"ROLE OF USE CASE DIAGRAM IN S/W DEVELOPMENT"

Author
Dr.S.S.Mule
Head & Research Guide,
Department of Commerce,
Swami Vivekanand Mahavidyalaya,
Mantha, Tq – Mantha, Dist – Jalna.

Co-Author
Mr.Yashwant Waykar
Department of Management Science
Dr.B.A.M.University, Aurangabad – 431001
waykar.yashwant@gmail.com

Abstract – *Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. The research article focuses on identification of significance of use case diagram in software development. As per findings of this article we can easily conclude the significance of use case diagram is increasing as the size of the project increases. As the size of project may be directly related with the extent of complexity of project. Large & complex projects always need the help of use case diagram so that developers can easily understand the requirements of the system.*

Keywords –UML, use case diagram, actor, extends

Introduction **-**

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.[1]

UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously putting effort to make a truly industry standard.

- UML stands for **U**nified **M**odelling **L**anguage.[2]
- UML is different from the other common programming languages like C++, Java, COBOL etc.
- UML is a pictorial language used to make software blue prints.

So UML can be described as a general purpose visual modelling language to visualize, specify, construct and document software system. Although UML is generally used to model software systems but it is not limited within this boundary. It is also used to model non software systems as well like process flow in a manufacturing unit etc.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization UML is become an OMG (Object Management Group) standard.

Overview:

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, *dynamic behaviour* means the behaviour of the system when it is running /operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So to model the entire system numbers of use case diagrams are used.

Purpose:

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view.

So in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

How to draw Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases.

So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

The actors can be human user, some internal applications or may be some external applications. So in a brief when we are planning to draw an use case diagram we should have the following items identified.

- Functionalities to be represented as an use case
- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. So after identifying the above items we have to follow the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important. So the name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships. Because the main purpose of the diagram is to identify requirements.
- Use note when ever required to clarify some important points.

The following is a sample use case diagram representing the order management system. So if we look into the diagram then we will find three use cases (Order, SpecialOrder and NormalOrder) and one actor which is customer.

The *SpecialOrder* and *NormalOrder* use cases are extended from *Order* use case. So they have extends relationship. Another important point is to identify the system boundary which is shown in the picture. The actor *Customer* lies outside the system as it is an external user of the system.

Where to Use Case Diagrams?

As we have already discussed there are five diagrams in UML to model dynamic view of a system. Now each and every model has some specific purpose to use. Actually these specific purposes are different angles of a running system.

So to understand the dynamics of a system we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output and the function of the black box is known.

These diagrams are used at a very high level of design. Then this high level design is refined again and again to get a complete and practical picture of the system. A well structured use case also describes the pre condition, post condition, exceptions. And these extra elements are used to make test cases when performing the testing.

Although the use cases are not a good candidate for forward and reverse engineering but still they are used in a slight different way to make forward and reverse engineering. And the same is true for reverse engineering. Still use case diagram is used differently to make it a candidate for reverse engineering.

In forward engineering use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.

So the following are the places where use case diagrams are used:

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering. [1]

