A Comparison between UML Tools

Lena Khaled
Dept. of Software Engineering
Zarqa Private University
Amman,Jordon
lenaumleen@yahoo.com

***Abstract*— through the last few years, building systems became very complicated; any complicated system needs a tool to be built. Thus choosing the best tool needs to know the most important features of it. Today UML is considered as a *de facto* standard in software development and is used in many domains ranging from scientific modeling to business modeling. This paper describes the most used UML tools; it defines the important features of each one then compares between them according to their features, and it also defines the criteria for choosing the best tool in building systems.**

Keywords- *Modeling, UML, Rational Rose, Argouml, Magic Draw, Eenterprice Archietect*

I. INTRODUCTION

A model is an abstraction of a design or a system before building it. Abstraction is a fundamental human capability that permits the designer to deal with complexity. To build any complicated system, the developer must use specific notations, verify that the model, which has been built, must satisfy the requirements of the system and then gradually add details to transform the building model into an implementation. Models serve many purposes, such as visualize and test entity before building it. Models enhance communication, planning, and they also reduce risk and costs [1, 3].

A modeling language is a way of expressing building a model, which has been produced during the developing process. Modeling language defines a collection of model elements. UML, the Unified Modeling Language, is the most popular example to the modeling language [2]. Building a model −that will be either by hand or drawing by the tool− needs a number of things that a modeling tool can provide; such as: reporting, integrating with other process model, synchronizations of models and code. Today there are several UML tools on the market that describe the semantics, notations and constructs of UML. Most of these tools base their descriptions on the meta-model −the layer above the model− of UML. This paper defines the most foremost tools nowadays like: Rational Rose, ArgoUML, MagicDraw, and Enterprise Architect (EA) then compares between them according to some important criteria described later in the following sections.

.

II.RELATED WORKS

Many different researches work on the most-used UML tools; Schweizer Daniel [4] uses Rational Rose as a tool for reverse engineering; he uses the API of this tool and builds UML design extractor which represents a prototype of reverse engineering tool and this prototype is written in Smalltalk language. Garone Steve [5] works on Rational Rose Real Time; this work represents the strong parts of this modeling tool, code generation and visualization of models during execution. Arun [6] works on selecting best tools for software system design; he explores a survey of tools to help determine which ones are the best suited to each phase of the project's life cycle and discovers how to improve time to delivery. This work is a comparison of UML tools according to some of the vital features as shown in the following sections.

III. UNIFIED MODELING LANGUAGE (UML)

UML is a standard language for writing software blueprints. UML may be used to visualize, construct, and document the artifacts of software. UML is appropriate for modeling systems ranging from enterprise information systems to distributed Web-based applications. The vocabulary of the UML encompasses three basic terms: Things, Relationships and Diagrams. Things are the abstractions that are firstly build in a model; relationships tie these things together and diagrams group collections of things [7].

A. Modeling tools

Software development basically measure communication between programmer and machine and between programmer and programmer. UML tools facilitate all these types of communications [8]; they couple with methodologies to give a way that represents complicated systems and comprehend it with their underlying source code and they also enable the developer to represent systems faster and cheaper. Those tools facilitate the following:

- Track, change and mange time.
- Reuse parts of the design through transferable abstractions.
- Communication idea with stakeholders and among them.

Most of the current UML tools are evaluated on the following features

1. HTML Documentation

UML tool should provide HTML documentation for a model; it must provide a static view of the model that any developer who uses the model can refer quickly in a browser without launching the model itself. The HTML documentation should include a bitmap picture of each of

978-0-7695-3937-9/09 $26.00 © 2009 IEEE
DOI 10.1109/ICECS.2009.38

Authorized licensed use limited to: Slovak University of Technology. Downloaded on October 07,2021 at 09:46:52 UTC from IEEE Xplore. Restrictions apply.

the diagrams in the model and should provide navigation through model by using hyperlinks [6].

2. Full support to UML diagrams

The UML tool must support all diagrams of UML; which consists approximately from 9 diagrams to 11 for version 1.5 and version 2.0 which represent the static and dynamic behavior of a system [3, 10]. These diagrams are the following:

- Use case diagram, which defines the behavior of the system, often use in requirement specification phase.
- Class and object diagram which helps explaining of collaborative and hieratical relationships between classes and objects.
- Interaction diagram which represents active communication between objects; it shows in detail how objects interact to perform a task.
- Package and collaboration diagrams. A package is a collection of model elements. Collaboration is a representation of objects when interacting to perform a task together with the links between them. Both diagrams describe the overall behavior of the system.
- State diagram that shows the state of objects or system changes.
- Activity diagram which describes sequence of activities. It shows how an operation could be implemented
- Physical diagram that defines high level of a system in its deployed state.

