International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Unified Modelling Language (UML) Tool for Software

Sweta Singh Patel[1], Arpana bharani[2]
[1]Research scholar at Dr APJ Abdul Kalam University Indore, Madhya Pradesh
[2]Assistant professor at Dr APJ Abdul Kalam University Indore, Madhya Pradesh

Abstract - The production of software has going to move a lot much beyond the traditional development of software, described as an interactive autonomous software component by the structured programming paradigm of the late sixties and early 1970s. The study seeks to identify a UML profile, outline software projects in which UML was used to evaluate the use and efficiency of UML diagrams, determine the use of CASE tools and record the perceived use of UML language. A study survey was developed for IT professionals and university students. The research was conducted. There have been mailing lists. The findings indicate that UML is used in most software development projects successfully and that many of the users consider UML to be good since it helps to build the system more rapidly and to produce excellent software systems. For performance risk assessment, UML diagrams are used, a software model is created for each scenario, and are translated into a system execution model through deployment data.

***Keywords*:** UML, Software, Case tool, XMI, Argo

Introduction

The development process has been the subject of research for decades. There have been several advances in the visual modeling of complex, large software systems during the last few years. The development, as well as stringent performance, reliability and safety criteria of satellite systems with large and complex requirements are becoming ever more demanding. The constantly growing need for quality, reliable and shorter development periods for on-board quality software with stringent demands for performance means that optimal software development methods need to be used. For safety-critical systems, model-driven development becomes increasingly common and the research examines how visual modeling may be used to design the system and software architecture for spaceflight systems. Recent trends have shown that 50-60% of every project's success is accountable for requirements. Inadequate collection, analysis and management of requirements are the cause of most of the project failures. If requirements are not adequately defined and documented, it will affect the entire project. Lack of requirements leads to unfulfilled systems, leading to unmet design components and functions. Contemporary approaches utilize a model-driven approach supported by CASE tools, such a unified modeling language to solve this problem (UML).

Software architectures are high-level architectural representations which enable effective segmentation and parallel software system developments, give a way to guide and assess the system and, in the final analysis, provide reuse opportunities. The term architecture is not new; the physical structure of artifacts has been utilized for thousands of years. The phrase is created by the community of software engineers to characterize software-intense systems at their gross level. The importance of structure was recognized early in the history of software engineering. The first software programs were created in programming languages that permitted numerical calculations using mathematical expressions and later with algorithms and abstract data. The programs that were created at the time had just one purpose and were extremely rudimentary compared with today's vast and diverse software systems. With the increasing complexity and size of applications, the global structure of the software system became an important issue. Dijkstra proposed the correct organization of the software system structure in 1968 before simply programming. In operating systems, the layered organisation idea was created, which divided related programs into layers and interacted with programs at the nearby levels. Parnas subsequently stated that the decomposition criteria employed affected the program structure, and numerous design principles were necessary in order to create a reasonable structure. Software engineers are now agreeing that the software system structure is important and that a good structure should be governed by specific design principles.

Stem and software architecture, design and conduct It helps to integrate the flexibility and speed of modeling notation of traditional programming languages. It offers both system engineers and software developers a single common language, enabling improved communication and cooperation. It also enables large and distant teams to collaborate while controlling ongoing changes and having a clear view into their possible impact, since changes are inevitable in a project.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

UML has been used by systems and software engineering experts to depict systems since its conception. Lockheed Martin has described the use of UML for operating concepts, system needs, software requirements and test cases. UML was utilized for system/segment designs, including distributed architectures, in some circumstances. The applications included "DoD, government, business and commercial applications, including distributing command, control, communications and information systems, embedded real-time signal, and mission-avionics subsystem data processing applications, system simulators, and general business systems.' At least one application is also available for a hardware-intensive aviation system."

Literature review

Aditya Kurniawan (2014) The advent of Web 3.0 allows web technologies today to cooperate in the performance of a job for each user. This technology enables us to create online UML diagrams and work together on a software project. Unified Modeling Language is one of the most often used tools for architectural modeling. This study seeks to create a tool for modeling UML diagrams are class schedules, case schedules and activity schedules utilizing the current web using HTML 5 technology, along with JSON Service that enables software developers to work on the same UML project and to work together more quickly than usual on the internet.

M.H. Aabidi (2017) In recent decades, the quantity of data generated by applications in science, engineering and biological sciences has grown in many orders. At the same time, in terms of functionality, structure and behaviour, apps themselves have grown more complicated. At the same time, the development and manufacturing cycles of these applications tend to become shorter because of factors like market pressure and fast development of supporting and enabling technologies. As a result, the expense of developing new appliances and production processes is increasingly shifting from producing new products to accommodating current ones. Knowledge of design, operations and behavior of existing fabricated artifacts, including software codes, hardware systems and mechanical assembly is one of the essential components of this activity. For example, in the software business, maintenance expenses are estimated at over 80 percent, and software comprehension accounts for upto half of the maintenance cost of the software product. It would be great to have tools to create UML (Unified Modeling Language) models from the source code automatically to simplify the software development process. Engineering reverse The source code software architecture offers software practitioners with a useful service. An significant potential for software development engineers are case tools implementing MDA and reverse engineering. MDA and reverse engineering are a vital element to make manufacturing space more efficient and productive.

