ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 6, Issue 2, pp: (4-11), Month: October 2019 – March 2020, Available at: www.paperpublications.org

A Study of the State of the Art UML Case Tools Suitable for Supporting Teaching and Learning of Software Design and compatible with an ALL approach to the subject

Abdulaziz Alkhuwayr

Abstract: **The *Unified Modeling Language* (UML) is referred to as unified since it attempts to unify existing methodologies, with the particular use of characteristics and symbols to put into use the various features of modeling. There are notable software changes still taking place in UML which involves an approach of the entire system. UML is a language unfortunately is not easy to understand but with enough time and practice one may understand it. For the researcher to establish a method for rating the tools, the researchers came up with two questionnaires to help in rating the tools. The Questionnaire-1 aimed at selecting the leading seven tools used by the interviewed subjects while questionnaire II was given to students and academic staff to fill. The main propose of the second questionnaire was to help in identifying the most used tools and rate them.**

With changes in technology taking place now and then, it is advisable to teach students new and popular tools to keep them updated with the latest technologies.

Hence the researcher recommended Rational Rose to be replaced. The criteria for coming up with a new tool should be based on cost as well as how easy to learn the tool is. The below options may be suitable for replacement: Several diagrams can help users to get a clear structure or picture of the system and the existing communication between the users and the modules. One can conclude that *Unified Modeling Language* is not hard to learn, teach and is fundamental for complex OOSD. The market has several UML CASE tools that assist in modeling and designing.

1. BACKGROUND

Unified Modeling Language UML is today a standard of application in the Object-Oriented Software Engineering. CASE UML tools have replaced papers in designing software. There are so many CASE UML tools available today with special code features like support of all UML 2.x diagrams, reserve engineering, and code diagrams which has made it difficult to choose the right CASE UML tool (Allen, 1999). Besides, people may not be readily willing to change their already used tools. The existing CASE UML tool in use at campus labs is Rational Rose (Bell, 2004). The main purpose of this research was to compare the available tools and to decide whether the Rational Rose tool should be replaced with some other tools or not.

2. METHODOLOGY

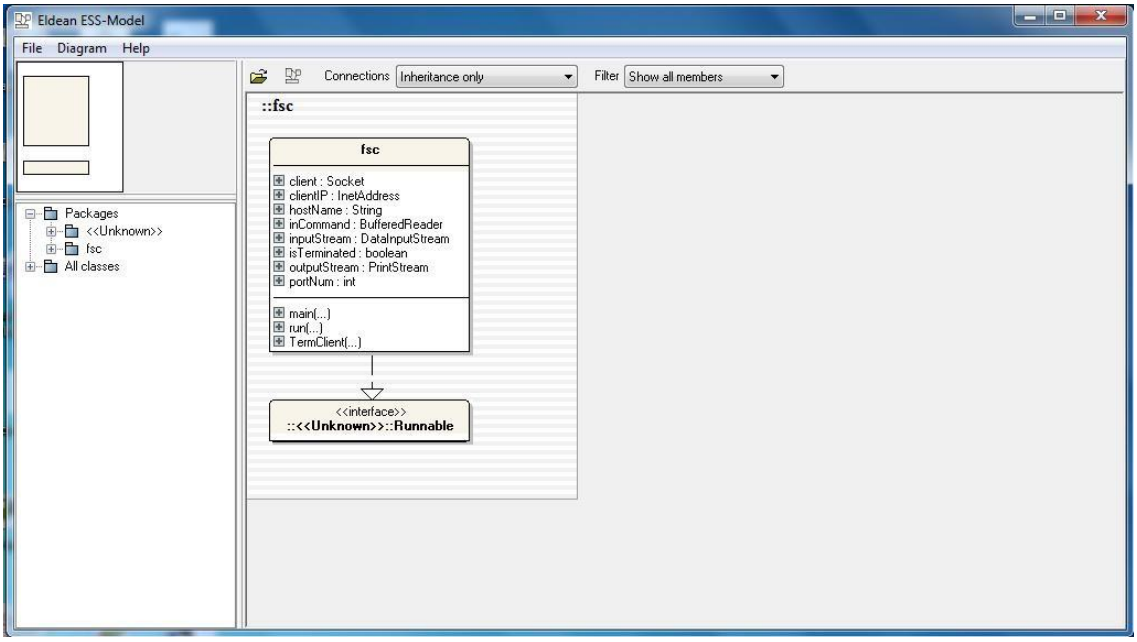
The UML is referred to as unified since it attempts to unify existing methodologies, with the particular use of characteristics and symbols to put into use the various features of modelling. There are notable software changes still taking place in UML which involves an approach of the entire system. UML is a language unfortunately is not easy to understand but with enough time and practice one may understand it.

