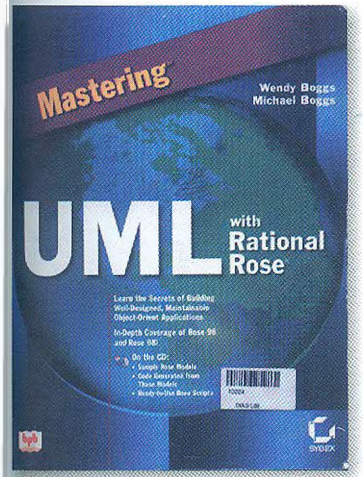

MASTERING UML WITH RATIONAL ROSE

By Wendy Boggs, Michael Boggs
BPBPublication

Pages: 957
Price: Rs.450

Reviewed by:
Ms. Sonia Gupta, Faculty, DIAS

Object oriented is the new buzz-word for the software industry. Organizations are switching from the traditional approach of software development to the object oriented paradigm. Object oriented modeling is a completely different way of viewing the applications. Using this technique we can divide the larger applications into various objects which are altogether independent of each other and also provide the ability to build them once and use them over and over again. With this kind of system development the software development process has become more organized and manageable because it encourages software developers to work and think in terms of the application throughout the software life cycle. Object oriented methodology plays an important role in the development of such systems. The complete process consists of various stages starting with system conception which is the formulation of tentative requirements. The analysis restates the requirements from the previous stage by constructing models. System design allows the team of developers to devise a high level strategy for providing the solution to the above stated requirements. In the next stage, the class designer adds more details to the models generated depending upon the system design. Finally, the implementation stage the system design and the class design is actually converted into a particular programming language database or hardware. Object oriented concepts are applied throughout the system development process. To promote better understanding of requirements, cleaner designs and more maintainable systems, development following object oriented principles and methodology Rational unified process (RUP) was introduced. It is an interactive software development process framework under the software called Rational Rose which uses UML i.e. Unified Modeling Language. Rose is designed to aid in the modeling and development of the software and even helps in planning the deployment of software. Rose has changed the way the softwares developed by making the developers view the entire software into the forms of models. Each of the model is independent of another making the complete software more manageable. Further, it uses UML as its modeling language which gives the users more independence in terms of well defined models and user defined stereotypes. Also, reverse engineering in Rose makes it more useful which helps to backtrack the complete software. The book "Mastering UML with Rational Rose" talks about the software development using the object modeling. Rational Rose is an object oriented UML software design tool intended for visual modeling and component construction of enterprise level software application. In the field of software engineering, the UML is a standardized specification language for object modeling that includes a graphical notation used to create an abstract model of a system called as UML model.

Further, it discusses about the two popular features of Rational Rose i.e., ability to provide interactive development and round trip engineering. Iterative development allows designers to take advantage of evolutionary development because the new applications can be created in stages with the output of one iteration becoming the input to the next. Round trip engineering means going back and updating the rest of the model to ensure the code remains consistent.

There are small segments of narratives and exercises at the end of most chapters to provide guidance. The book is divided into four sections. The text is indeed a helping hand to the readers for modeling an order entry system. Moreover, it can be used as a reference for specific Rose or UML questions.

Section 1 deals with the fundamentals of UML, the object modeling process and the Rational Rose tool. It thoroughly defines the concepts of polymorphism, inheritance, reusability and encapsulation. It discusses how visual modeling and Rational Rose are useful at several stages of the software development process in simple language and lucid style. It makes the readers understand in a better way that how by producing the visual models of a system one can show how the system works on several levels and how one can model the interactions between the users and a system, between objects within a system and between systems. The users who are new to the Rose Software can also learn to install the product and various aspects of Rose environment, including different parts of the screen and menus, exporting and importing models, different views in a rose model, documentation window and saving the models.

The building blocks of UML popularly known as UML diagrams are explained in Section 2. It includes how to work with use cases, actions and use case diagrams as the requirements of the system to be built are the set of all use cases and actors. The most versatile concept in UML i.e. interaction diagrams is also discussed. Object Interaction diagrams show how objects work together in order to implement the functionality of a use case. Further, the types of interaction diagrams – sequence diagrams which show the flow of information through time and collaboration diagram which illustrates the relationships between objects and show messages between object are explained. This section provides a nice clarification of diagrams as structure and behavior types. Class diagrams represents the static behavior and the system through attributes, functions and packages, relationship names, role names, qualifiers and multiplicity. To understand the dynamic behavior of system it is explained how to generate state transition diagram for a class which shows its various stages and transitions. The physical structure of the system is given with the help of component and deployment diagrams. Basically, the chapters under this section allows a novice user to understand that which diagram can represent a particular stage in the software development process. It presents the use of each diagram in a very systematic and concise manner which allows the readers to learn them step by step.

C++, Java, visual Basic, Power Builder and Oracle8 code generation capabilities of Rose are discussed in section 3. Each chapter under this section explains the step by step procedure of generating code in a particular programming language. One also examines the Interface Definition Language and the data definition language generation from Rose mode including Oracle8 schema. Basically the important feature of Rose to provide working of a large number of users on variety of platforms is highlighted.

The last section discusses the reverse engineering features of Rose. It explains the complete process of taking the information from source code and creating or updating a Rose model. Rose basically read the components, packages, classes, relationships, attributes and operations from the code. Once this information is in a Rose model, one can make any needed changes and then regenerate the code through the forward engineering. This makes it possible for the team of developers to backtrack the entire software process. The entire set of chapters explains the procedure of reverse engineering involving various platforms as C++, Java, 7 Power Builder, Visual Basic and Oracle 8. In addition, Rose will capture information about the variables, functions and relationships in the code. It explains the features in very simple language and represents them in graphical view which increases the knowledge and provides more readability.

Several end of chapter exercises help the readers to respond to real world information or problems. One can analyze on their own that what exactly they have gained so far.

Ideally a textbook should take into account the level of understanding of the students. So, this book is good for beginners and intermediate users of Rational Rose and UML. The reader will learn how to apply object oriented concepts through visual modeling to all stages of the software development life cycle. It does not assume any prior knowledge of object oriented concepts. The book is beneficial for learning how to design the software systematically and make it more readable through various models. It provides the complete set of procedures for multiple set of users working on various platforms. Prerequisites for readers of book include exposure to some programming languages and a knowledge of software engineering concepts as life cycle of software, pros and cons of various procedural languages and models in system development.

The text is presented in plain language and it is accompanied by steps and graphic views. It makes the readers to develop more interest while reading and also learn it quickly. This book is good for all types of users which is highly appreciable.