Omer Salih Dawood (2017) In the paper process, software requirements have been transferred to UML diagrams. It demonstrates the significance of this process and examines many comparative field investigations. In order to know the state of play, using requirement management tools, knowledge of UML software development, frequently used UML schedules, and the methodology for generating UML schedules from requirements, a survey study related questionnaire is distributed around the world to many research groups and academies as well as to industry. The article emphasizes that significant research is required to achieve UML diagrams within the scope of NLP requirements and to generalize automated or semi-automatic processes for generation of UML diagrams from requirements.

Bassam Atieh Rajabi (2017) UML is extensively used for the study and design of software. UML diagrams are broken down into several aspects of problem domain modeling. It is extremely important to preserve the coevolution between these diagrams to be constantly updated to reflect software changes. Decades of study have led to a broad range of methods for the control of the coevolution of UML diagrams. These methods may be divided into direct, transformational, formal semantics or ways to representation of knowledge. Formal approaches like as Colored Petri Nets (CPN) are often used in the detection and management of artifacts coevolution. Despite abundant progress, more effort still needs to be done to improve the efficiency and precision of cutting-edge coevolutionary methods in the management of UML diagram modifications further. A review of the ways to preserve coevolution in UML diagrams is given in this study. This study suggested coevolution relationships and changes on UML diagrams and components of diagrams. In addition, coevolution and inconsistencies of UML diagrams are presently addressed with problems and problems to identify and fix.

Deva and Muhammad (2009) Tools designed to create UML models utilizing NLP models, produced UML models such as the user case diagram, analysis model class diagram, collaboration diagram, and design model. UML Model Generator from the UMGAR (UMGAR). They integrated both Rational Unified Process (RUP) with object specifying and ICONIX with most excellent NLP tools.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Methodology

A literature study is needed in order to provide a framework for comparison with the results obtained from similar UML-usage studies. The UML study leads us to create the following topics of research (RQ):

- RQ1: How helpful UML language is regarded?
- RQ2: How much CASE tools are utilized?

The final section of the questionnaire covers a series of questions relating to CASE's use of tools for software projects and the use of mechanisms like UML profiles, SysML, Business Process Modeling Notation (BPMN), the perceived utility of UML and UML use.

The software package for SPSS has also been used for statistical testing for the purposes of detecting statistical correlations. Parametric testing of Wilcoxon was used to compare real usage and use of UML diagrams.

Result and discussion

While the distributed questionnaire was not required, UML was understood by all participants. The study found a Bachelor degree of education in 47 percent of the population (Table I), a Master degree in 22 percent and a Ph.D. in 8 percent. Previously completed, graduate students were included in the "Bachelor's degree category".

Table 1: Educational Level

Educational Level	Frequency	Percent
Bachelor degree	42	47.2
Master degree	20	22.5
Ph.D.	7	7.9
Students	20	22.4
Total	89	100.0

The professional profile of participants in Table 2, which indicates that the population sample is very equally distributed in three main categories of employment: public sector, private sector workers, employees, students or unemployed. Naturally, many students work before graduation.

Table 2: Work Profile

Work Profile	Frequency	Percent
Employee in the private sector	31	34.8
Employee in the public sector	35	39.3
Self employed	18	20.2
Student	2	2.2
Unemployed	3	3.4
Total	89	100.0

In general, participants' software engineering experience was classified as novices (with less than two years' experience), moderates (from 2 to 5 years' experience) and specialists (having more than 5 years of experience). Table 3 indicates that 49% of participants may be considered experts in software engineering. The percentage is really higher, since 20 of the participants in this study had less than 2 years of experience and 10 of the 20 reported. This is important since it indicates that the sample in question is very competent to assess the use of UML in software engineering.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Table 3: Participant's years of experience in software engineering

Years of experience in software Engineering	**Frequency**	**Percent**	**Cumulative Percent**
Less than 2 years	28	31.5	31.5
From 2 to 5 years	17	19.1	50.6
More than 5 years	44	49.4	100.0
Total	89	100.0	

