

Rational Rose® Enterprise Edition
Version 2003

A Guide for Evaluation and Review

Table of Contents

Introduction	3
Application Modeling Background.....	3
Visual Modeling.....	3
Importance of Architecture	4
Visual Modeling the Wrong Way	4
Unified Modeling Language	5
Rational Rose Enterprise Edition Version 2003.....	6
Additional Functionality	6
Application Modeling Competitive Landscape	6
Pricing, Packaging, Platforms.....	8
Feature Matrix.....	9
System Requirements.....	12
Training and Support.....	12
About Rational Software Corporation.....	13
Revenues	14
Products.....	14
Worldwide Organization.....	15
Resources.....	15
Recommended Reading.....	16
Reviewer Contacts	16
Appendix A – Evaluation Criteria.....	17

Introduction

This guide is intended for reviewers and evaluators of Rational Rose Enterprise Edition Version 2003. It is designed to provide an overview of the product and Rational Software Corporation. You can use this guide as a resource to facilitate your evaluation of Rational Rose Version 2003.

Your Rational Rose Reviewer's Guide is organized into the following sections:

- 1) Introduction
- 2) Application Modeling Background—brief overview of UML modeling
- 3) Rational Rose Enterprise Edition—overview
- 4) Application Modeling Competitive Landscape—overview of the competitive market and criteria for evaluating visual modeling products
- 5) Pricing, Packaging, Platforms
- 6) Feature Matrix
- 7) System Requirements
- 8) Training and support—description of training and support services
- 9) Corporate background—overview of Rational Software Corp.
- 10) Resources—list of additional informational resources and Web links
- 11) Reviewer Contacts—list of phone and fax numbers, addresses, and email addresses

Please see the Rational Rose RealTime Reviews guide for Rational Rose RealTime, our modeling tool for the real time and embedded markets.

Application Modeling Background

Visual Modeling

Visual modeling provides a way to understand complex problems and communicate that understanding to others. The resulting models capture the essentials of complex problems by eliminating nonessential detail that only serves to confuse the issue. This process of capturing essentials while filtering out the nonessential detail is referred to as abstraction. Abstraction allows people to focus on the big picture—how the components of the problem relate and interact—without getting bogged down in the specific details of any component or technical implementation.

Visual modeling enables software developers to analyze complex software problems at a highly abstracted level using a set of well-defined graphical icons. Software developers use visual modeling to create different diagrammatic views of the system they are building and gradually add detail to the models, which enables the models to evolve into

actual software implementations. The progression from modeling to implementation is an iterative process involving verification and testing at many points along the way.

Visual modeling and abstraction are critical in the development of modern, distributed, component-based applications. There simply is too much happening in even seemingly simple applications for developers to jump right in and begin programming. The days when one person could immediately grasp a software solution in its entirety and knock out the implementation in a burst of inspiration are long gone (if those days ever truly existed). At the very least, developers need to communicate what they are doing so others can maintain and enhance the application after they themselves have moved on to something else.

Importance of Architecture

Architecture is the set of significant decisions about the organization of a software system. The right UML models and diagrams will illuminate and offer insight into serious software development problems. The wrong models and diagrams, while they may provide a façade or meet documentation requirements, will mislead and cause teams to focus on irrelevant issues. A valuable architecture is rarely visualized from auto-generated diagrams.

Because architecture is important it is usually one of the fundamental reason for creating and maintaining models in UML. Therefore, tools that support UML should have the ability to facilitate architecture-first modeling and enforce conceptual control. An example is to be able to identify when low-level design details violate architectural policies or to have support for easily navigating where key classes are used in a model. Likewise, the ability to diagnose which classes and operations are needed to implement high-level requirements, such as use cases, can provide valuable feedback when assessing architectural impacts derived from project changes to scope, cost, or time.

Visual Modeling the Wrong Way

Unfortunately, there are several tools and processes in the market today that distract developers from producing truly effective software that is modifiable, maintainable, reliable, and scalable. These tools have developers focus solely on easily obtainable features to drive their development process. Features are important, but must be implemented with respect to the system architecture. Features must also be derived from the user requirements of the system, which are specified in the use cases of the model.