3. Round-Trip Engineering

This is the ability to both forward and reverse engineering source code. After giving the model, the tool should be able to generate code it should be able to generate the model from the existing code [9]. At least, the modeling tool should support forward engineering at the first time and support reverse engineering through the process. Reverse engineering is very important to resynchronize model with the code and to transform code to a model [6].

4. Integration with data modeling tool

UML tool should have a facility to integrate with data modeling too, this can be done through a feature allowing an object model to be transformed into data description language which needs SQL to create tables from classes [6].

5. Exporting diagrams

UML tool should have an ability to export diagrams into a format that may be imported either into a word or web page. These are the most popular used formats: GIF, JPEG [6].

6. Robustness

UML tool must be a high reliable tool to prevent users from losing product when the tool crashes in the middle of a design session.

Above definitions are the most important features that most UML tools must include. The following sections define UML tools according to defined features.

B. UML tools

Today in the markets, there are several UML tools which describe the semantics, notations and constructs of UML. Most of these tools base their descriptions on meta-model. The following sections define some of the most used tools.

1. Rational Rose tool

Rational Rose is a visual modeling tool; it helps developers to make visual models of software using UML diagrams to catch the aspects of software project.

Rational Rose is a powerful tool that support analysis and design of object oriented software systems. It helps modeling system before writing a program. This makes sure that the system is architecturally strong from the beginning [11]. Rose includes all UML diagrams such as: use case, objects, classes and components. This tool is called a picture of the system. The properties of this tool are these:

- **Reverse Engineering**

Rose tool can be integrated with multiple languages. This integration support reverse engineering into UML model; when requirement changes, it can change code directly, rather than changing model then changing code of that model. This feature reduces risks of having a model outdated [11]. In addition to transform model-to-code, rational software is able to transform model-to-model. For example, it is able to convert sequence diagrams to collaboration diagrams and vice versa [13].

- **Supports for UML notations**

Rational software's richness supports the UML 1.x diagram types and the new version which is UML 2.1. It supports Use case, class, sequence, collaboration, and state and activity diagrams [12, 13].

- **HTML documentation**

Rational Rose is able to generate HTML documentation in its work.

Other features:

- Automated traceability and that is from requirement to design.
- Easy installation and ease of use.

Authorized licensed use limited to: Slovak University of Technology. Downloaded on October 07,2021 at 09:46:52 UTC from IEEE Xplore. Restrictions apply.

- Tight integration with other IBM Rational software design, which make collaborative development very easy.

2. *ArgoUML tool*

This tool is based on UML diagram application; it is written in Java and is available on any platform supported by Java [14]. Its properties are:

- **Reverse Engineering**

ArgoUML supports this feature with Jar/class file importing. It also supports forward engineering where code generation is based on c++, c#, Java, PHP4, PHP5, Python, Delphi and SQL [14, 15].

- **Supports for UML notations**

This tool supports 9 diagrams of UML 1.4 version. One of them main weakness in this tool is that sequence diagrams do not function very well, and it doesn't support UML 2.0 version [14, 15].

- **HTML documentation**

ArgoUML seems to lack writing HTML documentation to some parts of the diagram [14].

Other features are [14, 15]:

- Export diagrams as GIF, PNG, PS, and SVG.
- Because it is based on Java, its platform is independent.
- It is very easy to install, in most cases it can be installed with one click when using Java web.

3. *MagicDraw tool*

MagicDraw tool is the product by NoMagic technology partner. MagicDraw provides excellent supports for UML-based modeling, particularly in the problem space [16]. These are some of its properties:

- **Reverse Engineering**

It is a powerful reverse engineering tool. When constructing a model, the source code can generate from it easily; writing more code in IDE then reverse it, make some changes by the tool and merge the modified model with the code. No details are lost.

- **Supports for UML notations**

This tool supports diagrams of UML 2.0 version.

- **HTML documentation**

This tool facilitates reporting. It can generate HTML documentation for each model element. It has the ability to select some parts of the model and make them appear in the report.

.

Other features are:

- More than one developer can work at the same time with the object oriented model.
- The ability to design the schema of database in UML class diagram then generates its DDL.
- Providing multiple set of design patterns including Java design patterns.

4. *Enterprise architect (EA) tool*

It is one of many CASE tools currently wide used to aid diagramming in UML within a development process. It is from Sparx systems. It has many properties such as:

- **Reverse Engineering**

EA supports this feature, and it also extends its functionality by using Model Driven Generator technology (MDG) that is downloadable plug-ins which enable engineering to require any particular language.