The participants' competence in UML has evaluated the number of software projects that participated in the previous 5 years (see Table 4). The findings show that every participant took part in about five projects and the overall number of completed projects is around 413 for all participants. In 190 projects, UML was used in the 412 projects, which means that 46% of the software development projects employ UML. "Respondents reported being involved in information technology over the average of 15 years (4.7 with UML) in 27 projects, on average, in the similar Dobing and Parsons investigations (about 6.2 with UML). In this survey, 46% were much above the 23% literary research report on the UML usage in software projects. This discrepancy shows that the usage of UML has increased significantly in recent years because the B.Dobing and J. Parsons study is based on the information collected between March 2003 and March 2004 and spans an average career of 15 years. The fact that (a) the Dobing and Parson survey lasts fifteen years from 2004, leads to a significantly lower percentage use since the UML was first implemented only for 1997 and (b) the results are not directly comparable in these two surveys, as these cover a completely different geographical area and a general population sample. This is also explained by how many UML projects have been used in the total number of software projects, in Table 5 in the second part. 34% of participants utilized UML just in software development projects from their study of these results, because in the last five years they had used UML in all their projects.

Table 4: Completed software project/project where UML was used

Number of CompletedSoftware Projects			**Number of projects where UML was used**	
	Frequency	Percent*	Frequency	Percent
None	9		21	23.6
Less than 5 projects	40	50.0	51	57.3
From 5 to 10 projects	21	26.2	13	14.6
More than 10	19	23.8	4	4.5
Total	89	100.0		

* Percentage values were calculated after excluding the "none" answer

With regard to the question "how do you assess the performance of these typical UML projects in which you were involved?," 60% or very many projects have been successful and just 13% of projects have failed. The other answers were impartial. With regard to UML, the majority of users have expressed positive opinions on the creation of rapid systems (75%), higher quality systems (73%) and reduced cost of systems (72 percent). In relation to the question "whether you think the UML is complex and hard to apply," 49% say that it is negative (unpleasant or relatively unanimous), 25% say it is positive (who agree on the statement), and 26% say it is non-partisan. Table 5 provides a detailed overview of the results. The second scale is the option "dispute," the third is the option "neutral," the four is the option "agree," and the final fifth is the option "agreement," a one to fifth scale has been used.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Table 5: The overall opinion on UML usage

	Agree		**Somehow Agree**		**Neutral**		**Somehow Disagree**		**Disagree**		**Median**
Results in higher quality systems	26	(29%)	34	(38%)	18	(20%)	8	(9%)	3	(3%)	Somehow Agree (4)
Improves our productivity when building systems	31	(35%)	29	(33%)	19	(21%)	8	(9%)	2	(2%)	Somehow Agree (4)
Enables us to build systems more quickly	25	(28%)	27	(30%)	24	(27%)	10	(11%)	3	(3%)	Somehow Agree (4)
Helps ensure that the final system meets the functional requirements.	38	(43%)	27	(30%)	17	(19%)	4	(4%)	3	(3%)	Somehow Agree (4)
Helps in ensuring that the clients and users understand what system analysts are proposing	23	(26%)	32	(36%)	20	(22%)	12	(13%)	2	(2%)	Somehow Agree (4)
Helps to estimate the size of the system to be developed	28	(31%)	31	(35%)	21	(24%)	7	(8%)	2	(2%)	Somehow Agree (4)
Helps to decrease the cost of enhancements to the system.	22	(25%)	23	(26%)	24	(27%)	15	(17%)	5	(6%)	Somehow Agree(4)
Improves the Communication within project	31	(35%)	36	(40%)	14	(16%)	6	(7%)	2	(2%)	Somehow Agree (4)
Assists developers in performing their job	35	(39%)	29	(33%)	17	(19%)	6	(7%)	2	(2%)	Somehow Agree (4)
Is complex and difficult to use	6	(7%)	16	(18%)	23	(26%)	25	(28%)	19	(21%)	Neutral (3)

CASE Tools

The computer-aided software engineering tools (CASE) enable an exceptionally easy development procedure. The overall cost of constructing the system is reduced when the available availability is high. Therefore, the component is responsible.

Table 6: Guidelines for CASE tools

Rating	Description
0	No CASE tools usage.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

1	IDE used.
2	IDE with documentation software.
3	IDE with documentation and diagram tools
4	CASE tools for design, development and testing.
5	Maximum use of CASE tools. Includes tools for configuration management & project management.

XMI/UML Support in CASE Tools

Our research first aims to find version combinations of different XMI and UML in various CASE tools such as Rational ROSE and ArgoUML accessible. Table 6 provides a tooling research results that demonstrate compatibility for the major industry standards UML versions 1.3, 1.4, 2.0 and XMI versions 1.0, 1.1, 1.2.

"Y" in the following tables implies the tool is supported by the UML/XMI and "N" combination, thus it is not a UML/XMI version combination The specified tool.

Table 6: Rational ROSE Enterprise Edition 7.0 withthe XMI Patch

UML/XMI	**1.0**	**1.1**	**1.2**
1.3	**Y**	**Y**	**N**
1.4	**Y**	**Y**	**N**
2.0	**N**	**N**	**N**

UML versions 1.3, 1.4, 2.0, and XMI versions 1.0, 1.1, 1.2 with Open-Source Argo UML 0.24 are available from Table 7.