The goal of visual modeling is to communicate software architecture and design by reducing the clutter of details and focus on the key decisions. However, tools that spend the majority of their time (and your system time) updating every minor modification to source code do not leverage the concept of abstraction or truly help teams communicate effectively.

If the artifacts of your software development tools and processes are merely the product of simple code parsing, new developers will spend a lot of time learning the system before effective maintenance or enhancements can take place. Or worse, they won't. Simply said, visual modeling the wrong way can be worse than no modeling at all because there will be no pretense of communication, architecture, or abstraction – only source code.

Unified Modeling Language

There have been many approaches to visual modeling, such as OMT, Booch, and OOSE. Each offers recognized strengths and drawbacks, and each has its adherents. While they all support the concept of visual modeling and abstraction, each uses a different method and set of notation, the graphical icons used to identify components, relationships, and interactions. The confusion resulting from the competing visual modeling approaches led to a period of religious modeling methodology wars in which adherents promoted their favorite approach. The obvious solution was a standard approach that incorporates the best of the individual approaches.

Unified Modeling Language (UML) emerged as the solution to end the modeling methodology wars. UML addresses data modeling (entity/relationship diagrams), business modeling (workflow), object modeling, and component modeling. It can be effectively used to visualize, specify, construct, and document the elements of a software application. Used with all processes throughout the project lifecycle (requirements capture, analysis and design, implementation, testing), UML provides a standard language that may be understood by everyone involved with the project. UML was officially introduced in June 1996. Version 1.0 was released in January 1997. In November 1997, the Object Management Group (OMG) approved Version 1.1 as the standard language for analysis and design. The OMG now owns the UML and UML version 1.3 is the most current version. UML version 1.4 was approved Feb. 2, 2001, but has not been released yet.

Rational Rose Enterprise Edition Version 2003

Rational Rose Enterprise Edition version 2003 is the latest release of Rational Software Corporation's market-leading visual modeling tool. It supports eight UML diagram types – activity, use case, sequence, collaboration, class, state, component, and deployment - as defined in the UML 1.3 specification. And with the various add-ins and capabilities, Rose offers solutions for all modeling needs: business, application, data, and web modeling. Everyone involved in the project from business analysts to data analysts to application analysts and developers can use one tool and one language.

Rational Rose Enterprise Edition supports multiple languages (Java, J2EE, ANSI C++, Visual C++, Visual Basic, CORBA IDL, MIDL and XML) within the same model. It also supports team development, even where team members may be geographically dispersed. Finally, it provides a range of usability enhancements and add-ins that extend the functionality of Rose.

In 1999 Rational introduced suites of tools that were optimized for teams of people. Various editions of Rational Rose are in different Rational Suites. Rational Rose Professional Data Modeler is part of the Rational Suite AnalystStudio. Rational Rose Enterprise is found in Rational Suite DevelopmentStudio and Rational Suite Enterprise.

Additional Functionality

Rational Rose Add-Ins, integrated companion products developed by Rational and Rational's RoseLink Partners, were introduced in Rational Rose 98. Rational Rose features a robust family of Add-Ins, which extend the functionality of the tool. Over 90 Add-In products can be used with Rational Rose today. Add-Ins may be quickly and easily installed through the graphical Add-In Manager. Complete partner information may be found at <http://www.rational.com/partners/index.jttml>.

Application Modeling Competitive Landscape

Rational Rose is the long-established market leader in the visual modeling tool segment. It has achieved and continues to maintain its leadership position by providing a more flexible, easier to use visual modeling tool than its competitors.

When evaluating visual modeling tools, developers repeatedly cite the following criteria:

- Ease of use—intuitive to use, even for developers who are new to application modeling methods and practices.
- Flexibility—support for the kind of complex systems that developers today must model, including systems using multiple components and languages.

- Ease of integration into the overall application lifecycle—able to participate in iterative development through bi-directional engineering.
- Scalability—support for large, complex projects and large, often dispersed teams of developers.
- Standards-based—support for UML.

