

**RUP Variant Guidelines for the
Life Cycle Architecture (LCA)
deliverables for
Model-Based (System) Architecting and
Software Engineering (MBASE)**

General permission to make fair use in teaching or research of all or part of these guidelines is granted to individual readers, provided that the copyright notice of the Center for Software Engineering at the University of Southern California is given, and that reference is made to this publication. To otherwise use substantial excerpts or the entire work requires specific permission, as does reprint or republication of this material. © Center for Software Engineering, University of Southern California. All Rights Reserved. 1997-2000.

Table of Contents

1	Introduction	4
2	Operational Concept Description (OCD)	5
3	System and Software Requirements Definition (SSRD)	8
4	System and Software Architecture Description (SSAD)	10
5	Life Cycle Plan (LCP)	13
6	Feasibility Rationale Description (FRD)	15
7	Appendix	17
	A) LCA-TREE, OCD & SSRD diagrams	
	B) LCA-TREE, SSAD diagrams	
	C) RUP CD, Right & Left Frame	
	D) Microsoft Project CD, Left Side	
	E) USC-COCOMOII.2000.0	

Version control

Date	Author	Changes	Version
02/15/00	Ebru Dincel	Initial version based on MBASE 1.2.0.	1.0
02/16/00	Ebru Dincel	Added introduction section	1.1
02/24/00	Ebru Dincel	Course guidelines incorporated from MBASE, sync with MBASE 1.2.0	1.2
02/24/00	Ebru Dincel	Sync with MBASE 1.4.2	1.3
02/29/00	Ebru Dincel	Revise introduction	1.4

1) INTRODUCTION

Model-based System Architecting and Software Engineering (MBASE) is an approach that integrates the process, product, property and success models for developing a software system. There are MBASE invariants as well as variants. This document explains RUP variant guidelines for the Life Cycle Architecture deliverables for the MBASE. Overall, this guideline should help you to improve the quality of your deliverables.

Appendix A) & B) show some screen views from the recommended LCA tree to help you structure your Rose diagrams. Using these guidelines, you can put the relevant diagram type in the appropriate place in the tree. This will both enhance communication power and accuracy of your models. The LCA tree is also provided as a separate .mdl file.

Appendix C) shows a screen view from the RUP CD that has been distributed. It shows the “Tree browser” right hand frame and “Artifact Guidelines Overview” to orient you towards specific references. There are references throughout this guideline that refers to a specific place in RUP CD in case you need extra information about the diagrams, concepts, etc..

Appendix D) shows a screen view from Microsoft Project CD that has been distributed. You should be using this tool for your planning, resource allocation, etc.

Appendix E) shows a screen view from COCOMO that is available from class web site for downloading. This tool is to be used for your budget and schedule estimates.

For general references on different sections such as OCD, SSRD, SSAD, LCP and FRD, please visit that section on this guideline and find out how you can reach that information.

General Tool Guidelines: Rational Rose is the standard tool to develop the diagrams. Rose may not directly support all the diagrams (e.g., block diagrams, layered views). Some add-ins provide additional capabilities (e.g., ErWin for E-R diagrams). However, avoid having your architecture models developed with a large variety of tools, ranging from picture editing programs to other specialized software (e.g., flowcharting software).

2) Operational Concept Description (OCD)

1 Introduction

1.1 Purpose of the Operational Concept Description Description

1.2 References

2 Domain Description: See sub items, all of which must take risk into account.

2.1 Organization Background: Degree of detail risk driven.

2.2 Organization Goals: Degree of detail risk driven.

2.3 Description of Current System—High level block diagram only top two level blocks to be identified, description of the interesting systems required and degree of detail risk driven. Do not include any of your provided (new) to be architecture(s).

- (preferred) Use-Case View Class Diagram with classes of Component Stereotype nested within classes of Node Stereotype, and associations as appropriate between components and/or
- Use-Case View Collaboration Diagram with Objects of Node or Component Stereotyped Classes (preferred) with a RUP Business Object Model if needed and/or
- High level block diagram (S&C) may be needed to put the purpose in context and/or
- A combination of Component and/or Deployment View, but with a business object model if needed.

Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set for information on *Business Object Model*.

Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set and Design Set for information on *Collaboration Diagram*.

2.4 Entity Model: Use either a Business Class model or a regular class model as appropriate; risk driven detail of content with all key attributes, no types allowed, no operations allowed. High level classes or aggregated classes are enough, not too many classes. Entire systems cannot be considered entities, only entities about which info is stored in the system.

Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set and Design Set for information on *Class Diagram*.

2.5 Interaction Model: Especially for every entity and activity, Team-Collaboration Diagram and Work-Collaboration Diagram (Sequence Diagram); risk driven level of content.

Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set and Design Set for information on *Sequence Diagram*.

See Section 2.3 for reference on *Collaboration Diagram*.

2.6 Organization Activity Model: Use an activity model or a Business Activity Model, with classes restricted to those needed, risk driven level of content.

Alternatively, based on risk, you may have to include a Use-Case Model or a Sequence Diagram.

Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set and Requirements Set for information on *Activity Diagram*.

Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Requirements Set for information on *Use-Case Model* and its constituent parts.

Use-Case Model

- Actor Generalization
- Communicates Association
- Extend Relationship
- Include Relationship
- Use-Case Generalization
- Use-Case Diagram

Use-Case description templates are available at
(http://sunset.usc.edu/classes/cs577a_99/tools/UseCaseTemplate.zip).

See Section 2.5 for reference on *Sequence Diagram*.

2.7 **Current System Shortfalls:** N/A

3 **Proposed System**

3.1 **Project Goals and Constraints:** N/A

3.2 **Capabilities:** Top/System [Business] Use-Case Model with all previous tabular data in, only functional elements, 5 top level enough.

Look on RUP CD Left Frame, Artifact Guidelines Overview

under the Right Frame, Guidelines for the Business Modeling Set for information on the *Business Use-Case Model* and its constituent parts.

See Section 2.6 for reference on *Use-Case Model*.

3.3 **Levels of Service:** N/A

3.4 **Proposed System Description:** see below.

3.4.1 **Statement of Purpose:**

- (preferred) Use-Case View Class Diagram with classes of Component Stereotype nested within classes of Node Stereotype, and associations as appropriate between components and/or
- Use-Case View Collaboration Diagram with Objects of Node or Component Stereotyped Classes (preferred) with a RUP Business Object Model if needed and/or
- High level block diagram (S&C) may be needed to put the purpose in context, two top levels of nesting required, with clear identification of system boundary and/or
- Combination of Component and/or Deployment View, but with a business model if the need is real.

See Section 2.3 for reference for the diagrams mentioned above.

3.4.2 **Proposed Entities:** Use either a Business Class model or a regular class model. Regular class diagrams with all important attributes, no tabular descriptions needed, no types allowed, no operations allowed. High level classes or aggregated classes are enough, not too many classes. Newly introduced entities, relevant existing entities and their relations.

See Section 2.4 for reference on *Use-Case Model*.

3.4.3 **Proposed Interactions:** You may use Use-Case model coupled with sequence diagrams if necessary (only 5-7 top level interactions). Structured use-case descriptions required. See Section 2.5 for reference on *Sequence Diagram*. See Section 2.6 for reference on *Use-Case Model*.

3.4.4 **Proposed Activities** An Activity diagram with the identification of the proposed workflow and roles. See Section 2.6 for reference on *Activity Diagram*.

3.5 **Redressal of Current System Shortfalls:** N/A

3.6 **Effects of Operation:** N/A

3.6.1 **Operational Stakeholders:** N/A

3.6.2 **Organizational Relationships:** N/A

3.6.3 **Operational Policies and Constraints:** N/A

3.6.4 **Operational Impacts:** N/A

3.6.5 **Organizational Impacts:** N/A

4 Common Definition Language for Domain Description

5 Appendix

3) System and Software Requirements Definition (SSRD)

- 1 **Introduction:** N/A
 - 1.1 **Purpose of System and Software Requirements Description Document:** N/A
 - 1.2 **References:** N/A
- 2 **Project Requirements**
 - 2.1 **Budget and Schedule** Consistent with LCP and FRD. Exact numbers should be identified.
 - 2.2 **Development Requirements** Development platform and tools for development. COTS required for integration and testing. Versions of programming languages. Resources provided by ISD, CSE and other sources. Standards of development for hardware and software to be adhered to test data sources.
 - 2.3 **Packaging Requirements** Install and uninstall clearly identified format and media for packaging and distribution.
 - 2.4 **Implementation Requirements:** Training, manuals and guides, data preparation.
 - 2.5 **Support Environment Requirements:**
- 3 **Capability Requirements**
 - 3.1 **System Definition:** System boundary, interacting users and systems identified.
 - 3.2 **System Requirements:**
 - 3.2.1 **Nominal Requirements:** One use-case diagram (not a business use-case diagram) for the entire system requirements set, or one per mode of operation for complex systems. Classes of users should be shown as hierarchies on the use-case diagram. Each actor (user or otherwise) should be described (defined) in the Documentation text box of its specification of the actor. Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Requirements Set for information on *Use-Case Model* and its constituent parts.
 - Use-Case Model
 - Actor Generalization
 - Communicates Association
 - Extend Relationship
 - Include Relationship
 - Use-Case Generalization
 - Use-Case Diagram

