UCaseNar: A Versatile tool for Specifying Narrative Use Cases

Tejas R. Shah[1]
[1]*Department of ICT, Veer Narmad South Gujarat University, Surat, India*
(E-mail: proftejas@gmail.com)

***Abstract*—**Requirement Engineering involves the different phases of eliciting, elaborating, modelling, prioritizing, negotiating, specifying the functional and non functional requirements of a system. The software industry has moved from structured system development to object oriented system development approach. Capturing software requirements from clients often leads to error prone and vague requirements documents. To conquer this issue, requirements engineers often choose to use UML models to capture their requirements. The diagrammatic use case not includes the narrative text and the textual document requires complex natural language processing. In this paper, a novel and versatile tool UCaseNar is presented which implements narrative use case modelling and specification through web and form based interface.

Keywords*—Requirement Engineering; UML (Unified Modelling Language; Use case; Use case tool*

I. INTRODUCTION

The crucial phase of software engineering is requirement engineering and all the sub sequent phases are dependent and worked parallel with the context of system requirements only. The requirement analysis and modelling phase models the requirements through various methodologies. There are various methodologies like OMT [1] and Yourdon [2] based on extensions and implementation of data modelling.

UML is considered as de facto standard in software development and used in many different domains. UML is a standard language for writing and software blueprints. UML is used to visualize, document, construct the artefacts of software. UML is appropriate for modelling systems which can be enterprise based or web based distributed systems.

The use case concept was introduced by Ivar Jacobson et al. [3] to begin the system development from what user wants. In this way, the system is built from the users' point of view. A use case specifies a set of associated usage scenarios where actors and the system interactions take place. A use case diagram shows a set of use cases, actors and their relationships as shown in Figure 1.

An actor consists of a person, a company or organization, a program, or a computer system. Actors of use cases are always stakeholders of the system, but not all stakeholders tend to be actors, because they never interact directly with the system.

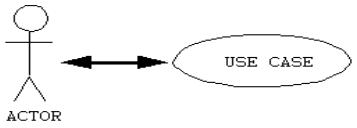

Figure 1 Primary Use Case Notation

Use cases can be written in simple text format, from use case brief, casual, outline, to fully dressed etc., and with miscellaneous templates. To get high quality requirements, the use of templates in use case is a regular industry practice. The template defined by Alistair Cockburn in [4] is one of the most frequently used writing styles of use case. The different styles may create the confusion to designers in understanding diagrammatic representation.

This narrative textual form of use cases are (legible requirement user stories), comprehensible by all stakeholders and complemented by visual UML diagrams encourage better and deeper communication of system. But, it is difficult to process natural language representation. In addition, the visual diagram is not including features like pre condition, post condition and trigger like elements.

The advantages, limitations and potential pitfalls of Use Cases are mentioned by many researchers [5][6][7].

A. The Benefits of Use Cases

- **Elicitation and Specification of Complex Requirements**

As a user-centred technique powerful technique, use cases helps to ensure that the correct system is build by eliciting the requirements from the user's perspective. The complexity of large projects can be reduced by use cases with decomposition of the problem into major functions (i.e., use cases) and by analysing applications from the users' perspective.

- **User Focused**

The use case tool is efficient and user-centric for the software requirements modelling and specification process. Use case modelling process starts from identifying key actors who interacts with the system by their goals to be satisfied by the system. These actor goals become the use cases which represent the desired functional requirements of the system.

- **Quality requirements by Narrative Use Cases**

To specify the structured, clear and unambiguous requirements, narrative use case facilitates step by step points of preconditions, post conditions, triggering event and system flow of the scenario. The elaborated, narrative and optimized use cases transforms system to better system interaction design and high end user experience.

- **Covering other aspects of software development**

The use case modelling can serve as a base for other aspect of software development as well, e.g. project planning, test case generation and documentation. To design a test case, well written and narrative use case serves as a basic development principles and valuable guidelines for test cases. The process of deriving functional test cases from a use case scenario is simple and efficient.