Rational Rose exceeds each of these criteria. Its adherence to the GUI conventions of the Microsoft Visual Studio family of tools makes it familiar to the majority of developers. It supports multiple components and languages within one model. Round-trip engineering ensures that Rose models are integral parts of iterative development. The team management features allow Rose to efficiently handle large projects with multiple developers. Rose is the first tool on the market to provide UML based data modeling and web modeling capabilities. In addition, Rose offers many advantages in other areas, such as customizability and productivity enhancements.

Pricing, Packaging, Platforms

Rational Rose is available in three editions. These are detailed in the following table:

Rose v2003	Distinguishing Features	Price – Node Locked	Price - Floating
Enterprise Edition	UML-compliant modeling Multi-language support (Java, J2EE, ANSI C++, Visual C++, Visual Basic, IDL, XML) Test code generation Data modeling Web modeling QualityArchitect All features and functionality	\$3495	\$6115
Professional edition	UML-compliant modeling One language Available in the following packaging: <ul style="list-style-type: none"> • Rose Professional Edition for C++ • Rose Professional Edition for Visual Basic • Rose Professional J Edition • Rose Professional Edition for Ada • Rose Professional Data Modeler Edition 	\$1995 (each edition)	\$3490 (each edition)
Rose Modeler	UML-compliant modeling No language support	\$1495	\$2615

All prices are given in U.S. dollars and do not include support.

Feature Matrix

The features of the various editions of the Rose product family are summarized in the table below:

Feature	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
Activity Diagrams	X	X	X
Use Case Diagrams	X	X	X
Sequence Diagrams	X	X	X
Collaboration Diagrams	X	X	X
Class Diagrams	X	X	X
State Diagrams	X	X	X
Component Diagrams	X	X	X
Deployment Diagrams	X	X	X
Structure Diagrams			
Data Model Diagrams	X	Data Modeler Edition	
Data modeling	X	Data Modeler Edition	
Workspace Support	X		X
Multi-user Support	X	X	X
Model Integrator	X	X	X
Web Publisher	X	X	X
Auto Synchronize source and model file	Java and ANSI C++ languages	Java and ANSI C++ Edition	
ClearCase Integration	X	X	X
Apex Integration	X	Ada Edition	
Visual SourceSafe Integration	X	X	X
Partner Add-Ins Support	X	X	X
Rose Extensibility Interface	X	X	X
Distributed Rose RealTime Designs			
Host UML model execution and debugging			
Target UML model execution and debugging			
Rose RealTime Extensibility Interface			
Basic Report Generation	X	X	X
Code templates	X	VB and VC++	

Feature	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
		languages only	
Framework Wizard	X	X	X
Full declarative and structural definition (algorithmic) code generation for C and C++			
Behavioral modeling using state diagrams with full code generation for C and C++			
ANSI C++ code generation and reverse engineering	X	C++ Edition	
ANSI C code generation			
Visual C++ code generation	X	C++ Edition	
Visual C++ reverse engineering	X	C++ Edition	
Visual Basic code generation and reverse engineering	X	VB Edition	
Reverse engineering of COM components	X	VB and C++ Editions	
Typelib reverse engineering	X	VB and C++ Editions	
Ada code generation and reverse engineering	X	Ada Edition	
Java code generation	X	Java Edition	
Java reverse engineering	X	Java Edition	
J2EE component generation and reverse engineering	X	Java Edition	
Gang-Of-Four patterns	X	Java Edition	
IBM VisualAge for Java IDE integration	X	Java Edition	
Webgain VisualCafe IDE integration	X	Java Edition	
Sun Forte for Java IDE integration	X	Java Edition	
Borland JBuilder IDE integration	X	Java Edition	
Corba/IDL generation		C++ and Ada	