The UML uses the concept participant or Actor within any activity domain starting from the top. Actors interact with processor data, or with one each. This interaction leads to a Use Case. Use Case in other words is view of the entire system or a particular section of a system. A Use Case as such does not reveal information about a whole system. Various related

Use Cases can also be grouped into a huge package known as Package. An individual Use Case may have several uses such as deposit, withdraw and check balances among others just like an ATM. Therefore, the primary aim of the Use Case is to observe and study the components and behaviours of the system abstractly and generically. That is also referred to as the specifications of the requirements of the system.

ESS-Model

ESS-Model is a freely available UML tool that makes it easy to view the class diagram without delays in the shortest time possible. Delphi/Kylix and Java are some of the programming languages supported by ESS MODEL (Eldean AB, n.d). Besides the class diagrams, the system also generates Html documents. ESS MODEL has a small footprint. It is only a single 700kb exe file that is needed (ESS-MODEL, 2011). It is easy to install as on can just copy paste the file to his or her computer. The file loads instantly. The main features of ESS-MODEL are: Delphi IDE integration, Java .java source files, aautomatic generation of UML, Delphi source files, Single 700kb exe file, no setup needed, Java, class binary files, Export model data to XMI format (Eldean AB, n.d).



Connections: Inheritance only

Filter: Show all members

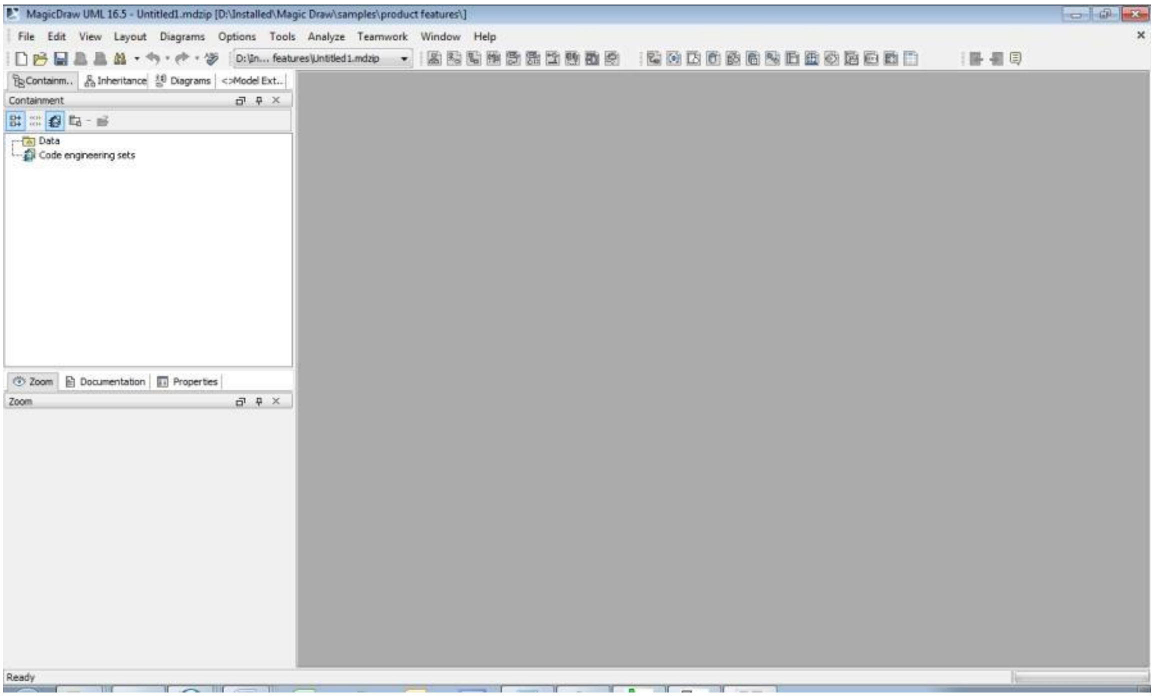
MagicDraw: MagicDraw is a tool that acts as a support for both the software and system modeling. In addition, it also supports business process. MagicDraw is supported by UML 2.o and HTML documentation generation which make it have a powerful reverse engineering capability (Lena, 2009). MagicDraw has great crucial features of providing code engineering mechanism for C++, C#, Java, and CORBA with DDL generation and database schema modeling (About StarUML, 2011). Some distinct features of the latest magicDraw include:

Cameo Team Server integration: Its function is to support the current version of the UML specification which includes UML meta-model and notation changes.