B. The Limitations of Use Cases

The use case modelling offers many important advantages and have become an essential component of object technology. However the misuse of use cases and issues related to diagrams leads to some limitations too.

- **Capturing Non Functional Requirement**

Use cases only represent the functional behaviour of system and not including how efficiently other features of system can be linked. The non-functional requirements such as security, privacy, performance are not to be modelled efficiently with use cases.

- **Lack of formality in natural language narrative use cases**

For defining use cases, practitioners are advised to turn to succinct and focused textual descriptions. There is no formal language for specifying the descriptions of the use cases. The changes of miscommunication and ambiguity can be increased if use cases are specified in natural languages. There is no formal definition of terms use case, actor, extends, and uses.

- **Inconsistency and Interpretation**

IT-oriented nature of use cases does not give any answers to the questions about their completeness and consistency. There may be conflicts among use cases and gaps that can be left in system requirements. Each project must form its own interpretation, as there is no standard definition of use case. Some use case relationships, such as extends and uses are ambiguous in interpretation.

- **Limitation of Diagrammatic Representation**

The concepts of preconditions, post conditions, triggers, and business rules properties shall be added to use cases for efficient analysis of system. The visual diagram notations do not easily handle the branches and loops in the narrative form. The relationship of use cases (uses, extends, includes) may create ambiguity in a diagram with different of level of abstraction. Without a clear purpose for a diagram, it is likely to end up a confused and aimless demonstration of notation.

Important requirements may be missed or taken incomplete because of simplified assumptions about the problem domain. But there are also inherent problems with use cases; in particular they are not equally suited for all kinds of requirements. Various studies have determined that eliciting system requirements and extracting their use cases can be gruelling and can lead to rather imprecise analysis [4] [5]. To overcome some of the issues like diagrammatic limitations, missing requirement repository and natural language complexity for processing narrative use cases, the UCaseNar tool is proposed.

II. RELATED WORK

There have been several areas like UML tool development, Use case mapping from requirements, extracting use cases from natural language documents.

In [8] authors have discussed the motivation for using essential use cases (EUC) to help model and structure textual natural language requirements. The work has also identified some of the problems faced by requirements engineers and end users while using the EUC approach and developed a prototype tool for automated tracing of abstract interactions. Requirements Use Case Tool (RUT) specified in [9] is a web-based tool that assists project team members to create, view, and modify use cases and requirement database repository for a particular project. To achieve the consistency, this valuable tool facilitates a standard use case template for all the use case entry into the repository.

In [10], a domain independent tool called UML Model Generator from Analysis of Requirements (UMGAR) is described. This tool generates UML models like the Use-case Diagram, class model from natural language text requirements using Natural Language Processing (NLP) techniques. UMGAR implements a set of syntactic re-enactment rules to process complex and large requirements into simple requirements.

The tool mentioned in [11] aims to develop a tool to assist novice system analyst in classifying UML diagrams. This RMTool tool is web based objected oriented modelling tool and it is tested with students based on questionnaire for effectiveness and usefulness.

The following table 1 compares the tool with other tools available in literature. The features which are considered for comparison includes web based support, narrative inclusion of use cases, requirement repositories, System and sub system linking, robustness of tool, diagrammatic features NLP support, ease of use, non functional requirement and many more.

It can be seen from above tools that very few tools are supporting narrative use case modelling in web based environment. Some of the tools are very efficient in NLP of requirements. However the NLP requires the complex processing of natural language statements to be converted into use cases and classes of the system. In addition to that some

of the tools do not support use case repositories. The other tools are partially supporting some features and some are technically inclusive in the tool. So compared to existing use case tool, UCaseNar form based elicitation tool is distinguished by several features to model the narrative use case.

