Comparative analyses for the performance of Rational Rose and Visio in software engineering teaching

Zhaojun Yu¹, Zhan Xiong², a

¹ Computer Sciences College, Wuhan Donghu University, Wuhan 430212, China
² School of Electrical Engineering and Telecommunication, Faculty of Engineering, University of New South Wales, Sydney, Australia

Abstract. Modelling is one of the most difficult tasks of software engineering. UML (unified modelling language) is an object-oriented standard modelling language, which can express both the dynamic and static information during the processes of software analysis and design. Among the over 100 UML modelling tools, Rational Rose and Visio are the most extensively used ones. Here we comprehensively compare the performance of these two tools in software engineering teaching and discuss their advantages and disadvantages. The points presented in this review provide guidance for the selection of modelling tools and arrangement of teaching activities for software engineering.

1. Introduction

Software engineering is one of the core professional courses for the majors including computer science and technology and software engineering, which deals with the basic principles, developing methods, modeling tools and project managing of software engineering. It is very important for improving the abilities of students in developing software. The main thread for teaching the software engineering course is usually arranged according to the three elements of software engineering: methods, tools and processes (namely the techniques, supporting environments and processes for developing a software). The teaching activities are scheduled according to the “processes”, the “methods” are preferentially taught, and the corresponding “tools” are used for modelling according to specific projects. After these training steps, the students will get familiar with the processes of software developing and acquire the skills for preparing the technical documents and professional diagrams.

Many kinds of professional charts are used during teaching the software engineering course. Data flow diagram, modular structure chart and program flow diagram are used during instruction of the methods for developing software. The object-oriented methods require use case diagram, class diagram, collaboration diagram, sequence diagram, activity diagram, state diagram, component diagram and deployment diagram. Database design and project managing need the entity-relationship diagram, and Gantt chart and PERT (Project evaluation and review technique), respectively. The diversity of the professional charts renders high demand for the “tools”. They should not only satisfy the need for drawing the professional charts, but also prepare the students for their future studies and works.
2. A summarization for the modelling tools

Modelling is one of the most difficult tasks of software engineering teaching. The purpose of modelling is to help the personnel who are responsible for software developing, design and maintenance understand and maintain the software better. This is a process of extracting essence from the complex information. The unified modeling language (UML) is thus derived. UML is an object-oriented standard modelling language, which can express both the dynamic and static information well during the processes of software analysis and design. UML has become the industrial standard for visible modelling language.

UML modelling allows the application of methods and theories for object-oriented analysis and design. Based on specific definitions and expressions, UML can make the processes of modelling become more intuitive and comprehensible, and to generate corresponding documents and codes according to the specifications [1]. The modelling tools can realize functions including model drawing, model organization, collaborative support, code generation, reverse engineering, document generation, model base management, printing support and scripting programming.

There are more than 100 modelling tools based on UML. Different tools have their own orientations, characteristics and market tactics. Different products and even the different version of the same product may differ in their functions. Among the diverse UML modelling tools, Visio, Rational Rose and Power Designer are most extensively used. Furthermore, for software engineering teaching, the former two tools are mostly implemented. Here, we will compare the advantage and disadvantage of Rose 2007 and Visio 2010 in software engineering teaching.

3. Rational Rose

Rational Rose (simplified as Rose) is a suite of visible modelling tools developed by IBM. The intention of which is to provide comprehensive, robust and efficient resolutions for developing software. Currently, most of the software engineering teaching books use Rose as demonstrating software. Rose is a powerful tool for software engineering teaching, which can be used for all the phases of developing a software, like project analysis, design, realization and testing. Moreover, it also supports the Rational Unified Process (RUP), which is required for large software developing projects.

3.1. Application of Rose in UML modelling

3.1.1. Generation of business and use case models at the demand analysis phase. Business models are used to express the relationships among the business use case, business actors and business workers of a project, which can clearly present the organization structure of the project to the users and vividly demonstrate the business operation relationship of a real organization. Based on the results of demand analysis, the use case models can be established to depict the functions of the system from a participant view.

