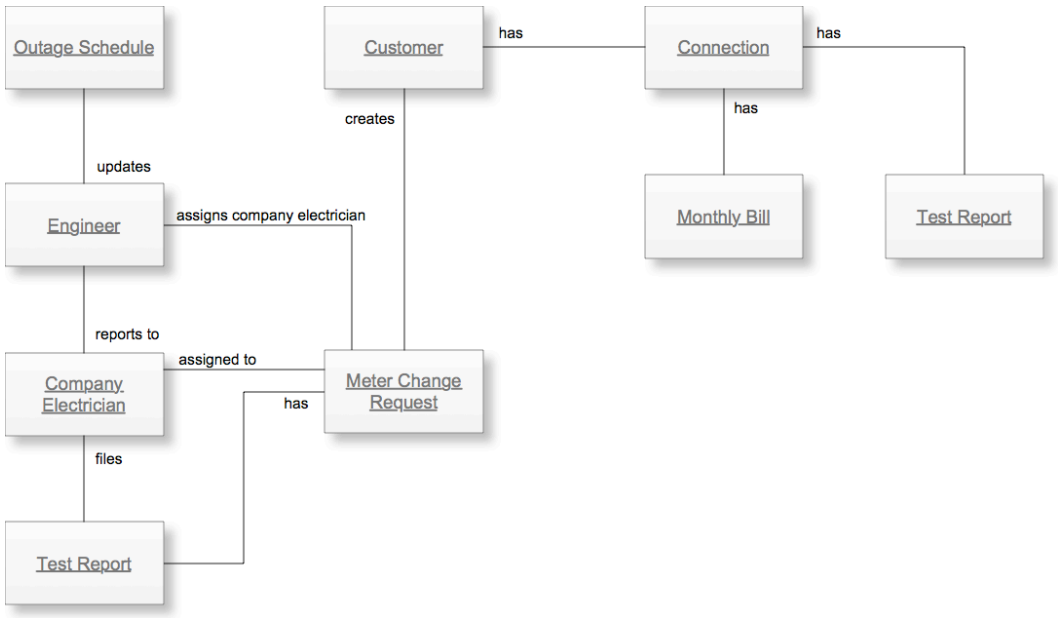
Post Outage

Use Case Element	Description
Number	UC.07
Application	The maintenance is an ongoing activity for the power company. Some of the activities are short and other for a longer duration. Thus the company publishes a schedule for 2 weeks in advance for the customers to plan better.
Use Case Name	Post Outage
Primary Actor	Engineer
Secondary Actor	None
Pre-condition	Engineer is a valid registered user for m-Power application.
Trigger	User clicks on the **Outage** link on the landing page

Use Case Element	Description
Basic Flow	The system displays locations under the logged in Engineer. User selects a location and views the outage for next 2 weeks. User enters the outage data as Date, Start Time, End Time, Activity type(Regular, Load Shedding, Infra Upgrade) For a location, there can be multiple entries of outage in a day. The system should validate any overlapping of the start or end time being entered. Each entry is saved and the next outage entry fields are displayed, till the user exits. The display of outage is location wise in a Calendar format to the Customer. The Use case for viewing is not mentioned, The developer is required to build the calendar and give a menu option to users for viewing the outage. The outage calendar will also show the total number of hours of outage daily
Alternate Flow	
Output	None

Logical Object Model

A high level logical object model of the system is shown below. During technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.



Logical Object Model

1. Customer has one or more connections.
2. These connections when commissioned get verified by the company engineer, who writes a test report. The test report is a formal record of the connection working fine.

3. For each of the connection, a monthly bill is generated as per the consumption plus standard charges that apply as per policies.

4. For a given connection, the customer can opt for a meter change in case of Manual reading to Digital or Time of Use meter or cases where meter has got a defect or there is a genuine need to increase the load thus the new meter is required.

5. Meter change request is made online, the request gets queued for the Engineer to assign it further.

6. The Engineer assigns the request to Company Electrician.

7. Company Electrician Schedules the visit to customer premises.

8. The Test report is created and filed online for the records on successful completion of the installation.

9. The Engineer has additional responsibility of posting the Outages for locations covered by the company. Outages are normally planned and displayed as a calendar for next 2 weeks with date and time as well. The reason for outage is also entered along with each outage entry.

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimized physical database table designs.

Typically persistent classes will map to table(s) with their attributes as columns of the table. In some cases a high level object may map in to a master-child table. Invoice is one such example where it maps in to "invoice_header" and "invoice_line_item" table.

Associations between two persistent objects are realized as foreign keys to the associated objects. A foreign key is a column in one table that contains the primary key value of the associated object.

Similarly, a standard technique in relational modeling is to use an intersection entity to represent many-to-many associations. Following is a broad checklist for physical database database design:

1. Database must be properly normalized except those instances where de-normalization help improves performance. This option must be used with special care.

2. All persistent classes that use the database for persistency must map to database structures.

3. Many-to-many relationships must have an intersecting table.

4. Primary keys should be defined for each table, unless there is a performance reason not to define a primary key.

5. Indexes should be defined to optimize access.

6. Data and referential integrity constraints should be defined.

Testing Approach

Quality of the software can be achieved with basic hygiene and consistency followed during design and development of User Interface(UI), Navigation, Validations as per the business process requirement.

To ensure the project delivers acceptable quality to the customer, its important to create a checklist of the conventions to be followed across. Common checks as below are for your reference during design and development:

Common Checks	Validation Type
Page Title is valid for the feature being provided on the page	UI
Order of the Data Entry Fields is logical as per the functionality being provided by the feature	UI
Order of the Display only Fields makes viewing and understanding easy for the user	UI
Spellings and Correctness of Label for the Data Entry and Display fields	UI
The labels are not wrapping onto another row thereby adding a blank row on the page	UI
The fields with drop down are displayed in single row instead of drop down coming on the next row	UI
Data Entry field basic validations are working i.e Text field /Numbers / Dates allow data for their type only	Functional
The dates are following a standard format dd/mmm/yyyy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI
The action buttons are performing the desired action e.g. "submit" is creating a new record if there are no errors and recording all the input fields, whereas 'cancel' is not creating a new record in the database	Functional
The links provided on the forms are opening correctly.	Functional
The data feed mechanism for Read and Write files is generating a log with count of entries.	Navigation

Suggested Technical Reading

The project is aimed at making the student understand concepts of Design and Development using IBM Rational tools, Web Sphere Application Server and DB2 Database. The following reading reference is easy to understand and should be read to get a clear understanding of capabilities of the tools and how you would leverage them to execute a project.

Technical Reference	URL to access
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html
IBM Education Assistant - Rational Application Developer 7.5	http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/