TABLE 1 COMPARISON OF UCASENAR TOOL WITH OTHER TOOLS

Features	RUT	EUC tool	UMGAR tool	RMTool	UCaseNar tool
Web based	√	≠	≠	√	√
Diagrammatic Features	≈	√	√	√	≠
GUI Interface Support	√	√	√	√	√
Database of Use case	√	≠	≠	√	√
Narrative Properties of use case	≠	≠	≠	≠	√
Report Generation	≠	≠	≠	≠	√
System and Sub System Linking	≠	≠	≠	≠	√
Non Functional Requirement	≠	≠	≠	≠	≈
Robustness	√	√	√	√	√
NLP Support	≠	√	√	≠	≠
Ease of use	√	√	√	≈	√

√ Full Support ≈ Partial Support ≠ No Support

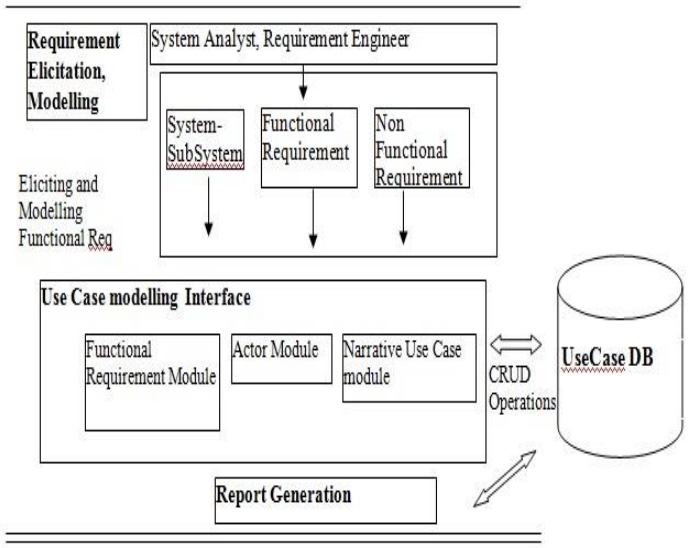

Figure 2 UCaseNar tool System

There are main 3 components of UCaseNar tool as show in figure 2. First component mentions the functional, non functional requirements by taking input from the system analyst and requirement engineer. The web based GUI modeling interface contains different modules like, system, sub system detail, functional and non functional requirement, actor and narrative use case properties. Second component is the interaction with Use Case database which interacts with all the modules and facilitates the CRUD (Creation, Read, Update and Delete) operations on use cases.

The report generation facility searches the requirement based on non functional features and provides detail regarding all use cases specific to a system. The editor is implemented using JSP in net beans IDE. The database of use case properties are created in MySQL.

III. ARCHITECTURE AND IMPLEMENTATION OF TOOL

A. Components of tool

UCaseNar is narrative GUI based use case modeling tool which supports state of the art functional requirements analysis and modeling. UCaseNar is implemented in Java and uses web based architecture to model use cases. The forms are created with the help of JSP technology. This tool is not providing graphical drawing feature to draw the Use case diagrams, but provides powerful form based editor to manage the requirements, use cases, actors and various artifacts of use cases.

B. Database design of tool

The database repository of tool as shown in figure 3 is created in MySQL. The 7 tables describe the schema of system, requirement and different properties of the use cases. The database schema provides the interaction between use case scenarios, connection with the system, requirement metadata and non functional attributes selection and narrative properties of the use cases. Very few tools support database oriented form based management of use case artifacts of requirements.