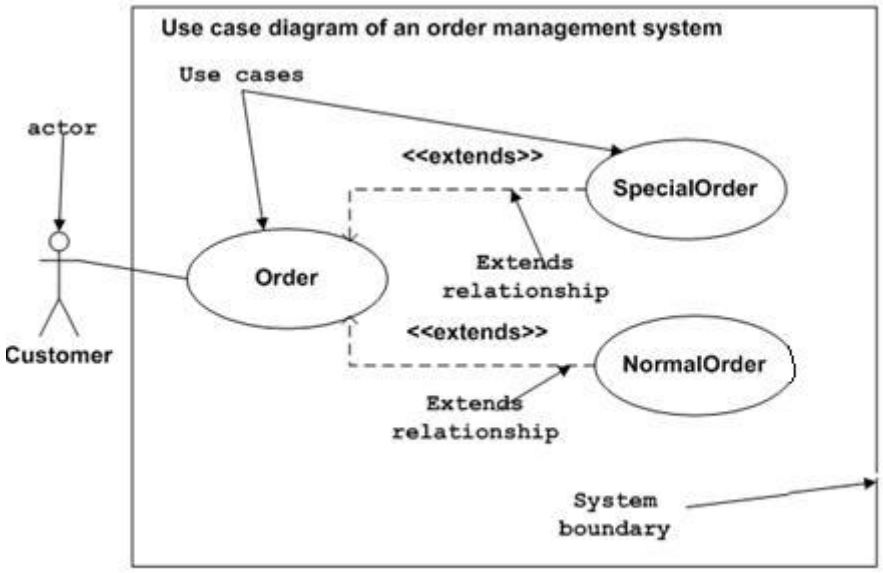

Figure: Sample Use Case diagram

Review of Literature –

The study of IritHadar and Orit Hazzan, Technion – Israel Institute of Technology, Israel *"On the Contribution of UML Diagrams to Software System Comprehension"* found that
"UML was utilized by the students as a multifaceted expression tool. The way in which the different teams sorted the diagrams in preparation for the comprehension process, the different pivotal diagrams that they leaned on, and the number of "visits" made to each of the different diagram types, all indicate that the process of comprehension and information extraction from UML diagrams varies between different people. It was also found that, when taken together, no one diagram type was globally less or more important than the others for the performance of the comprehension task. In other words, the differences in preference between the various teams cancelled out each other. The above conclusions are based on the work of senior computer science students."[4]

Another study of FahadAlhumaidanCollege of Computer Sciences and IT, King Faisal University, Hofuf, KSA on *"A Critical Analysis and Treatment of Important UML Diagrams Enhancing Modelling Power"*found that "Unified Modelling Language (UML) is used at initial phases of software development because of having a reasonable support of diagrams and notations but has not proved sufficient for the complete modelling of functional and non-functional requirements of a system. Based on our experience of applying UML, some weaknesses in the diagrams are identified in this paper and a treatment is presented. For example, most of the UML structures are based on graphical notations and are prone to causing errors. The hidden semantics under the diagrams allow ambiguities at design level and multiple notations produce inconsistent and ambiguous models. Further, the models described using UML diagrams may have multiple interpretations and the recipients of the design may not be able to understand what has been put in the diagrams. There exists some well-established approaches, for example for- mal methods, which can capture the semantics hidden under the UML diagrams. [5]

Formal methods are useful at all stages of software development because of having rigorous mathematical and computer tools support. However, at the current stage of development, formal methods are not sufficient in complete modelling of a system. In this way, UML and formal methods are both useful for design and specification of software systems but an integration of these approaches will facilitate the software development process."

The Study of Jos_eMerseguer, Javier Campos, Dpto. de Inform_atica e Ingenier__a de Sistemas ,Universidad de Zaragoza, Zaragoza, Spain on *"Exploring roles for the UML diagrams in software performance Engineering"* found that "It is not an overstatement to say that the gap between software design and performance evaluation techniques has caused the misuse of the last ones by softwareengineers. The UML profile for schedulability, performance and time arose from the intention to close fields,software engineering and performance analysis. Nevertheless the gap remains, since it is difficult for software engineers to devise which parts of their designs are suitableto represent performance requirements. The profile hasstarted to study this problem from a scenarios viewpoint.In this work, we explore other viewpoints to deal with performance requirements at software design level."[6]

The study of Mario KušekSašaDešićDarkoGvozdanović on "*UML Based Object-oriented Development: Experience with Inexperienced Developers"*found that UML offers many advantages. New projects havebetter survival prospects with UML. The experimenthas shown that UML has a very strong impact onnewcomers and that it can markedly increase theirwork and design capabilities.Utilization of UML tools can additionally improvesoftware design process. With these tools,documentation process can be included intodevelopment process, because the documentation iscreated during design time.Practical application of UML proved to be Successful. Requirement analysis and architecturaldesign benefit from UML. Additionally, UML toSDL translation will also strengthen the process of the systems' dynamic structure design.[7]