Use-Case description templates are available at (http://sunset.usc.edu/classes/cs577a_99/tools/UseCaseTemplate.zip). Each use case should be described at the Expanded/Essential level [refer to **TT-03**. Rose & UML Introductory Tutorial that is available at (http://sunset.usc.edu/classes/cs577a_99/tutorials/RoseTutorialUML.pdf)].
 - 3.2.1 **Off-Nominal Requirements:** These should appear in the "Alternate Course of Action" section of the Expanded/Essential use-case description. Major, high-level exceptions should be indicated in the "Exceptional Course of Action" section.
- 4 **System Interface Requirements**
 - 4.1 **User Interface Requirements:**
 - 4.1.1 **Graphical User Interface Standards:** No more than one screenshot. Only standards of UI development.
 - 4.1.2 **Command-Line Interface Requirements:** N/A
 - 4.1.3 **Diagnostics Requirements:** N/A

- 4.2 **Hardware Interface Requirements** If scanner, bar code reader or printer interface is required.
- 4.3 **Communications Interface Requirements:** N/A
- 4.4 **Other Software Interface Requirements:** If interaction with another system is required, then identify the interfaces, e.g. SIRSI, OCLC, QTVR, browser.
- 5. **Level of Service Requirements:** Desirable and acceptable levels should be clearly identified
Short titles desirable.
- 6. **Evolution Requirements**
 - 6.1 **Capability Evolution Requirements:** Clear detailed description of the future features
 - 6.2 **Interface Evolution Requirements:** Any possible change in the user interface such as specialized client to browser.
 - 6.3 **Technology Evolution Requirements:** Browser, server, database upgrade strategy
 - 6.4 **Environment and Workload Evolution Requirements:** Growth profile of number of users, traffic, storage and workload.
- 7. **Common Definition Language for Requirements**
- 8. **Appendices**
 - A. **Standards Specifications**
 - B. **Interface Specifications**

4) System and Software Architecture Description (SSAD)

- 1 Introduction
 - 1.1 Purpose of System and Software Architecture Description Document
 - 1.2 Standards and Conventions
 - 1.3 References
- 2 [Software System] Architectural Analysis
 - 2.1 **Component Model:** Business Class Model with some high level design information for the selected architecture. Class diagrams for each block and entity from OCD. Each component can be derived from OCD Proposed System Entity Model (All Object) or OCD Block Diagram (Some Object, Some Process). Each Component can consist of only all Objects or only all Process at a time. UML package and component diagrams.
Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set for information on *Class Diagram*.
 - 2.2 **Behavior Model:** Use-Case diagrams consistent with OCD and SSRD. Exception and alternate courses of action identified Uses, extends and includes relations to discover internal use cases. Sequence diagrams for important and complex behaviors. Put each behavior in separate diagrams.
Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Requirements Set for information on *Use-Case Model* and its constituent parts
Use-Case Model
 - Actor Generalization
 - Communicates Association
 - Extend Relationship
 - Include Relationship
 - Use-Case Generalization
 - Use-Case Diagram*Use-Case description templates* are available at (http://sunset.usc.edu/classes/cs577a_99/tools/UseCaseTemplate.zip).
Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set and Design Set for information on *Sequence Diagram*.
 - 2.3 **Enterprise Model:** Business Class Model with domain classes associated with business objects, but with only few important operations and attributes from LCO level with business relevant type information, or represented as classes with structures. Elaboration of the entities in the OCD. No actors allowed in the model. Associations should be named. Proper data types required, design types are not allowed. Proper cardinality. Adequate and proper use of association types such as navigation, aggregation and generalization. Role names desirable.
Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set for information on *Class Diagram*.
- 3 System Design