Matrixes: Matrixes allows a quick analysis of the system. They determine and indicate if two models are related and what type of relationship exists between them any. This feature helps in the management of the ordinary relationship between elements and makes it easy to create faster traceability links between elements.

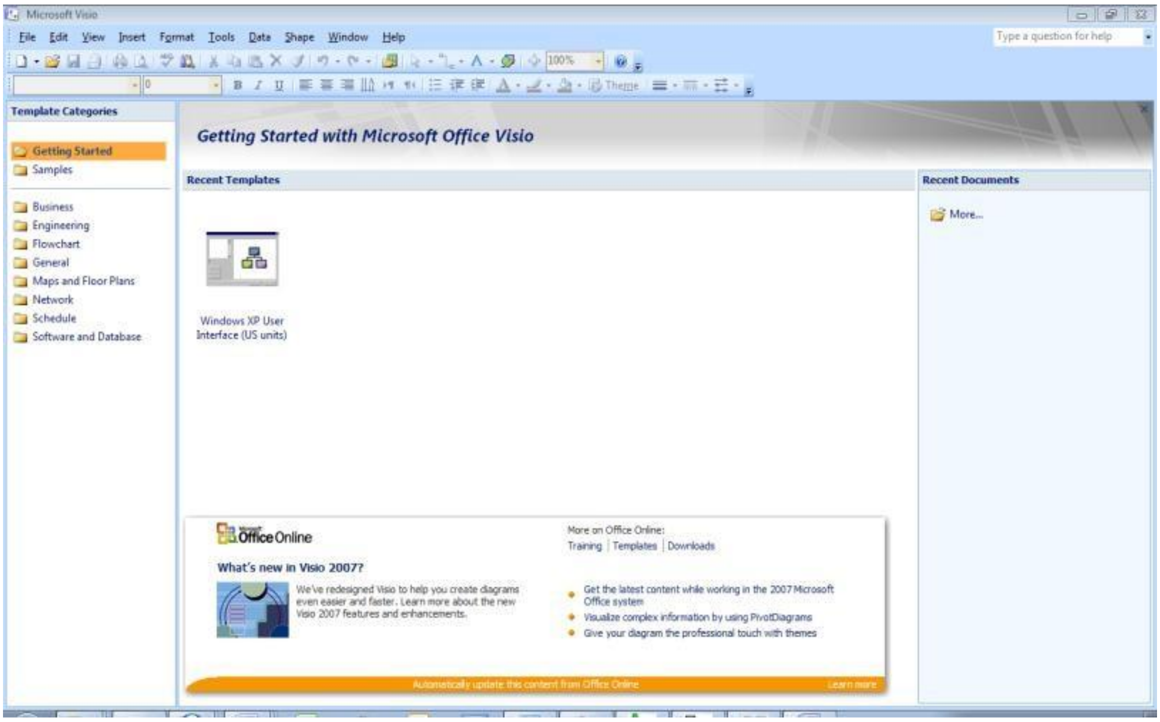
QVT (Query/View/Transformation): The object management group defines QVT standard. The main purpose of QVT is to specify model to model transformations. QTV is a transformational language under the many Model to model transformation. The new QVT ensures that MagicDraw is fully utilized. Transformations can be run directly in MagicDraw.

TOGAF: TOGAF is a standard that deals with the design, planning, implementation and governance of enterprise information architectures in terms. TOGAF plug-in comes with the following features for Magic Draw users: A well updated architecture met-model for the content sample project, TOGAF project template, TOGAF diagram for content presentation.



Microsoft Visio

This is a modern art diagramming solution for dimensions such as UML. Visio can be used by any IT user as a drawing tool. Microsoft Visio for 2010 widows is available in standard, premium and professional editions. Mac OS X or Linux operating systems do not have a Microsoft Visio. Not many Linux diagramming and Mac OS X programs can read Visio files. One can read and write visual flows using Omnigrafle Pro on the Mac.



What's new in Visio 2007?