Feature	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
and reverse engineering	X	Editions	
MIDL generation and reverse engineering	X		
XML DTD generation and reverse engineering	X		
Web modeling ASP and JSP pages (round trip engineering)	X		
Forward engineering from object model to data model to database or to DDL script	X	Data Modeler Edition	
Reverse engineering from database or DDL script to the data model then to object model	X	Data Modeler Edition	
Maintain object to relational table mapping and changes	X	Data Modeler Edition	
Views and tablespace modeling support	X	Data Modeler Edition	
Data storage modeling	X	Data Modeler Edition	
Webgain TOPLink wizard	X	Data Modeler Edition	
Common Licensing	X	X	X
Rational QualityArchitect	X		

System Requirements

Item:	Requirement:
Processor	A Pentium-based 150 MHz PC-compatible computer system
Memory	128 MB of RAM (128 MB recommended)
Disk space	400MB (200 MB minimum swap space recommended)
Other	An SVGA-compatible display (256 or more colors recommended) with resolution set at 800 X 600 pixels Any pointing device with at least two buttons

Rose Version 2003 is supported on the following operating system platforms:

- Windows XP Professional
- Microsoft Windows NT 4.0 Service Pack 6
- Microsoft Windows 98
- Microsoft Windows 98 SE
- Microsoft Windows 2000
- Microsoft Windows ME

Rose Version 2003 requires DCOM under Windows 95. If you do not have DCOM on your Windows 95 system, you will be given the opportunity to install it from the installation CD.

The VC++ and VB Add-Ins require Visual Studio 6.0.

The Rose J add-in requires JDK 1.1.6 or another Java IDE.

Reverse Engineering Oracle and DB2 databases using Rose Data Modeler requires installation of an RDMS client.

Training and Support

Rational provides a full range of support:

- Annual maintenance and support contracts
- Online technical support (<http://www.rational.com/support/index.jsp>)
- Telephone support (800-433-5444, 408-863-5000)

Rational's telephone support staff is available to answer customer questions from 9:00 A.M. to 6:00 P.M. Pacific time, Monday through Friday.

Rational University—speeds learning by combining a proven software development process and practice with formal education and training. In addition to expert product and professional education programs, Rational University courses provide a conceptual framework that will change the student's approach to software development. More information on Rational University is available at <http://www.rational.com/university/index.jsp>.

About Rational Software Corporation

Rational Software provides a software development platform that improves the speed, quality, and predictability of software projects. This integrated, full life-cycle solution combines software engineering best practices, market-leading tools, and professional services. Ninety-six of the Fortune 100 relies on Rational tools and services to build better software, faster. This open platform is extended by more than 500 partners who provide complementary products and services.

Founded in 1981, Rational is one of the world's largest software companies, with revenues of \$815 million in the fiscal year ended March 2001, and over 3,500 employees worldwide. Rational is a component of the Nasdaq-100 Index®.

Additional information is available on the Internet at www.rational.com.

The problem: "chaos" in software development

Wasted time, unproductive meetings, burdensome status reports, tedious rework. These are the persistent problems of software development – obstacles that waste your time, sap your creativity, and undermine the morale of your team.

They don't have to. Equipped with the right tools and the right software development approach, you can transform development confusion into order and clarity. The Rational platform for software development liberates your team from obstacles that can hinder creativity and delay results.

The solution: A more Rational approach

How can you rise above the thousand-and-one barriers that stand between you and project success? With the Rational platform for software development – the industry's only life-cycle foundation for designing, developing, testing, and deploying software. This unique platform gives you the freedom to work at the level of abstraction that's right for each task. Choose the development environment and IDE that's right for your team. And tailor your process and project environment to meet your specific needs. No matter what your job title or development environment, the Rational solution that can improve the quality of your code and your work life.

The Rational platform features seamless integration among the best practices you follow, the tools you use, and the services you call upon to accelerate success. Everyone benefits from this solution – development team members can focus more easily, project leaders can free their teams from unproductive activities, and managers can execute more easily against business goals. The result: high-quality software delivered at the speed of business.

Revenues

Founded in 1981, Rational is one of the world's largest software companies, with revenues of \$815 million in the fiscal year ended March 2001, and over 3,500 employees worldwide. Rational is a component of the Nasdaq-100 Index®.