The study of Maria Kutar, Carol Britton and Trevor Barker ,*Department of Computer Science University of Hertfordshire, UK* . on*"Comparison of Empirical Study and Cognitive Dimensions Analysis in the Evaluation of UML Diagrams"* found thatThey have investigated whether there is a difference between a user's ability to extract informationfrom sequence and collaboration diagrams which form part of the UML modelling toolkit. The twodiagrams aresemantically equivalent but differ in their structural representation of information. they examined the diagrams using both a theoretical approach (analysis using the cognitive dimensionsframework) and a experimental approach (carrying out a study of users of the diagram). Their hypothesis was that the structure of sequence diagrams, which have a clear direction and explicitlyshow the sequential ordering of events, would be more easily understood by readers.The hypothesis was supported by the theoretical investigation, with analysis under several of thecognitive dimensions indicating that the difference in structure would provide a significant differencein user understanding. However, the second investigation did not yield the same result; here there wasno significant difference in performance according to diagram type.We believe that the difference in findings between the two investigations may be caused by any of thefollowing:

- Additional effort from users on tasks they perceive as more difficult
- User preference for a particular type of diagram
- Scenario effect
- Problems inherent in the CD's framework
- Problems inherent in the design of the study [8]

The study of Sunguk Lee, Research Institute of Industrial Science and Technology Pohang, Korea on *"Unified Modeling Language (UML) for Database Systems and Computer Applications"* observed that, the study focus on concepts of database systems as well as the overview of the use of Unified Modeling Language (UML) as a standard notation of real-world objects in developing object-oriented design methodology for computer applications. The UML is a tool for specifying software systems that include standardized diagrams to define illustrate and visually map or model a software system's design and structure. UML diagrams include the use case diagram, class diagram, sequence diagram, statechart diagram, activity diagram, component diagram, and deployment diagram. The integration of these diagrams to different software processes has been discussed.[9]

The study of Mohammad Ibrar, Department of Computer science, Metas Adventist College, Ranchi, India on *"UML Diagrams: an aid to Database Design Specification: A Review*" observed that UML is being used as the de-facto standard in the software industry. There are many tools used in the industry to develop information system. These tools provide the initial specification in UML that eventually leads to the database development.UML provide for conceptual, logical and physical database modeling design. Ofcourse it plays important roles in reverse engineering i.e., it allows the user to create a data model based on the already existing database structure. UML base tools can read the schema of the data models and create the database and the data storage model and generate appropriate code DDL code for the same.[10]

The study of VipinSaxena, and Deepa Raj on *"UML Modeling for Instruction Pipeline Design"* observed that UML Class modelis a powerful model used to depict the software developmentproblems and the hardware problems. In the InstructionPipelining Designing, the Internal Data Forwarding saves theexecution time of the block or group of instructions of theprocess. The present work is further extended by consideringthe different kinds of instructions executed in the differentkinds of the buffer known as sequential buffer, loop buffer,target buffer, etc. The present work is considered only for thelinear Instruction Pipeline Design therefore, the UML modeling application for Instruction Pipeline Designing can befurther extended for the non-linear Instruction PipelineDesigning.[11]

Need of the Study –

Unified Modelling Language is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development. During this different diagrams are drawn according to need & the types of requirement. The need of this study is to find out significance of use case diagram in s/w development. We have to observe The changing significance of use case diagram in different size of s/w projects.(e.g. small, medium , large, very large).

Research Methodology –

The survey was conducted among the developers, analysts, associate systems, Test engineer, senior subject expert, consultants that are working in different IT companies across the Pune region. A small questionnaire was prepared & near about 27 samples was collected in random selection manner across the IT companies. Also few interviews & informal discussions were made to identify the importance of use case diagram in s/w development.

Analysis & Results –

During the study a questionnaire was given to the IT persons (developers, analysts, associate systems, Test engineer, senior subject expert, consultants). These persons are working on different s/w development project & also have IT experience of 1 to 10 years. During their work they came across the different types of projects (web-portals, embedded s/w, products, desktop applications etc).