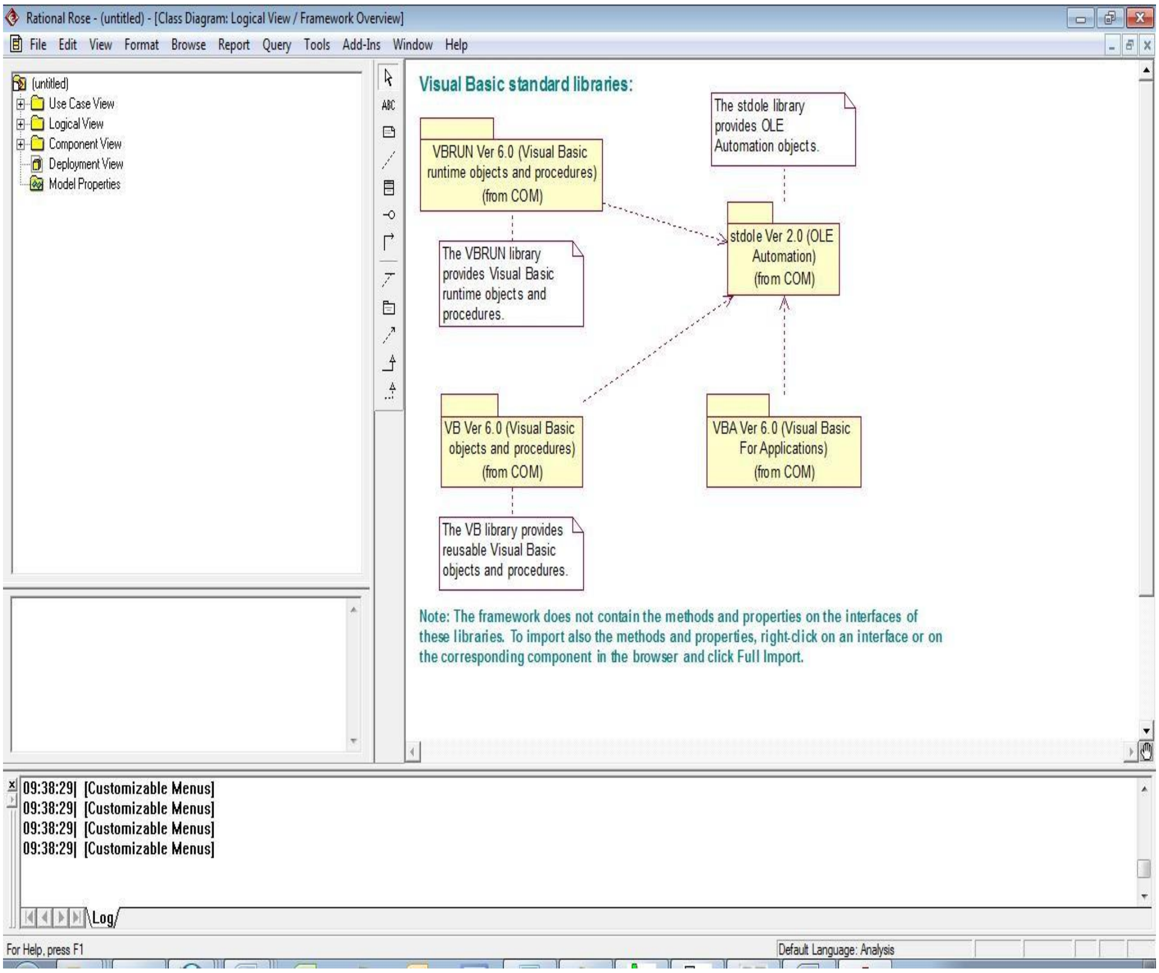
We've redesigned Visio to help you create diagrams even easier and faster. Learn more about the new Visio 2007 features and enhancements.

- Get the latest content while working in the 2007 Microsoft Office system
- Visualize complex information by using PivotDiagrams
- Give your diagram the professional touch with themes

Automatically update this content from Office Online

Rational Rose

One of the commercially available case tool software is Rational Rose. It supports controlled iterative and component-based development elements of modern software engineering. Versions created with this tool can be observed with several Ultra Model Language diagrams. This software tool also supports *Round Trip engineering* (Zhang, Williams & Gatherer, 2016). Rational Rose can support several kinds of user activities and link models with source code. It supports both UML 1.x and UML 2.1 which can generate HTML documentation (Lena, 2009). It is is an object-oriented (UML) software design tool that assists in component construction of enterprise-level software applications. As the software is being constructed, it keeps a record of the diagrams and then generates code in the designer choice of Java, Corba, Oracle8, v or Data Definition Language. Rational Rose can provide iterative development.