3.1.2. Specialization of use case charts at the object-oriented modelling phase. Flow charts can be generated based on the sequence diagram, collaboration diagram and state diagram. Rose can be used for static structure modelling by drawing class diagram and package diagram, for dynamic behaviour modelling by drawing interacting diagram and state diagram and for determining components of the system by drawing component diagram. The processes of analysis and modelling is iterative and model of the software project is finally determined after revision and improvement.

3.1.3. Generation of code frames through forward engineering at the software integration phase. Rose can use multiple languages including Ada83, Ada95, ANSI C++, CORBA, Java, COM, Visual Basic, Visual C++, Oracle 8 and XML DTD to generate codes. Therefore, it can satisfy the modelling requirements of nearly all software projects and seamlessly integrate multiple developing environments. After improving the code frame, Rose can revise the models through reverse engineering. For the systematic structure designers, network engineers and system engineers, Rose
generates deployment diagrams to depict the topologic structure of the system hardware and the soft components perform on this structure.

3.1.4. Updating the models at the software delivery phase. At the delivery phase, Rose can update the models according to the revised component diagram and deployment diagram to assure the consistence between the diagrams and the project. This is very convenient for anaphase project maintenance.

3.2. Advantages and disadvantages of Rose in software engineering teaching

3.2.1. Advantages. Team works are usually performed in software engineering teaching to imitate industrial software developing. The manager of the team can use Rose to decompose the huge project and to assign tasks and authorities (full or read only authority) to different team members. When examine the files, you can only extract the needed packages without open the whole system and all the files. This is convenient for team collaboration and parallel software development by multiple users.

Rose supports logical model generation by the relational databases like Oracle, Sybase and Sql Server. The results can be used to generate logical models and concept models by the database modelling tools. The Data Modeler function of Rose can be used for database modelling, which uses UML object model as logic model and data model as physical model and assists the user to synchronize these two processes [2]. Rose can link the model, code and database together from the object-oriented view, and organically integrate the data model and object-oriented development, thus satisfy the requirements of applying multiple databases. The results of modelling can be easily transferred to corresponding code frame and relational database by the automatic transformation function. Application of these functions helps to reduce the developing time and increases the efficiency of database creation [3].

3.2.2. Disadvantages. In addition to graphing of charts for object-oriented modelling, diagrams associated with progress like Gantt diagram and PERT also need to be taught in the software engineering course; however, these are the weak points of Rose. Rose does not support generation of data flow diagram, flow chart, decision tree and critical table. Moreover, Rose has some operation inconvenience, such as the incompatibility to template exportation, the undo operation can only retreat one step and the preference for mouse operation.

4. Visio

Visio is the product of Microsoft, which is dedicated to drawing of all kinds of graphs. As a graphical managing tool, Visio expresses the managing activities as graphs and makes them trackable. In software engineering, Visio can be used for UML modelling and graphing.

4.1 Application of Visio in UML modelling

In software engineering teaching, using the “UML model diagram” function under the “software and database” option can generate all object-oriented UML graphs. Visio integrates the 10 model diagrams needed for UML modelling into the 4 stages of software system development, namely use case model phase, domain model phase, computer model phase and implementation model phase [4]. It is seemly that these stages are continuous, but the processes of software developing are inherently iterative, each stage will be repeated several times.

4.1.1. Use case model phase. In the software demand analysis phase, Visio can depict the interaction between participants and system by generating use case diagram and facilitate understanding the functional requirements of the system by establishing narrative explanations of the systematic processes.
4.1.2. Domain model phase. Visio transforms natural language to object-oriented language after analyzing the use case model to make the users understand the application area of the system under development. The tasks of this phase include generation of static structure diagram, package diagram and sequence diagram. Architecture modeling of software is performed by defining the packages and the inter dependence among packages. The conceptual static structure diagram (class diagram) created by Visio is the most extensively used diagram in UML modelling, which is the important result of the project-oriented analysis and design, and the primary model foundation for system coding and testing. The class diagram depicts the classes of the system and the inter class relationships, such as association, dependence and generalization.