The basic assumption made for this research regarding the size / volumes of s/w are as follows:

Project	**Duration (s/w development time)**
small	1 to 6 month
Medium	6 Months to 1 year
Large	1Year to 2 Years
Very -Large	More than 2 Years

As a part of study they have to fill the questionnaire in which they have asked to select any one option out of 4 as per the importance of use case diagram.

Ex.

Use case Diagram –

1. Mandatory (most important)
2. Important
3. Less Important
4. optional

The points are assigned to each option of the above question.

1. Mandatory (most important) - 4 points
2. Important - 3 points
3. Less Important - 2 points
4. Optional - 1 point

Ex. If 10 respondents select options 1.Mandatory for class dia. then 10 X 4= 40 points will be considered.

Following Results are found during the study for use case diagram for different sized s/w:

1) Small Sized Projects –

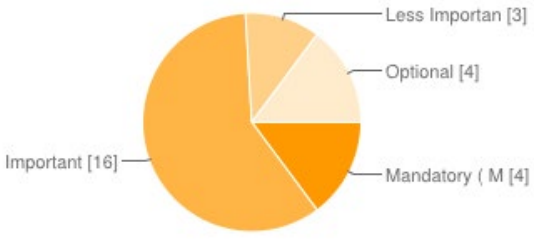

Fig 1. Responses for use case diagram

Table - I

Question Options	**No. Of Respondents (persons)**	**In % (27 as 100%)**	**Points**
Mandatory	4 Persons	15 %	4X4=16
Important	16 Persons	59 %	16X3=48
Less Important	03 Persons	11 %	3X2=6
Optional	04 Persons	15%	4X1=4
		Total Points	**74**

As per Table-I we can say that 59 % respondents feels that use case diagram is important diagram or it must be drawn. 15 % respondents feel that drawing use case dia. is optional for small projects or even though we won't draw it, it doesn't affect much.

2) **Medium Sized Projects** –

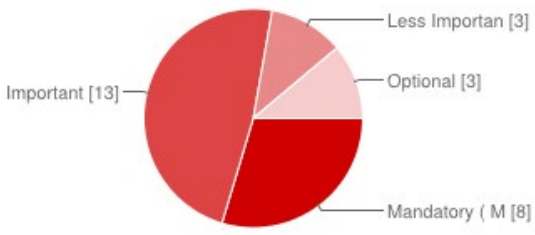

Fig.2 Responses for use case diagram

TABLE-II

Question Options	**No. Of Respondents (persons)**	**In % (27 as 100%)**	**Points**
Mandatory	8 Persons	30 %	8X4=32
Important	13 Persons	48 %	13X3=39
Less Important	03 Persons	11 %	3X2=6
Optional	03 Persons	11 %	3X1=3
		Total Points	**80**

As per Table-II we can say that 48 % respondents feel that use case diagram is important diagram or it must be drawn. Only 11 % respondents feel that drawing use case dia. is optional for Medium-sized projects or even though we won't draw it, it doesn't affect much.

3) **Large Sized Projects –**

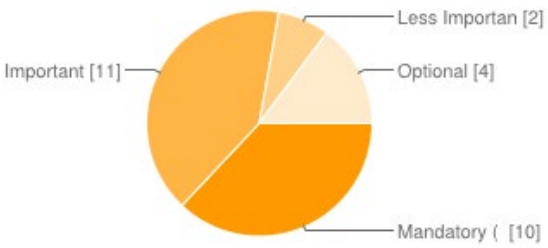

Fig.3 Responses for use case diagram

TABLE-III

Question Options	No. Of Respondents (persons)	In % (27 as 100%)	Points
Mandatory	10 Persons	37 %	10X4=40
Important	11 Persons	41 %	11X3=33
Less Important	2 Persons	7 %	2X2=4
Optional	4 Persons	15 %	4X1=4
		Total Points	**81**

As per Table-III we can say that 41 % respondents feel that use case diagram is important diagram or it must be drawn. Only 7% respondents feel that drawing use case dia. is less important for Large-sized projects.