- 3.1 **Architectural Views**
- 3.1.1 **System Topology:** Collaboration diagram, clear identification of blocks to be built vs. bought. Further breakdown of components. Identification of the interaction protocols. Nested <<component>>s in <<node>>s with <<interface>>classes if appropriate for the selected architecture. Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Design Set for information on *Collaboration Diagram*.
- 3.1.2 **Component Specifications:** Specification of the component in terms of the services provided and required by each.
- 3.1.3 **Framework and Protocol Specifications:** Very brief description of framework and protocol versions and important characteristics (Java VM, HTTP, JDBC, ODBC, OLE, CGI, JSP, AWT, Swing, MFC, TCP/IP, FTP and any other protocols required with versions and vendors) embedded in the components' specification where the component is a framework or protocol, interface.
- 3.1.4 **System Deployment View:** Major large <<component>>s assigned to <<node>>s with indication of hardware, operating system and netware for the architecture selected. Look on RUP CD Left Frame, Introduction, Key Concepts, Software Architecture under the Right Frame Typical Set of Architectural View for information on *Deployment View*.
- 3.1.5 **Logical Component View:** A package diagram showing how design classes are grouped. Look on RUP CD Left Frame, Introduction, Key Concepts, Software Architecture under the Right Frame Typical Set of Architectural View for information on *Logical View*.
- 3.2 **Class Model:** Class model with all classes. Should identify all classes used in the operations model. The design should be enough for someone to start coding on that basis, but not necessarily complete as the actual code would be. Class model should not be humongous, it should be partitioned for proper organization and understanding. Classes should be described in at least a couple of lines. All the attributes required to be printed out should be identified. Attribute types should either be standard design types such as string, list, tree, etc or explicitly modeled in the class model. The accessibility of classes (public, private and protected) should be properly designed. Use of role names desirable. Database schema clearly identified. Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Design Set for information on *Class Diagram*. See Section 2.2 for reference on *Sequence Diagram*.
- 3.3 **Operations Model**
- 3.1.1 **Critical Algorithms:** Identification of critical algorithms to be implemented. Pseudo code, flowcharts or activity diagram for each one. Look on RUP CD Left Frame, Artifact Guidelines Overview under the Right Frame, Guidelines for the Business Modeling Set and Requirements Set for information on *Activity Diagram*.
- 3.1.2 **Operation Specifications:** Most important services required. Should start with an actor. The sequence diagram should be one continuous flow. The message names (which might include parameter types) should be method calls that are described in the appropriate classes of the class model. All objects should belong to some class in the class model. Return flows are not necessary. See Section 2.2 for reference on *Sequence Diagram*.

- 3.4 **Configuration Model:** Identify directory structure for development including data, text, images, programs, tools as planned (as implemented at IOC).
Equipment required for deployment.
- 4 **Common Definition Language for System Design**
- 5 **Appendices**
 - 5.1 **Reference**
 - 5.2 **Vendor documents**

5) Life Cycle Plan (LCP)

1. **Introduction**
 - 1.1 **Purpose of the Life Cycle Plan Document**
 - 1.2 **Assumptions:** Relevant project related assumptions specified.
 - 1.3 **References**
2. **Milestones and Products**
 - 2.1 **Overall Life Cycle Strategy:** Overall Gantt chart needed. Clear and correct identification of stages and phases. Description of each phase correctly done along with an overview of the internal milestones.
Look on Microsoft Project CD under the left Frame for information on *Gantt Chart, Pert Chart*.
 - 2.2 **Phase Milestones and Schedules**
 - 2.2.1 **Engineering Stage:** Detailed milestone plan, listing of all the deliverables. See Section 2.1 for reference on *Gantt/Pert Chart*.
 - 2.2.2 **Production Stage:** Detailed feature lists to be constructed, tasks to be performed during construction phases, at least one build and integration in the middle of the semester, increments logically arranged, sufficient parallelism, milestones and deliverables identified.
See Section 2.1 for reference on *Gantt/Pert Chart*
 - 2.2.3 **Support Stage:** Tasks and deliverables identified, future release dates and features to be included identified.
See Section 2.1 for reference on *Gantt/Pert Chart*.
 - 2.3 **Project Deliverables**
 - 2.3.1 **Engineering Stage**
 - 2.3.2 **Production Stage Course Guidelines:** All deliverables are properly stored in the Class Archive, in accordance with the course guidelines (a "Where?"). Internal deliverables include weekly Effort Reports and weekly Status Reports.
3. **Responsibilities**
 - 3.1 **Stakeholder Responsibilities**
 - 3.1.1 **Stakeholder Representatives:** Identified all stakeholders and their detailed roles in each phase, representatives of stakeholders identified
 - 3.1.2 **Engineering Stage:**
 - 3.1.3 **Production Stage:** Identified responsible stakeholders for review, testing and training, Identified stakeholders providing technical know how
 - 3.1.4 **Support Stage:** Maintenance and administration stakeholders identified and responsibilities described.
 - 3.2 **Development Responsibilities**

Course Guidelines:
The Work breakdown structure is used to address team roles and responsibilities. Teams should develop the WBS (LCP 5.1) and the Effort Estimates (LCP 5.2), before proceeding to Section 3.