Rational Rose - (untitled) - [Class Diagram: Logical View / Framework Overview]
File Edit View Format Browse Report Query Tools Add-Ins Window Help
(untitled)
Use Case View
Logical View
Component View
Deployment View
Model Properties
Visual Basic standard libraries:
VBRUN Ver 6.0 (Visual Basic runtime objects and procedures) (from COM)
stdole Ver 2.0 (OLE Automation) (from COM)
VB Ver 6.0 (Visual Basic objects and procedures) (from COM)
VBA Ver 6.0 (Visual Basic For Applications) (from COM)
Log
For Help, press F1
Default Language: Analysis

The stdole library provides OLE Automation objects.

The VBRUN library provides Visual Basic runtime objects and procedures.

The VB library provides reusable Visual Basic objects and procedures.

Note: The framework does not contain the methods and properties on the interfaces of these libraries. To import also the methods and properties, right-click on an interface or on the corresponding component in the browser and click Full Import.

```
09:38:29|  [Customizable Menus]
09:38:29|  [Customizable Menus]
09:38:29|  [Customizable Menus]
09:38:29|  [Customizable Menus]
```

Star UML

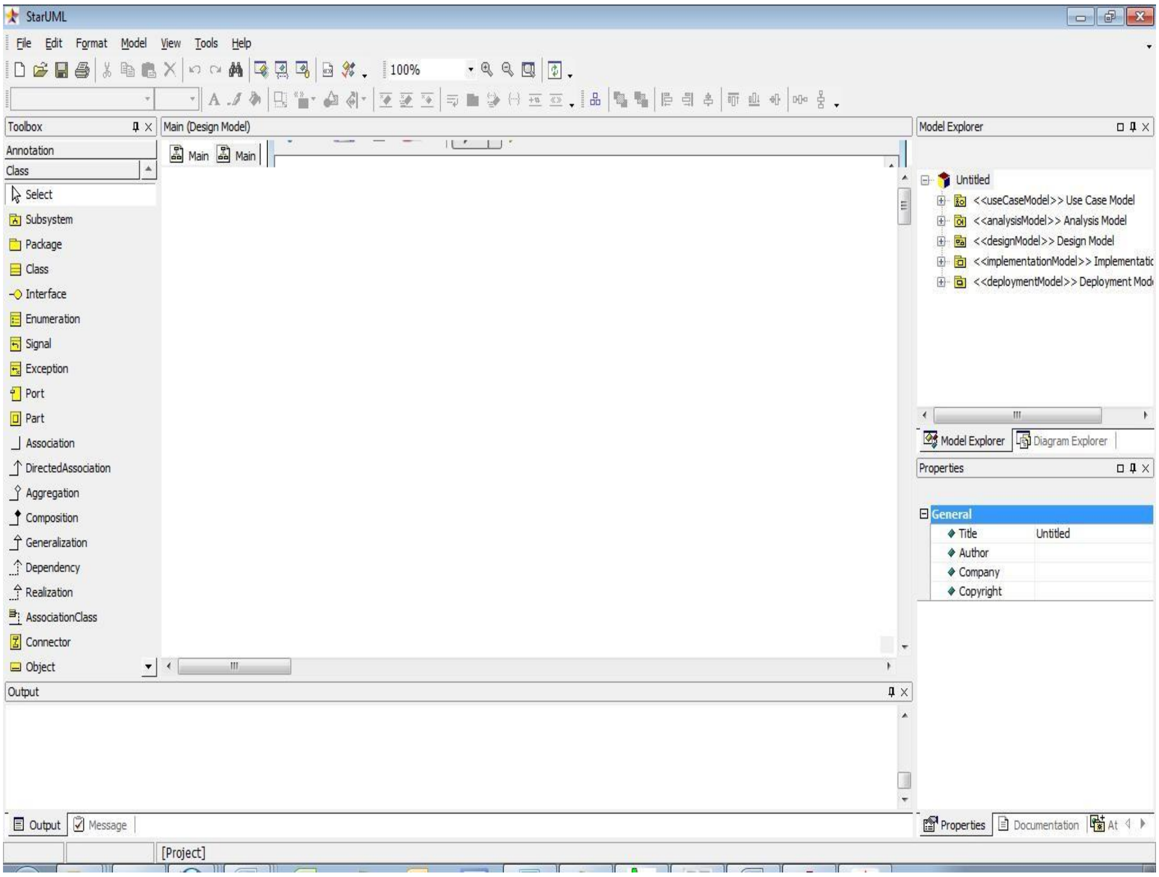
Star UML is a more comprehensive tool that provides support for UML 2.0 and MDA. StarUML has a high likelihood of replacing tools such as Rational Rose which are commercially available. Star UML is an excellent plug-in structure that enables plug-in modules to be improved in COM compatible languages like C++ and Delphi among others (StarUML, n.d). StarUML primary goal is to come up with a ssoftware modeling tool and an excellent platform that can replace Unified Modeling Language tools like Rational and Together among others (StarUML, n.d).