Products

Rational provides market-leading products for each area of the software development lifecycle. These include [Rational Suite](#) for unifying software teams, with five editions to provide a complete solution for specific roles-- [Rational Suite AnalystStudio](#), [Rational Suite DevelopmentStudio](#), [Rational Suite ContentStudio](#), [Rational Suite TestStudio](#), and [Rational Suite Enterprise](#); [Rational Requisite® Pro](#) for requirements management; the [Rational Rose®](#) family for visual modeling; [Rational Rose RealTime](#) for modeling real-time embedded software development; [Rational TeamTest](#) for client/server functional testing; [Rational SiteLoad](#) for automated Web site performance testing, [Rational TestManager](#) for open, extensible test management, [Rational Purify®](#) for run-time error checking; [Rational Quantify®](#) for the detection of software performance bottlenecks; [Rational PureCoverage®](#) to verify the completeness of software testing activities; the [Rational Apex®](#) family of integrated C/C++ and Ada programming environments; [Rational ClearCase®](#) for software configuration management; [Rational ClearQuest™](#) and [Rational ClearDDTS™](#) for change request management and defect tracking; [Rational SoDA®](#) for automatic generation of software documentation; and the [Rational Unified Process™](#), a configurable software process framework adaptable to any project.

Rational's products support the major implementation technologies used on Windows NT, Windows 2000, UNIX, and Linux, including Java, J2EE, ANSI C++, Visual Basic, Visual C++, PowerBuilder, Ada, Fortè, PeopleTools, Oracle Developer/2000, ActiveX, COM, WinDNA, ASP, and CORBA. Rational's products can be purchased and used individually or as part of integrated suites. Rational's suites provide powerful out-of-the-box integration among multiple products, leveraging the power of each individual product.

Rational Rose provides the leading implementation of the **Unified Modeling Language (UML)**, the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. The UML simplifies the complex process of software design, enabling developers to create a "blueprint" for construction. Rational developed the UML with contributions from other industry leaders, including Microsoft, IBM, and Oracle. A copy of the UML specification can be obtained at <http://www.rational.com/uml/>.

Worldwide Organization

Rational Software has more than 3500 employees worldwide. Corporate headquarters are located in Cupertino, California. Major development centers are located in California, Massachusetts, Oregon, Wisconsin, Colorado, North Carolina, Washington, Pennsylvania, Sweden, Ottawa and India. Regional sales and support offices are located throughout the United States, Canada, the United Kingdom, France, Sweden, Germany, The Netherlands, Switzerland, Italy, Japan, Singapore, Hong Kong, Australia, New Zealand, Taiwan, Korea, India, Brazil, and Mexico. Partners and distributors provide local sales and support elsewhere in the world.

Resources

Rational Software Corporation

<http://www.rational.com/>

Getting Started:

<http://www.rational.com/products/rose/gstart/gstart.jsp>

Product Support:

<http://www.rational.com/support>

Information on Rose Partner products:

<http://www.rational.com/partners>

UML Resource Center:

<http://www.rational.com/uml>

Recommended Reading

1. Visual Modeling with Rational Rose 2000 and UML, Terry Quatrani, Addison Wesley, ISBN 0-201-69961-3
<http://www.amazon.com/exec/obidos/ASIN/0201699613/o/qid=954952883/sr=8-1/103-6716550-0235812>
2. The Unified Modeling Language User Guide, Grady Booch, Jim Rumbaugh, Ivar Jacobson, Addison-Wesley, ISBN 0-201-57168-4
<http://www.amazon.com/exec/obidos/ASIN/0201571684/o/qid=954953058/sr=8-2/103-6716550-0235812>
3. The Unified Modeling Language Reference Manual, Jim Rumbaugh, Grady Booch, Ivar Jacobson, Addison Wesley, ISBN 0-201-30998-X
<http://www.amazon.com/exec/obidos/ASIN/0201571684/o/qid=954953145/sr=8-1/103-6716550-0235812>
4. Building Web Applications with UML, Jim Conallen, Addison Wesley, ISBN 0-201-61577-0
<http://www.amazon.com/exec/obidos/ASIN/0201615770/o/qid=954953193/sr=8-1/103-6716550-0235812>

Reviewer Contacts

Bill Durling
Public Relations
Rational Software, the e-development company
781-676-2489
bdurling@rational.com

Note: the above contact information is provided for the use of the reviewer and is not intended for publication.