 - 3.2.1 **Development Organization Charts:** Organizational chart for roles not persons. Identify only the developers in the production phase.
 - 3.2.2 **Staffing:** Identify the staff profile, skill requirements and availability dates and levels. Exact numbers of each kind should be identified. Specify how the positions would be filled and who would be responsible for it.
Look on Microsoft Project CD under the left Frame for information on *Resource Allocation Graph*.

- 3.2.3 **Training:** Special skill needs that would be satisfied by on-the-job or specialized training. Identify resources required for that.
See Section 3.2.2 for reference on *Resource Allocation Graph*.

4 Approach

4.1 Process Management

4.1.1 Planning

- 4.1.2 **Monitoring Control:** identify how milestones will be tracked, who is responsible for monitoring and controlling what.
See Section 2.1 for reference on *Gantt/Pert Chart*.

- 4.1.3 **Risk Management and Monitoring Procedures:** Only the approach, not the same risk descriptions as in FRD. A risk management technique commonly used in is a weekly top-10 Risk Items Lists

4.1.4 Management Reviews:

4.1.5 Project Communication

- 4.1.6 **Tools, Environment, Facilities:** Development tools, infrastructure and admin personnel needs. ISD, CSE and other sources identified. Accounts for developers and development locations.

4.1.7 Maintenance

4.2 Product Management

4.2.1 Configuration Management

4.2.1.1 Product Element Identification

4.2.1.2 Configuration Change Management

4.2.1.3 Element Management

4.2.1.4 Configuration Status Management

4.2.2 Data Management

4.2.2.1 Data Element Identification

4.2.2.2 Data Element Change Management

4.2.2.3 Data Element Management

4.2.2.4 Data Status Management

Course Guidelines: Data to be managed should include project effort, defect tracking, COCOMO II milestone runs, and any client-critical data.

4.2.2 Product Integration

4.2.2.1 Integration Strategy Identification

4.2.2.2 Integration Change Management

4.2.2.3 Integration Element Management

4.2.3 Integration Status Management

Course Guidelines: The integration strategy should be included in the Overall Life Cycle Strategy and Phase Milestones and Schedules in LCP 2.1 and 2.2. A single set of integration and test plans should cover Product Integration, Verification, and Validation.

4.3 Properties Management

4.3.1 Quality Assurance

4.3.2 Verification and Validation

4.3.2.1 Inspections

4.3.2.2 Testing

4.3.3 Measurement and Analysis

4.4 Success Management

4.4.1 Shared Vision:

4.4.2 Stakeholder WinWin

4.4.2.1 Transition

4.4.2.2 Support

4.4.2.3 Release Description

4.4.2.4 Users Manual

4.4.3 Benefits Realization: Results Chain

[refer to **EC-04** MBASE, CMMI and Benefits Realization that is available at http://sunset.usc.edu/classes/cs577b_2000/EC/04/EC-04.pdf].

5 Resources: Use COCOMOII for cost estimates.

[refer to **EC-05, EP-08, EP-9** that is available

http://sunset.usc.edu/classes/cs577a_99/EC.html#EC-05,

http://sunset.usc.edu/classes/cs577a_99/EP.html#EP-08,

http://sunset.usc.edu/classes/cs577a_99/EP.html#EP-09]. To install the tool refer to

http://sunset.usc.edu/classes/cs577a_99/

5.1 **Work Breakdown Structure:** Based on architecture. Modification of the base MBASE WBS.

Refer to [Royce, 1998] Section 10.1 for the WBS guidelines.

5.2 **Budgets:** Costs of development, costs of tools and COTS, costs of training and transition, costs of support. Provide a manual COCOMO II analysis.