UML 2.0: Object Management Group manages UML as it continuously expands. MDA (Model Driven Architecture) came up when OMG was introduced. For the full advantage or effect of MDA to be felt, the software modeling tool should support customization variables like a Model Framework, UML profile, Document template and MDA code among others.

Plug-in Architecture: StarUML has a less complex and yet a powerful plug-in architecture which is crucial as anyone can develop plug-in modules in languages that are COM compatible.

Usability: When StarUML is fully implemented, it comes with many user friend characteristics like Diagram overview and Quick dialog among others.

The main language of witting StarUML is Delphi; however, its use is not limited to one specific language and can be used by any programming language.



General	
Title	Untitled
Author	
Company	
Copyright	

Visual Paradigm

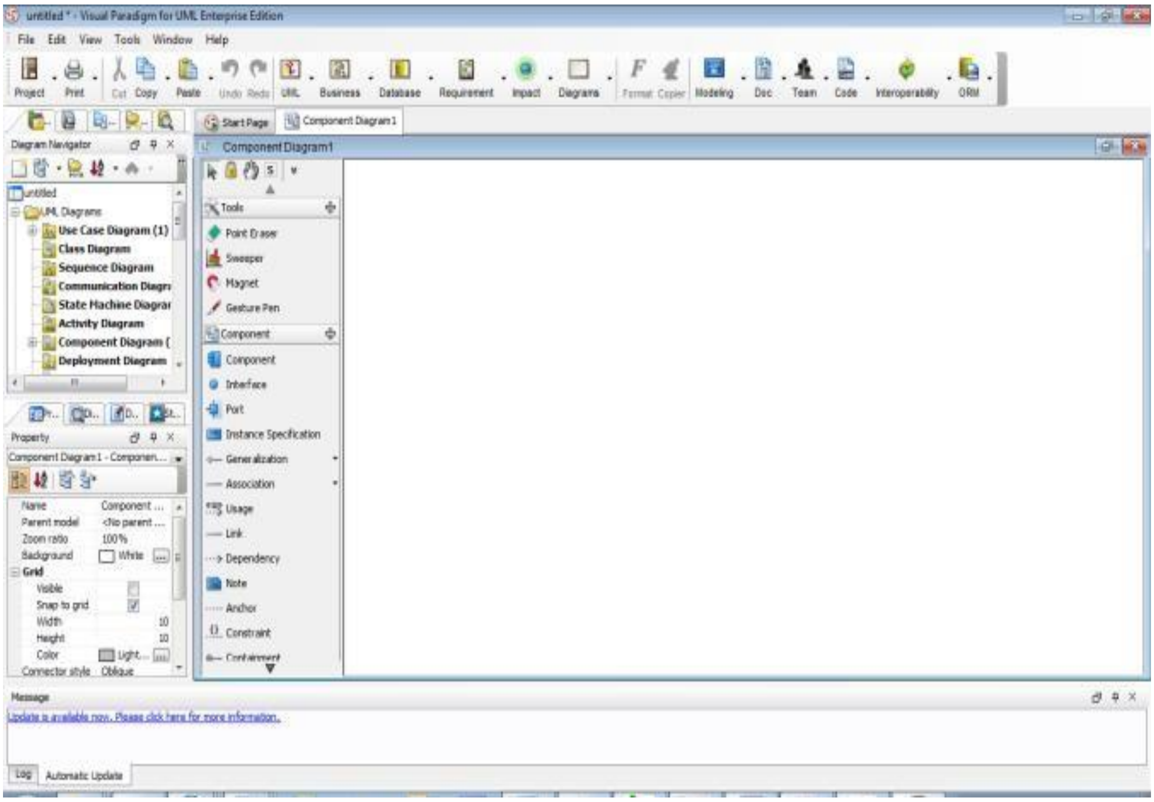
Visual Paradigm is a modern CASE Tool that supports both UML 2.0 and Business Process Modelling Notations and SysML. Visual Paradigm also has report generation capabilities (Visual Paradigm International, n.d). Despite Visual Paradigm being a powerful tool, it is easy to use. VP UML enables software developers to develop a platform to build a quality application within a short period at a low cost (Visual Paradigm for UML 2011).