- **Supports for UML notations**

This tool supports super structure of UML 2.0 extensional which include the new diagrams. Interaction view and timing also extend functionality like fragments and composite structure [10].

- **HTML documentation**

EA supports HTML documentation [10].

Other features are

- Ability to create multiple classes that communicate together to form a pattern.
- The creation of a platform independent by using MDA.
- Supports Tagged Value. Tagged Value is label and can be attached to the element of UML.

IV. The Results

In these days, many tools appear in the market to support building system. The architect is almost involved with all phases of software development life cycle and makes decision about which tool he must use through the cycle of the system. Most important tools are those, which are described through this paper, can be used across analysis, design, coding, and testing with quality assurance. The Table that follows is a brief comparison between the tools that are described.

Table I. A brief comparison between UML tools

Features	Rose	Argo UML	Magic Draw	EA
HTML Documentation	√		√	√
UML NOTATION	support UML 1.x UML 2.	supports UML 2.	supports UML 2.	supports UML2. extended
REVERSE ENGINEERING	√	√	√	√

Authorized licensed use limited to: Slovak University of Technology. Downloaded on October 07,2021 at 09:46:52 UTC from IEEE Xplore. Restrictions apply.

The following paragraphs summarize the usage of tool through the stages of life cycle as the following:

If a developer is looking for a tool that supports activities and links all diagrams, which are built through, with code, Rational Rose is the best choice. Rose can be integrated with other Rational tools such as requisite pro, test manger and SODA.

If a developer is looking for a tool that is used for design and architect system, then ArgoUML is the best choice. Rational Rose would be useful too.

If a developer is looking for analysis, to both software and system quality engineer, and also for writing documentation, in this case MagicDraw is the best.

EA is good for business modeling because it has its own words in modelling such as business goals which represents a use case in other UML tools, business events which represents an event, and business process which represents activities.

Table II. A brief usage for UML tools

UML tools	
Rational Rose	**The best choice for support all activities of the system and link diagrams with code.**
Argouml	**The best choice for design and architect system.**
MagicDraw	**The best choice for analysis and writing documentation.**
Enterprise Architect	**The best choice for business modeling.**

V. CONCLUSION

UML has been considered as a standard *de facto* in software development processes of many companies, large or even small. Selecting the right tool for modeling is very important point to any developer when he wants building the system. This paper is intended to define criteria to select the best tool for building a complicated or even a simple system. It defines the most important tools in market such as: Rational Rose, ArgoUML, MagicDraw EA, also it describes the features of each one, then compares between them.

REFERENCES

[1] Balaha Michael,Rumbaugh James, “Object oriented modeling and design with UML,” Pearson education, 2005.

[2] Stevens Perdita, “Using UML Software Engineering with Objects and Components,” Addison Wiesly, 2006.

[3] Garret Christopher, “software modelling introduction,” Borland, 2003.

[4] Schweizer Daniel, “Exporting MOOSE models to Rational Rose UML,” University of Berne, 2000.

[5] Garone Steve, “Rational Rose Real Time,” Rational Software Corporation, 2000.

[6] Object by design corporation, 2005. http://www.objectsbydesign.com/tools/modeling_tools.html.

[7] Booch Grady,Jacobson Ivar, Rumbaugh James, “ The Unified Modeling Language User Guide,”addison wiesly , 1999.

[8] keuffel warren, “ A Trio of Object–Modeling CAEE tools,” DBMS, 1997.

[9] Gasso Wilson, Jonathan P., “Evaluation of some of the current UMl tools", South Bank University, 2002.

[10] Meyer Jeremy , “comparison of UML modeling tools,” Dunstanthomas, 2005. http://consulting.dthmas.co.uk.

[11] Rational Rose, “USING ROSE,” Rational Software Corporation, 2000.

[12] Chapter Arune, ”Selecting the Best tools for your software system design,”IBM, 2007.

[13] Meyer Jerermy, “Comparison of UML modeling tools,” IBM Rational, 2003.

[14] lum Rosalyn, “Archietecture& Ddesign : Reader’s choice Awards,” 2003. http://www.ddj.com/architect.

[15] Smith Grant, “Survey of open source modeling tools for software development,” Wazi, 2009.

[16] Bhuvan Unhelkar, “Process Quality Assurance for UML-based projects,” Pearson education, 2003.

Authorized licensed use limited to: Slovak University of Technology. Downloaded on October 07,2021 at 09:46:52 UTC from IEEE Xplore. Restrictions apply.