4) **Very - Large Sized Projects –**

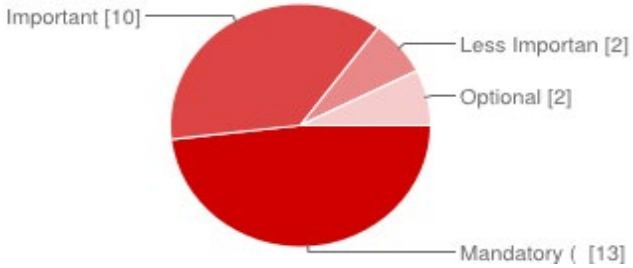

Fig.4 Responses for use case diagram

TABLE-IV

Question Options	No. Of Respondents (persons)	In % (27 as 100%)	Points
Mandatory	13 Persons	48 %	13X4=52
Important	10 Persons	37 %	10X3=30
Less Important	2 Persons	7 %	2X2=4
Optional	2 Persons	7 %	2X1=2
		Total Points	**88**

As per Table-I we can say that 48 % respondents feels that use case diagram is very important diagram or it must be drawn. Only 7% respondents feel that drawing use case dia. is less important for very Large-sized projects.

Conclusion –

From the study it has been observed that:

TABLE-V

Project	Total Points	% of Respondents Answered ***Mandatory***
Small	74	15%
Medium	80	30%
Large	81	37 %
Very-Large	88	48 %

From the TABLE-V, we can easily conclude the significance of use case diagram is increasing as the size of the project increases. As the size of project may be directly related with the extent of complexity of project. Large & complex projects always need the help of use case diagram so that developers can easily understand the requirements of the system.

During further research papers I would like to find out the importance of different UML diagrams in different category/types of s/w development.

References -

[1] "Introduction to UML diagrams" [online] available : *http://www.tutorialspoint.com/uml/*

[2] "Introduction to UML" [online] available: *http://www.uml.org/*

[3] "use case diagram " [online] available : *http://sourcemaking.com/uml/modeling-it-systems/structural-view/use case-diagram*

[4] IritHadar and Orit Hazzan, Technion – Israel Institute of Technology, Israel , *"On the Contribution of UML Diagrams to Software System Comprehension"* , Journal of Object Technology published by ETH Zurich , Vol. 3, No. 1, January-February 2004

[5] F. Alhumaidan, "A Critical Analysis and Treatment of Important UML Diagrams Enhancing ModelingPower,"*Intelligent Information Management*, Vol. 4 No. 5, 2012, pp. 231-237. doi: 10.4236/iim.2012.45034.

[6] Jose Merseguer, Javier Campos, Dpto. de Informatica e Ingenier__a de Sistemas , Universidad de Zaragoza, Zaragoza, Spain ,*"Exploring roles for the UML diagrams in software performance Engineering" , 2003*

[7] Mario KušekSašaDešićDarkoGvozdanović , "*UML Based Object-oriented Development: Experience with Inexperienced Developers" , 6th International Conference on Telecommunications +,MajaMatijašević and AlenBažant (eds.), June 13–15, 2001, Zagreb, Croatia*

[8] Maria Kutar, Carol Britton and Trevor Barker ,*Department of Computer Science University of Hertfordshire, UK , "Comparison of Empirical Study and Cognitive Dimensions Analysis in the Evaluation of UML Diagrams" ,* In J. Kuljis, L. Baldwin & R. Scoble (Eds). Proc. PPIG 14

[9] Sunguk Lee, Research Institute of Industrial Science and Technology Pohang, Korea on *"Unified Modeling Language (UML) for Database Systems and Computer Applications"* International Journal of Database Theory and Application Vol. 5, No. 1, March, 2012

[10] Mohammad Ibrar, Department of Computer science, Metas Adventist College, Ranchi, India on *"UML Diagrams: an aid to Database Design Specification: A Review*" , International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 3, March 2013 ISSN: 2277 128X

[11] VipinSaxena, and Deepa Raj on *"UML Modeling for Instruction Pipeline Design" ,* World Academy of Science, Engineering and Technology 18 2008

[12] C.R.Kothari , " Research Methodology – Methods & Techniques" , 2nd Edition New Age International Publishers.