Data modeling: Apart from modeling the database table, one can also model stored procedure, sequence, triggers and database view in an ERD. Entity Relationship Diagram and ORM Diagram can be drawn to help in database modeling.

Business process modeling can draw business process diagrams, Event drive process chain diagrams and business process diagrams to help in business process modeling.

BoUML

BOULM is a free UML 2.X tool that helps in generation code in Php, Python, C++, Java, and IDL. BOULM tool supports operating systems such as macOS X, Linux, and Windows (BOUML, 2011). BOUML has high speed and does not require large memory to manage the numerous classes. Bouml is extensible and the plugin outs can be written in Java or C++. BOULM has the following inbuilt tools, C++ code generator, round trip, reverse, Roundtrip body and Php code generator among others.



Name Component ...
Parent model <No parent ...
Zoom ratio 100%
Background □ White
Grid
Visible □
Snap to grid ☑
Width 10
Height 10
Color □ Light...
Connector style Oblique

Update is available now. Please click here for more information.

3. DISCUSSION

The UML was a significant contributor to the success of these projects. Besides, the software developers and database managers who began using SDL and ER diagrams found UML to contribute a lot in their work. At the end of this project, one could conclude that software modeling is crucial in making designs better even in cases that seem too hard to solve. The researcher also saw the need advanced modeling tools for making the advanced and complex software systems with efficient and reliable design. This made companies to come up with numerous limited methodologies while still looking for an efficient tool that be a ssolution many problems. In the 1990s only a few people could understand field methods and notations. Ivar Jacobson came up with (OOSE) that did not provide the designers with several dimensions. Meantime (OMT) that emphasized analysis was developed by James Rumbaugh whereas Grady Booch presented his methodology that emphasizes design and implementation. The two were combined later to a single comprehensive software modeling standard named UML (History of UML 2011). UML was established in 1994. Since it was launched, UML has made many positive changes to the IT industry that more than three-quarters of developers use it as a modeling tool for Object-Oriented. More than half of the software modeling tools UML are based (Pender 2003:3).

UML Overview

In this project, *Unified Modeling Language* proved to be a standard in the modeling of OOS. In UML modeling a set of graphic notations creates visual models. UML provides a methodology to visualize elements such as business processes, activities, actors, programming language statements, database schemas, and reusable software components. In the beginning, UML was not considered to be a complex modeling tool however; it later proved to come up with changes in programming from low level to advanced modeling.

Today, not only one can find tools to help in developing visual models but also generate codes thus making work easy for programmers.

UML Diagrams

There were 13 diagrams in UML 2.X and 3 classifications. The 3 classification of diagrams is as follows:

Behavior: Behavior includes the entire diagram that shows the behavioral characteristics of the process and system. Some examples of behavior include state, activity, and use case diagrams.

Interaction: Interaction is among behavior diagrams that subject primary emphasis on object interactions.

Structure: Structure is a diagram that shows the characteristics of the specifications. Below is a brief description of the 13 diagrams:

Activity diagram: The primary function of the activity diagram is to model the logic complete within a system. This is the most preferred diagram for learners.

Classic Diagram: Class diagram represents static model elements such as classes and their relationship. The diagram also has a high learning priority.

Communication Diagram: Initially, it was known as the Collaboration Diagram. Communication diagram represents the objects, how they relate and message flow. The diagram has a low priority.

Component Diagram: Component diagram represents the elements of an application, interfaces, interaction, and their relation.

Composite Structure Diagram: This is composed of the internal structure of a class, component and interaction points.

Deployment Diagram: This is a diagram that represents the execution of the architecture of systems.

Interaction Overview Diagram: The main function of these diagrams is to control flow within a business process or a system.

Object Diagram: This kind of diagram represents objects and their relationships at a specific point in time.

Package Diagram: Package diagrams give the structure of the organization of elements into packages. Besides, it also shows the dependencies between packages.

Sequence Diagram: The primary function of the Sequence Diagram is used to model the sequential logic and deliver messages in time.

State Machine Diagram: State machine shows how the object looks like as well as what takes place between one state to another.

Timing Diagram: Timing Diagram shows how and how an object changes with respect to the internal and external environment.

Use Case Diagram: It represents the actors, use cases, and their relationship.

