UML AND ITS APPLICATIONS

Priyanka Makkar[1], Assistant Professor
Amity University Haryana
Dr. Sunil Sikka[2], Associate Professor
Amity University Haryana

ABSTRACT

The Unified Modeling Language (UML) is an important standard for developing software. As UML describes the real-time systems, it is very important to make a conceptual model and then proceed gradually. There are Some UML tools generate program language code from UML. UML can be used for specification, design and implementation of modern embedded systems. UML can also be used for modeling the system from functional requirements through executable specifications. This paper provides a background study of UML and its applications.

Keywords: UML, UML diagram, UML versions, UML application

1. INTRODUCTION

UML (Unified Modeling Language) is a standard language for constructing, visualizing, documenting and specifying the artifacts of software systems. Initially it was capture the behavior of non-software system and complex software. Modeling is seen as a way to better handle the growing complexity of software development by helping engineers to work at higher levels of abstraction. Model-driven development is supported by the UML [1]. There are 14 UML diagrams divided into two categories seven represents structural information and another seven represents dynamic behavior of system [2]. The rest of the paper is divided into three parts section-2 presents various versions of UML. Section-3 represents the various applications of UML in software engineering. Section-4 represents the conclusion and future work of the paper.

2. BACKGROUND OF UML

In the year 1997 UML 1.1 was introduced and adopted by the Object Management Group. After that in year 2000 UML1.1 undergo number of changes to the UML semantics, meta model and notation, but should be considered a minor upgrade to the original proposal and UML version 1.3 released [5]. It was further upgraded in year 2001 to the original proposal. Mostly "tuning" release but not completely upward this version is compatible with the UML 1.3. Addition of profiles as UML extensions grouped together. Stick arrowhead in interaction diagrams now denotes asynchronous call**.** Model element may now have multiple stereotypes**.** Refined definitions of components and related concepts are updated. Artifact was added to represent physical representations of components. New version was introduced that is UML 1.4.In the year 2003 UML version 1.5 was release with added actions - executable actions and procedures, including their run-time semantics, it also defined the concept of a data flow to carry data between actions, etc.

In the year 2005 when UML version 2.0 was release with new diagrams like object diagrams, package diagrams, composite structure diagrams, interaction overview diagrams, timing diagrams and profile diagrams were introduce. Collaboration diagrams were renamed to communication diagrams. Activity diagrams and sequence diagrams were enhanced. Activities were redesigned to use a Petri-like semantics.

Edges can now be contained in partitions. This version was accepted as ISO specification (standard) ISO/IEC 19501. UML 1.5 was released 2 years before. Partitions can be hierarchical and multidimensional. Explicitly modeled object flows are new. Classes have been extended with internal structures and ports (composite structures). Information flows were added. Collaboration now is a kind of classifier, and can have any kind of behavioral descriptions associated. Interactions are now contained within classifiers and not only within collaborations. It is now possible for use cases to be owned by classifiers in general and not just packages. New notation for concurrency and branching using combined fragments. Notation and/or semantics were updated for components, realization, deployments of artifacts. Components can no longer be directly deployed to nodes. Artifacts should be deployed instead. Implementation has been replaced by «manifest». Artifacts can now manifest any package able element (not just components, as before). It is now possible to deploy to nodes with an internal structure. New meta classes were added: connector, collaboration use, connector end, device, deployment specification, execution environment, accept event action, send object action, structural feature action, value pin, activity final, central buffer node, data stores, flow final, interruptible regions, loop nodes, parameter, port, behavior, behavior classifier, duration, interval, time constraint, combined fragment, creation event, destruction event, execution event, interaction fragment, interaction use, receive signal event, send signal event, extension, etc.

Later on in year 2006, 2007 and 2009 minor revision has been done in UML 2.0, UML 2.1.2 and UML 2.2 like corrections, consistency improvements and added clarifications to UML 2.1.2.

Year after year UML undergoes revision. In year 2010 and 2011 minor revision has been done in UML 2.2 and UML 2.3 like how to send and receive signal events, renamed destruction event to destruction occurrence specification, profiles - applied stereotypes and changed stereotypes to have upper-case first letter - «Meta class» and stereotype application.

In the year 2015 UML version 2.5 was release there are no longer two separate infrastructure and superstructure documents, the UML 2.5 specification is a single document. Within the specification no longer package merge is used. Four UML compliance levels (L0, L1, L2, and L3) are eliminated, as they were not useful in practice. UML 2.5 tools will have to support complete UML specification [3]. Information flows, models, and templates are no longer auxiliary UML constructs. At the same time, use cases, deployments, and the information flows became "supplementary concepts" in UML 2.5. UML 2.5 has a number of fixes, clarifications, and explanations added. They updated description for multiplicity and multiplicity element, clarified definitions of aggregation and composition, and finally fixed wrong «instantiate» dependency example for Car Factory. New notation for inherited members with a caret '^' symbol was introduced. UML 2.5 clarified feature redefinition and overloading. They also moved and rephrased definition of qualifiers. Default for generalization sets was changed from {incomplete, disjoint} to {incomplete, overlapping}. There are few clarifications and fixes for stereotypes, state machines, and activities. Protocol state machines are now denoted using «protocol» instead of {protocol}. Use cases are no longer required to express some needs of actors and to be initiated by an actor.

In December 2017 latest version UML 2.5.1 was released with minor revision to the UML 2.5 which is easier to read by increasing clarity and removing redundancy. Till now 14 UML versions are available.

3. APPLICATIONS OF UML

UML standard is very useful in requirements capturing, decomposing the system into objects and defining their relationships [4]. UML can be used for specification, design and implementation of modern embedded systems. UML can also be used for modeling the system from functional requirements through executable specifications and for that purpose it is important to be able to model the context for an embedded system – both environmental and user-driven [6]. With the help of UML models software engineers can detect bad

smells. Bad smells tend to have a negative impact on software by degrading its quality [7]. It is beneficial to detect model smells to avoid their propagation to later stages of software development.

4. CONCLUSION AND FUTURE WORK

In this paper major efforts are done to find out the different applications of UML which include requirements capturing, specification, modeling, detecting bad smells. However detailed work is required to study the various applications of UML in depth. Therefore the extension to the same study is targeted for future work.

REFERENCES

[1] G. Booch, J. Rumbaugh, and I. Jacobson, "The Unified Modeling Language User Guide." Addison Wesley, 1999.

[2] Priyanka Makkar, Dr. Sunil Sikka "Review of UML Tools and UML Based Software Metrics for Improving Software Quality" IJRASET,Volume 5 Issue V, May 2017.pp 1652-55

[3] Meenakshi sharma, Nasib S. Gill, Sunil Sikka "Survey of object-oriented metrics: focusing on validation and formal specification" ACM Volume 37 Issue 6, November 2012 Page1-5

[4] M. Genero, M Piattini, C. Calero: "A Survey of Metrics for UML Class Diagrams", in Journal of Object Technology, vol. 4, no. 9, November-December 2005, pp. 59-92,

[5] James Rambaugh, Ivar Jacobson, Grady Booch, Unified Modelling Language Reference Manual, The (2nd Edition) 2004

[6] Fairfax, Virginia, Designing concurrent, distributed, and real-time applications with UML

ICSE '01 Proceedings of the 23rd International Conference on Software Engineering page 737-738

[7] Haris Mumtaz, Mohammad Alshayeb, Sajjad Mahmood and Mahmood Niazi "A survey on UML model smells detection techniques for software refactoring" 01 February 2019