6 Appendices

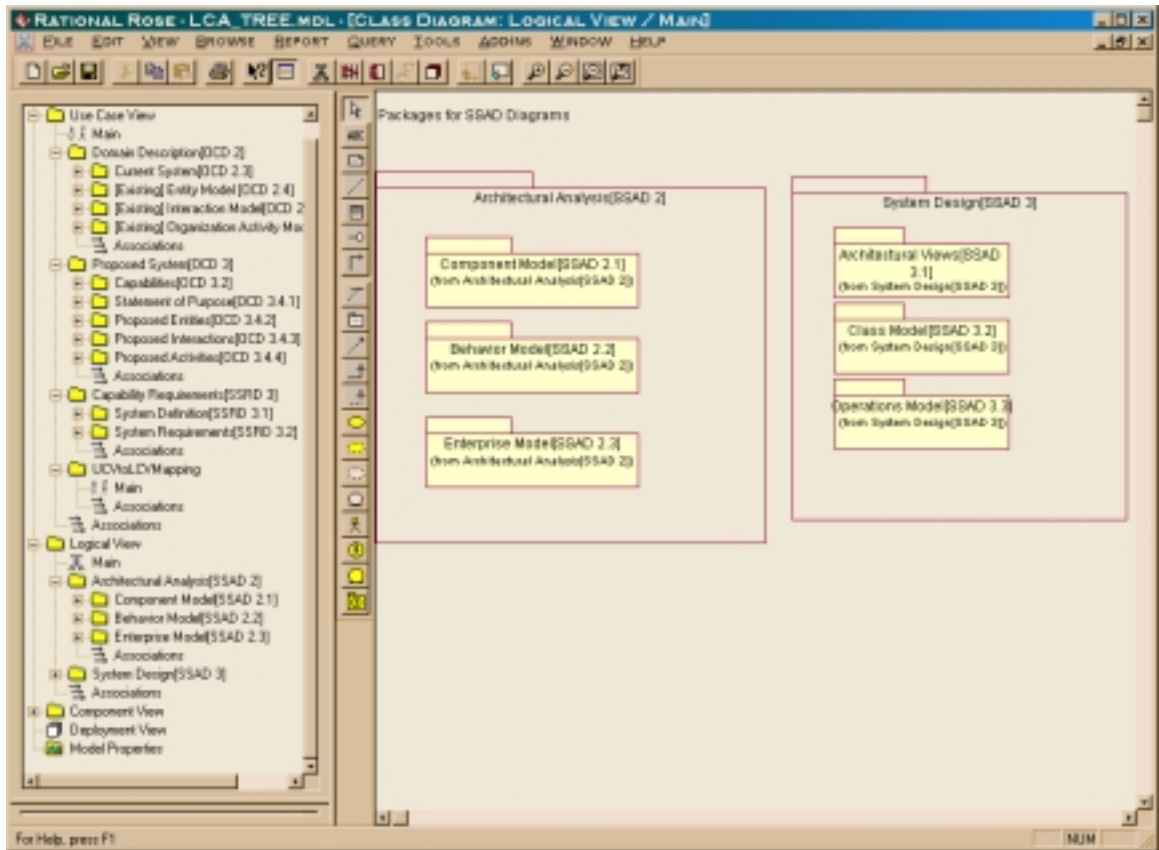
A. COCOMO Results

B. Gantt Charts

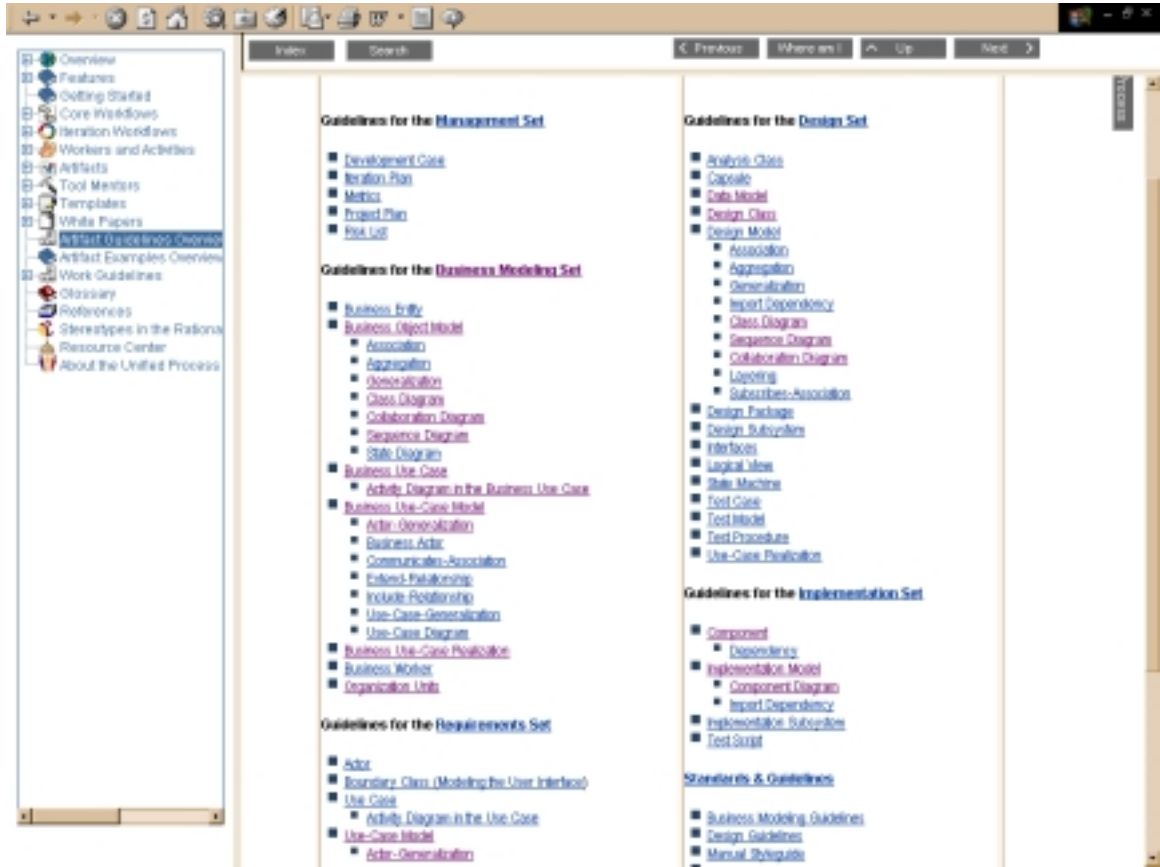
6) Feasibility Rationale Description (FRD)

- 1 **Introduction**
 - 1.1 **Purpose of the Feasibility Rationale Document**
 - 1.2 **References**
- 2 **Product Rationale**
 - 2.1 **Business Case Analysis:** All costs should have an associated \$ value.
 - 2.1.1 **Development Cost Analysis:** Based on LCP budget for schedule and time.
 - 2.1.2 **Transition Cost Estimate:** Cost of data entry, new equipment, COTS licenses.
 - 2.1.3 **Operational Cost Estimate:** Cost of administrator, operators.
 - 2.1.4 **Maintenance Cost Estimate:** Cost of upgrading system, fixing bugs, providing new features.
 - 2.1.5 **Estimate of Value Added and Return on Investment:** Benefits analysis and comparison with total costs.
 - 2.2 **Requirements Satisfaction**
 - 2.2.1 **Operational Concept Satisfaction:** Activities in operational concepts are supported by the architecture.
 - 2.2.2 **Project Requirements Satisfaction:** Plan and architecture consistent with SSRD 2
 - 2.2.3 **Capability Requirements Satisfaction:** How SSAD covers all system requirements of the SSRD.
 - 2.2.4 **Interface Requirements Satisfaction:** SIRSI, BRS, etc are linked to the system?
 - 2.2.5 **Level of Service Requirements Satisfaction:** How does the SSAD address the levels of service requirement. How close to the desirable level can the architecture get?