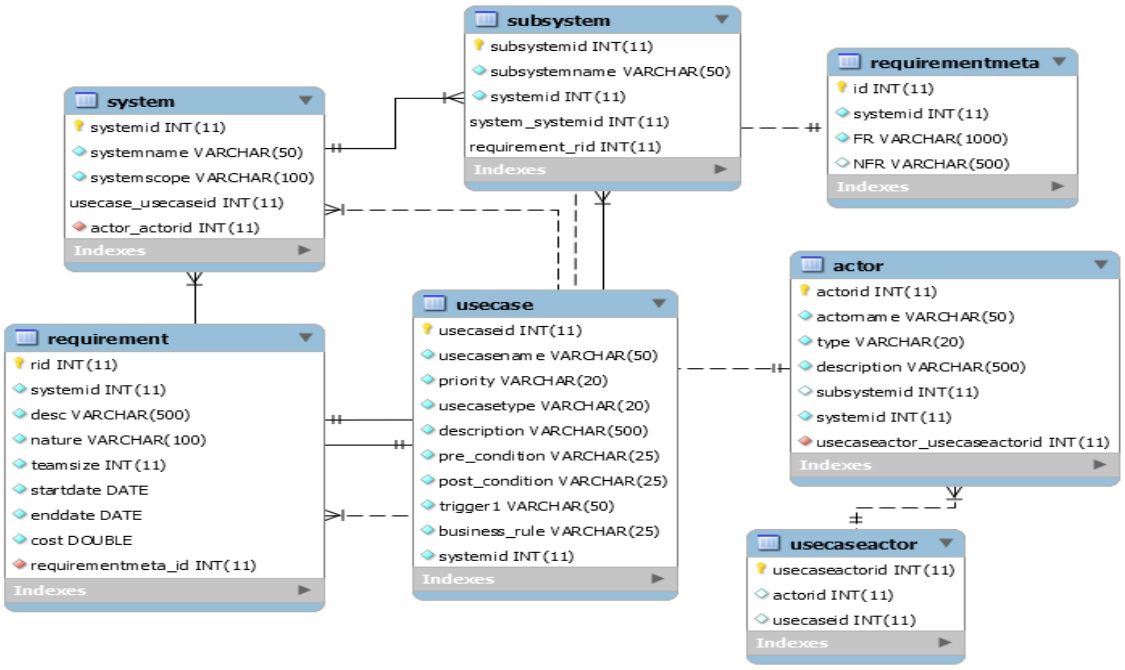

Figure 3 Database of UCaseNar tool

C. *Interface of tool*

The different forms are shown in Figure 4, 5 for requirements, NFR and narrative properties of use cases. It supports elaborative properties of use cases including pre condition, post condition, trigger as narrative part. In addition, this tool is not for high level graphical diagrammatic representation of use cases and other diagrams. The NFR properties like security, privacy, validation are provided in objective manner to be incorporated with functional requirements. It supports more fine-grained actor to use case relationship with types of actors and their roles.