Table 7: ArgoUML 0.22 and 0.24

UML/XMI	**1.0**	**1.1**	**1.2**	**Version**
1.3	**Y**	**Y**	**N**	**0.22**
1.4	**Y**	**Y**	**Y**	**0.24**
2.0	**N**	**N**	**N**	**0.24**

Conclusion

The research included all areas of software development and systems of all sorts. In Greece, IT professionals. The findings indicate that, while UML is limited in most software projects, UML was used for over half of the 413 projects reported in this study. Typically, the "Class Diagram," the "Use Case Diagram," is the most frequent UML diagram followed. The most rare diagrams are "package diagrams," "timing diagrams" and "machine diagrams." Findings indicate that UML is mostly utilized for analysis and design instead than for coding and testing. Though UML is often used, it is foreign to uses the UML extensions like SysML, XMI and many more (negative responses). Regarding the benefits of UML, UML users said that they can build their systems more quickly, that it leads to higher quality systems, lowers the cost of improving the system. Finally, users believe they will reuse UML in future projects and expect that UML will continue to be used in the years to come.

Our task is mainly to provide reusable UML diagrams via different UML modelling tools without losing any data. Leading ROSE and open source software standards are the representative tools used in this case study. Using an XMI parser with XMI Reader as well as XMI Writer, we can identify and modify data loss during model exchange. Preliminary results analysis shown that it is possible to enhance common XMI support of UML tools and XMI's ability to properly describe UML diagrams.

References

[1] Dr. Arvinder Kaur 2012," Systematic Review of Automatic Test Case Generation by UML Diagrams," International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 6, August - 2012 ISSN: 2278-0181

[2] Bassam Atieh Rajabi 2017," Current Issues in UML Diagrams Coevolution and Consistency Techniques and Approaches," Corresponding author. Email: bassamrajabi@gmail.com Manuscript submitted September 03, 2017; accepted November 24, 2017. doi: 10.17706/ijcee.2018.10.2.158-173

[3] Arturs Bartusevics 2015," Models for Implementation of Software Configuration Management," Procedia Computer Science Volume 43, 2015, Pages 3-10

[4] N. K. Sharma 2013," Study Of Approaches For Generating Automated Test Cases By UML Diagrams," International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 www.ijert.org Vol. 2 Issue 6, June - 2013

[5] Abdelgawad, A 2013, 'Low Power Multiply Accumulate Unit (MAC) for Future Wireless Sensor Networks', published in 2013 IEEE Sensors Applications Symposium Proceedings, pp. 129-132.

[6] Alia Ahmed Eleti & Amer R Zerek 2013, 'FIR Digital Filter Design By Using Windows Method With MATLAB', 14th international conference on Sciences and Techniques of Automatic control & computer engineering - STA'2013, Sousse, Tunisia, pp. 282-287.

[7] Kokila Bharti Jaiswal, Nithish Kumar V, Pavithra Seshadri & Lakshminarayanan, G 2015, 'Low Power Wallace Tree Multiplier Using Modified Full Adder', 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1-4.

[8] Kunjpriya Morghade & Pravin Dakhole 2016, 'Design of Fast Vedic Multiplier with Fault Diagnostic Capabilities', International Conference on Communication and Signal Processing, India, pp. 0416-0419.

[9] Kurniawan, Aditya & Harefa, Bina & Sujarwo, Surya. (2014). Unified modeling language tools collaboration for use case, class and activity diagram implemented with html 5 and javascript framework. Journal of Computer Science. 10. 1440-1446.

[10] M.H. Aabidi, Benefits of reverse engineering technologies in software development makerspace, ITM Web of Conferences 13, 01028 (2017)

[11] Deeptimahanti, Deva Kumar, and Muhammad Ali Babar. "An automated tool for generating UML models from natural language requirements." Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering. IEEE Computer Society, 2009.

[12] Bassam Atieh Rajabi, Current Issues in UML Diagrams Coevolution and Consistency Techniques and Approaches, International Journal of Computer Electrical Engineering, Volume 10, Number 2, June 2018

[13] Reder, A., & Egyed, A. (2012). Incremental consistency checking for complex design rules and larger model changes. Proceedings of International Conference on Model Driven Engineering Languages and Systems (pp. 202-218). Springer.

[14] Elaasar, M., & Briand, L. (2004). An overview of UML consistency management (Technical Report No. SCE-04-18). Carleton University, Canada.

[15] Gongzheng, L., & Guangquan, Z. (2010). An approach to check the consistency between the UML 2.0 dynamic diagrams. Proceedings of 5th International Conference on Computer Science and Education (ICCSE). IEEE.