 - 2.2.6 **Evolution Requirements Satisfaction:** How does the SSAD allow future satisfaction of the evolution requirements? How does the LCP plan for future expansion?.
 - 2.3 **Stakeholder Concurrence**
- 3 **Process Rationale**
 - 3.1 **System Priorities:** Identifying features in the increments, at least two priority levels, grouping high priority requirements into the first increment. How is the construction approach parallel?
 - 3.2 **Process Match to System Priorities:** Increments consistent with LCP and Project requirements.
 - 3.3 **Consistency of Priorities, Process and Resources:** Staffing, training and tools/facilities related to the increments.
4. **Project Risk Assessment:** Project specific life cycle risks both technical and social, complete description of risks, risk exposure, risk reduction leverage, contingency plan.
5. **Analysis Results**
 - 5.1 **Product Features**
 - 5.1.1 **Advantages:** Shortfall satisfaction
 - 5.1.2 **Limitations**What does the SSAD not allow, or makes difficult? How does this not severely affect the stakeholders.
 - 5.1.3 **Tradeoffs Considered:** Architectural. Operational concept and prioritization tradeoffs made. Cost of development tradeoffs made
 - 5.1.4 **Changes Not Considered****Commercial-Off-The-Shelf Solutions:** List of COTS being investigated or chosen. Why those chosen are the most appropriate?
6. **Appendices**
 - A. Cash Flow Statement