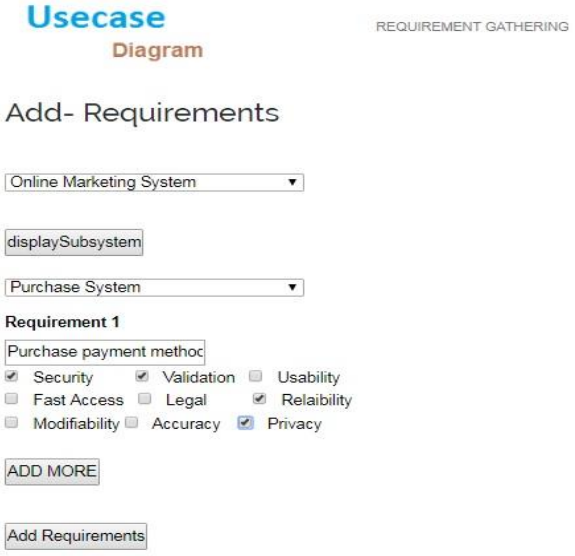

Figure 4 System and NFR of tool

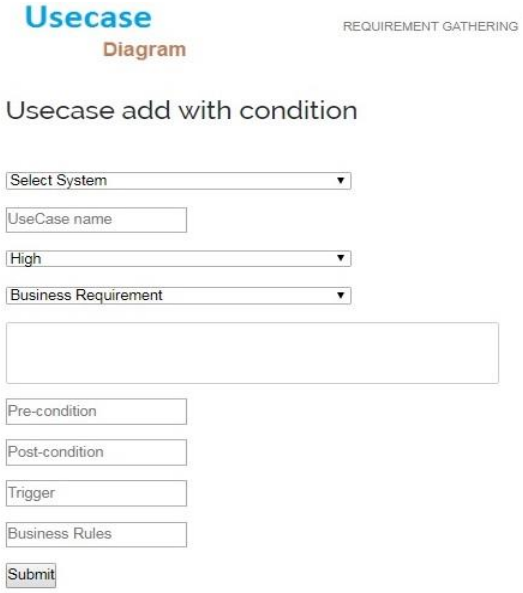

Figure 5 Narrative Use Case Module

The journal information system is considered as demonstration for narrative use case modeling. The report shows the project information, sub system detail, requirement and actor details. The use case tab shows the sample use cases: review paper and submit paper with all the narrative properties like priority, pre condition, post condition, trigger, business rules etc. The case study output is given figure 6 to include the overall image from system to minute properties of use cases. The validation and testing is completed to verify the linking of use cases with requirements.

Journal Info System

Project Information

#	Description	Nature	Team Size	Start Date	End Date

Sub Systems

#	Available Sub Systems
1	Manage Paper

Requirements

#	Functional Requirements	Non - Functional Requirements
1	Paper SUBMISSION	Security
2	Paper SUBMISSION	Relaibility
3	Paper SUBMISSION	Privacy

Actors

#	Actor	Type	Description	Sub System
1	Author	Primary	Primary Author for regi	Manage Paper
2	Review Paper	Primary	Reviewing Paper for Analysis	Manage Paper
3	Review Paper	Primary	Reviewing Paper for Analysis	Manage Paper

Use Cases

#	Use Case Name	Priority	Use case Type	Description	Pre-Condition	Post-Condition	Trigger	Business Rule
1	Submit Paper	High	Business Requirement	Submitting a paper for review	Registration	Successful Paper ID	Paper Uploading	Word format
2	Review Paper	High	Business Requirement	Reviewing Paper by reviewers	Paper Access	Successful Reveiw	Paper Recevied	Comments of review

Figure 6 Report-UCaseNar tool

IV. CONCLUSION

Requirement modeling and specification is one of the vital tasks to specify, manage and document the clear, complete and unambiguous requirement. The requirement repository is an efficient way to preserve and maintain functional, non functional requirements and narrative use case properties. The diagrammatic representation and textual based use case modelling results into complex and rigorous processing of requirements. This paper presented a web based tool UCaseNar to perform various operations on use cases through efficient form based interface. The tool is tested and validated for different fields pertaining to requirements and use cases. In future, other modeling diagrams can be added to support activities, classes and object interaction.

REFERENCES

[1] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and B. Lorensen, Object-oriented modeling and design. Prentice Hall, 1991.

[2] P. Coad and E. Yourdon, Object-Oriented Analysis, 2nd ed. Prentice Hall, 1991.

[3] I. Jacobson, Object-oriented software engineering : a use case driven approach. ACM Press, 1992.

[4] A. Cockburn, "Structuring Use cases with goals," Journal of Object Oriented Programming, vol. 10, no. 7, 1997.

[5] S. Lilly, "Use Case Pitfalls: Top 10 Problems from Real Projects Using Use Cases."

[6] D. G. Firesmith, "Use Cases: the Pros and Cons," no. August 1995, 2004.

[7] "Use Case." [Online]. Available: https://en.wikipedia.org/wiki/Use_case. [Accessed: 21-Aug-2018].

[8] M. Kamalrudin, J. Grundy, J. G. Hosking, and J. Hosking, "Tool support for essential use cases to better capture software requirements SEE PROFILE Tool Support for Essential Use Cases to Better Capture Software Requirements," in IEEE/ACM international conference on Automated software engineering, 2010, pp. 255–264.

[9] J. R. Mccoy, "Requirements Use case Tool (RUT)."

[10] D. K. Deeptimahanti and M. A. Babar, "An Automated Tool for Generating UML Models from Natural Language Requirements," in IEEE/ACM International Conference on Automated Software Engineering, 2009.

[11] E. Rosi Subhiyakto, S. aplikasi pemodelan, D. Wahyu Utomo, K. Kunci-RMTool, R. Persyaratan, and P. Lunak, "RMTool; Sebuah Aplikasi Pemodelan Persyaratan Perangkat Lunak menggunakan UML," J NTETI, vol. 6, no. 3, 2017.