Rational Software Corp.
18880 Homestead Rd.
Cupertino, CA 95014
800-728-1212
408-863-9900
Fax: 408-863-4120
Email: rosebud@rational.com
Web: www.rational.com

Appendix A – Evaluation Criteria

Installation and setup	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
Complete, easy-to-follow installation instructions	X	X	X
Single licensing mechanism for all Rational products	X	X	X
Floating and node-locked licenses available	X	X	X
OS and system requirements clearly stated	X	X	X
Various installation media available (CD default)	X	X	X
Uninstall option	X	X	X

Documentation and Tutorials	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
Installation Guide	X	X	X
Release Notes	X	X	X
Licensing Guide	X	X	X
User's Guide	X	X	X
Documents in pdf format on Documentation CD	X	X	X
Complete on-line product tour and tutorials	X	X	X
Sample models available	X	X	X

Help and Support	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
On-line help in Winhelp format	X	X	X
All on-line help searchable	X	X	X
Complex help searches available (wildcard, and, or, multiple keyword)	X	X	X
On-line help printable	X	X	X
Help for help	X	X	X
Toll-free support during business hours ¹	X	X	X
E-mail support during business hours	X	X	X
24x7 support available ²	X	X	X
Customer email user's group and contribution area	X	X	

¹ North America only

²

Tool Integrations and API	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
Requirements			
Rational RequisitePro	X	X	X
Configuration Management Version Control Tools			
Rational ClearCase	X	X	X
Rational ClearCase LT	X	X	X
Documentation			
Rational SoDA (Software Documentation Automation)	X	X	X
Software Development Process			
Rational Unified Process	X	X	X
Development Tool Suites			
Rational Suite DevelopmentStudio	X		
Rational Suite AnalystStudio		Data Modeler	
Rational Suite Enterprise	X		
API			
100% open API accessible via scripting	X	X	X

Language support	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
ANSI C++	X	C++ Edition	
Visual C++	X	C++ Edition	
Visual Basic	X	VB Edition	
Java	X	J Edition	
Ada	X	Ada Edition	
Data modeling	X	Data Modeler Edition	
Web modeling	X		
CORBA IDL	X		
MIDL	X		
XML DTD	X		

Modeling	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
Use Case Diagrams	X	X	X
Activity Diagrams	X	X	X

Modeling	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
Sequence Diagrams	X	X	X
Collaboration Diagrams	X	X	X
Class Diagrams	X	X	X
State Diagrams	X	X	X
Component Diagrams	X	X	X
Deployment Diagrams	X	X	X
Data Model Diagram	X	Data Modeler Edition	
User-definable color for all modeling elements	X	X	X
Supports the model-view-controller paradigm	X	X	X

Tool Usage	Rose 2003 Enterprise	Rose 2003 Professional	Rose 2003 Modeler
Intuitive, customizable GUI	X	X	X
Model Explorer-like browser	X	X	X
Parallel development support	X	X	X
Distributed development support	X	X	X
Multi-user support	X	X	X
Model and code synchronization	X	X	
Graphical visual differencing and merge utility	X	X	X
Support for RoseLink Partner add-ins	X	X	X
Rose Extensibility Interface (REI) for scripting	X	X	X
VBScript environment, including debugging	X	X	X
Basic textual report generation	X	X	X
WYSIWYG diagram printing	X	X	X
HTML web model publishing	X	X	X
Show usage report	X	X	X
Show instances report	X	X	X
Show access violations report	X	X	X
Show participants in use case	X	X	X
Documentation report	X	X	X
Rose98i, Rose2000, Rose 2001 model importing	X	X	X
Integrated documentation and URL attachment via drag and drop (docs, spreadsheets, pictures, etc..)	X	X	X
Diagram linking	X	X	X
Generate test scripts from UML model	X		