UML Class diagrams

In UML a class represents common state and behavior. Besides, the relationship between the seven classes is also necessary to represent the behavior of the systems. In UML, the class is also known as a classifier through a component, a data type, and interface is referred to as classifiers (Pilone 2005:11).

UML CASE Tools Comparison

For the researcher to establish a method for rating the tools, the researchers came up with two questionnaires to help in rating the tools. The Questionnaire-1 aimed at selecting the leading seven tools used by the interviewed subjects while questionnaire II was given to students and academic staff to fill. The main propose of the second questionnaire was to help in identifying the most used tools and rate them.

Below is a list of the 7 tools selected based on the analysis of the first questionnaire: *Visual Paradigm, MS Visio, StarUML, BoUML, ESS-Model, MagicDraw and Rational Rose*.

After tools selection, the subjects were then trained on how to use these tools and answer questions in questionnaire II. In the process of filing questionnaire two, the selected tools were rated.

Comparison of the chosen tools

The comparison aimed at creating a deeper understanding of their performance abilities using parameters such as *UML, MDA, XML Metadata Interchange, Language Generated, Reverse Engineering Languages and integration* which were compared for different parameters.

The result indicates that MagicDraw supports most of the primary characteristics including reserve engineering and language generation. MagicDraw generates many languages from diagrams that include *C++, C#, CIL, IDL, DDL, EJB, XML Schema, CORBA, Java, and WSDL*. Besides, MagicDraw also supports reverse engineering of the mentioned languages. Another significant characteristic of MagicDraw are its integration with EMF, NetBeans IDEs and Eclipse. Though Microsoft Visio is a good UML tool, it fails to support code generation and reserve engineering despite supporting many other features. Rational Rose misses basic features such as MDA support and 2.0 UML although it is the most preferred and used CASE UML tool. StarUML supports almost all available features while Visual Paradigm does not support the MDA feature.

4. RECOMMENDATIONS

After comparing several tools in this research, it was evident that Rational Rose, StarUML and Visual Paradigm are the most powerful and the most preferable modeling tools for learning. However, today, Rational Software Architect and Rational Software Modeller are replacing Rational Rose. With changes in technology taking place now and then, it is advisable to teach students new and popular tools to keep them updated with the latest technologies.

Hence the researcher recommended Rational Rose to be replaced. The criteria for coming up with a new tool should be based on cost as well as how easy to learn the tool is. The below options may be suitable for replacement:

(i) Upgrade existing Rational Rose with Rational Software Modeller and Rational

(ii) Software Architect (iii)Replace it with StarUML

(iii) Replace it with Visual Paradigm

5. CONCLUSION

The research first described what UML means and how it acts as a reliable technique and methodology for designing a comprehensive software system that can be used in many applications since it is easy to understand. Several diagrams can help users to get a clear structure or picture of the system and the existing communication between the users and the modules. One can conclude that Unified Modeling Language is not hard to learn, teach and is fundamental for complex OOSD. The market has several UML CASE tools that assist in modeling and designing. However, only seven out of the many tools those were described and compared in this project.

REFERENCES

[1] Allen, P. (1999) *Component-based development for enterprise systems: applying the Select Perspective*. Cambridge: Cambridge University Press

[2] Bell, D. (2004)‘UML basics: The class diagram An introduction to structure diagrams in UML 2’. *IBM Corporation.* Available from <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/> [2011]

[3] BOUML [2011] *BOUML Features* [Online] Available from <http://bouml.free.fr/features.html> [2011]

[4] Deacon, J. (2005) *Object-Oriented Analysis and Design*. USA: Addison-Wesley Eldean AB (n.d) *ESS-MODEL* [Online] Available from <http://essmodel.sourceforge.net/index.html> [2011]

[5] MicroSoft Corp. *MS Visio* [Online] available From < http://visio.microsoft.com/en-us/Pages/default.aspx> [2011]

[6] Thomas, D. (2003) ‘UML - unified or universal modelling language’. Journal of object technology 2 (1), 7-12

[7] Visual Paradigm International (n.d) *Timing Diagram* [Online] available from <http://www.visual-paradigm.com/VPGallery/diagrams/TimingDiagram.html>[2011]

[8] Zhang, L., Williams, R. A., & Gatherer, D. (2016). Rosen's (M, R) system in Unified Modelling Language. *Biosystems*, *139*, 29-36.

[9] StarUML (n.d) *About StarUML* [Online] Available from <http://staruml.sourceforge.net/en/about.php> [2011