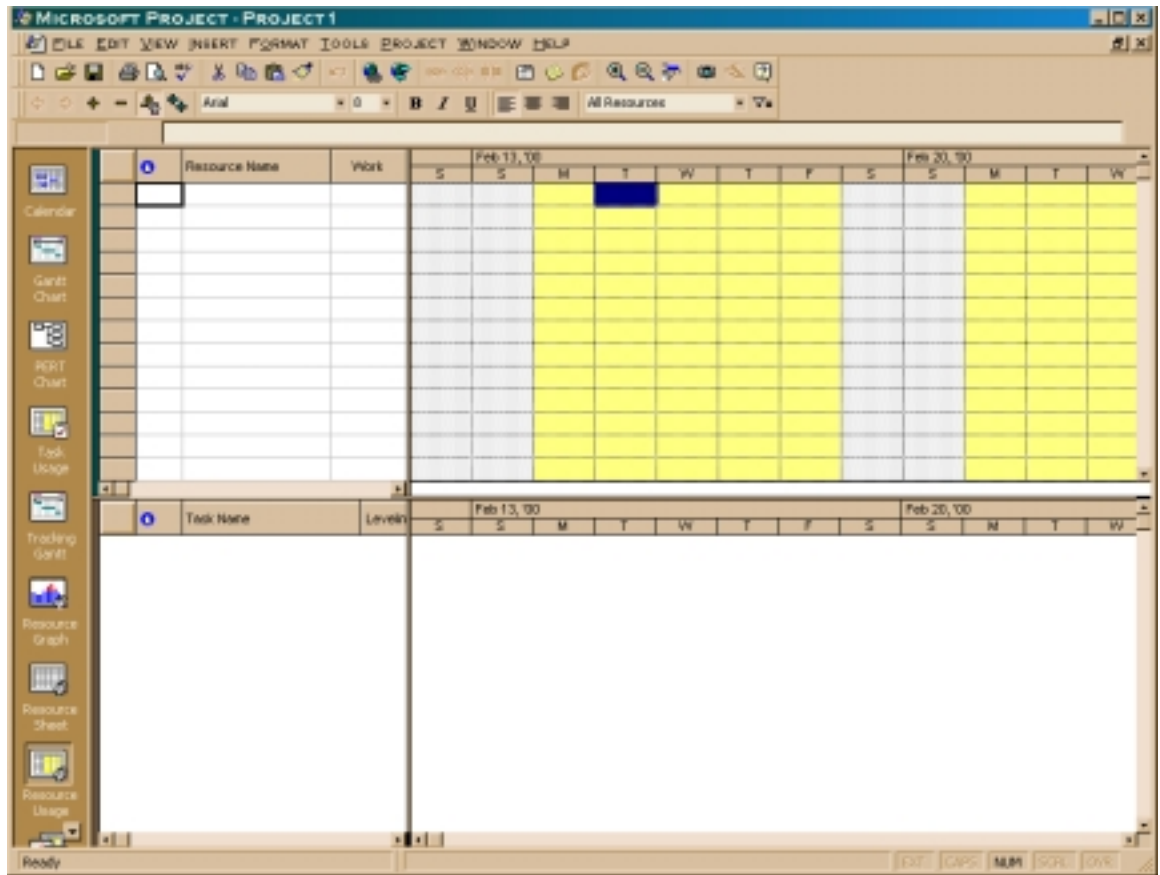
B) LCA-TREE, SSAD diagrams
 [refer to LCA-TREE.MDL]



C) RUP CD, Right & Left Frame



D) Microsoft Project CD, Left Side



E) USC-COCOMO II.2000.0

Project Name: Scale Factor: Schedule:

Development Model:

X	Module Name	Module Size	LABOR Rate (\$/month)	ERF	Language	NRE BEV	EST BEV	PROB	COST	INST COST	Staff	RISK

Total Lines of Code:

	Estimated	Effort	Sched	PROB	COST	INST	Staff	RISK
Optimistic	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0
Most Likely	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0
Pessimistic	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0

READY