4.1.3. Design model phase. Use case and domain model phases emphasize understanding of the demanding and definitions associated with the system under development. While in the design phase, this comprehension should be transferred to resolutions. The collaboration diagram is applied to determine the mode of communication between objects, the static diagram is used to determine the classes that need to be realized in the software, and the status diagram or activity diagram are used to explore the life cycle of a specific object [5].

4.1.4. Implementation of the model phase. The physical and component structures of the developing environment are depicted by creating component diagram and deployment diagram. Visio can generate code frame wrote in program language according to the design model. The complete code can be accomplished by improving and updating the code frame. This is analogous to the bidirectional engineering of Rose, but these two functions support different languages [4]. Visio supports languages including Microsoft Visual Basic, Visual Basic.NET, Microsoft Visual C# .NET and Microsoft Visual C++.

4.2 Other applications of Visio
The “database” function of Visio can draw the entity relationship diagram. The ORM (Object Role Modeling) template under the “database” option can be used to obtain business regulations and all the graphs required by database conceptual model design. The physical database frame of multiple database management systems can be generated by the logic database frame resulted by the forward engineering of ORM frame. The reverse engineering function of Visio can reversely generate the E-R diagram and ORM data source diagram, then generate all the documents required by the database from the E-R diagram and ORM data source diagram, which is a good reference for the systems possess only database but without design documents [6]. The databases compatible with Visio include Microsoft SQL Server, Microsoft Access, Oracle, IBM DB and Sybase.

The flow chart and data flow diagram required by software development can be generated by the “flow chart” function of Visio. The “schedule” function of Visio can be used to draw the Gannt diagram and PERT diagram and the calendar and time line for the software developing progress plan. In addition to the functions mentioned above, Visio can draw the Jackson data structure diagram, website design map, hierarchy diagram. Moreover, Visio also can be used to create the system diagram, COM (component object model) and OLE (Object linking and embedding) during project-oriented program designing.

5. A comprehensive comparison between Rose and Visio

5.1. Applications if UML modelling
Rose is currently the most popular UML modelling tool. It provides clear expression methods and corresponding software models for the phase from system demand analysis to class analysis and software realization and testing. It supports RUP (Rational unified process) and reflects the iteration and is use case driving. Visio also supports UML modelling, but UML modeling is only a small part of
its function. In comparison with Rose, Visio has better graph quality but is weaker in consistence and correlation.

5.2. **Database modelling integration**

Rose realizes two-way conversion among model, code and database through intermediate plug-ins, while Visio realized these functions through macro. Rose is better than Visio in terms of database modeling integration.

5.3. **Convenience for use**

Rose is oriented to the software developing staffs. Modeling with Rose can assure the consistence between the model and the real situation; however, Visio is suitable for making demonstration graphs and conveying ideas between software developers and the users due to its simplicity and convenience in use. Rose has comprehensive RUP theory foundation and good correlation and is suitable for professional and collaborative software developing project, while Visio is good at graphing. Furthermore, Rose supports more languages than Visio and can satisfy the requirements of users at the maximum extent.

Taken together, in software engineering teaching the teaching contents should be planned according to the course time budget. For application-oriented university students, Rose is the most suitable tool for the modelling tasks of software development and Visio can be used as complement tool. The students will be competent for software developing or service works.

**References**

[1] Li M 2008 Journal of Liaoning Institute of Technology (Natural Science Edition) 28 380-383
[2] Sarman M 2007 Computer Society 155-158
[3] Gao S and Xu H 2010 Journal of Eastern Liaoning University (Natural Science) 17 303-308
[4] Liu X 2004 UML foundation and Visio modelling (Beijing: Electrical Industry Press) p 24-25
[5] Wang L Liu B and Lu M 2011 Chinese Journal of Aeronautics 24 181-194
[6] Jin Z 2004 Journal of zhejiang shuren university 4 84-86