International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Ameliorated Methodology to Represent UML use Case Diagram into Table Format

R. N. Kulkarni, C. K. Srinivasa

Abstract - The UML is a defacto tool used by the software industry for designing the software system. The UML comprises 13 diagrams which are used to represent static and dynamic behavior of the system. The usecase is one of the UML diagram which is used to describe the interaction between the user and the system. In this paper, we are proposing an automated tool to translate the UML diagram and its artifacts into the text format and is represented in form of a table. This translation is carried out in two stages.

1: Converting the UML diagram into intermediate text form using generic tool called WhiteStar UML.

2: The intermediate text is converted to table form using the tool proposed in this paper.

The output achieved at the end of the second stage is further used for the abstraction of predicate logic.

Keywords: functionality, requirements, software design, Use Case Table, Use case diagram.

I. INTRODUCTION

Unified Modeling Language (UML) is a standard Language for specifying, designing, viewing, and documenting software systems. The UML usecase diagram is used to describe the functionality of a system in a horizontal way. The idea of usecases to describe functional requirement was introduced in 1986 by Ivar Jacobson, the main contribute to the UML. Usecases are functional requirements that indicate what the system will do. Developing the usecase diagram is important task because it represents a transition between requirement of a system and the design of a system. This use case diagrams has four major elements i.e. the actor, system, usecase and relationship between elements. In this paper, we are proposing a new methodology which facilitates the tool that translates the usecase diagram which is in pictorial form into XMI format further the required information and relationship is abstracted from XMI format and it is represented in form of a table Nowadays we find different approaches in the software engineering literature to identify the usecases. The usecase view captures the behavior of the system or part of the system. The use case diagram comprises actors, set of use cases enclosed in the system boundary, association between the actors and the usecases, relationship among the usecases and generalization among the actors and use cases within the software systems.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Dr. R. N. Kulkarni* Prof. & Head, Dept. of Computer Science & Engineering BITM, Ballari, India. rn_kulkarni@rediffmail.com

C. K. Srinivasa Associate Prof., Dept. of Computer Science & Engineering BITM, Ballari, India. srinivasck9@gmailmail.com

II. LITERATURE SURVEY

UML is a general-purpose modeling language, it provides modeling concepts and an intuitive graphical notation for modeling various application areas, enabling a software system to be specified, designed, visualized, and documented. A UML Diagram provides a view of that part of reality described by the model. There are diagrams that express which users use which functionality and diagrams that show the structure of the system but without specifying a concrete implementation [1].

In paper [2], the author discussed about a methodology to transform the UML Class diagram in a standard arrangement and additionally records out the relationship between classes in a table form. In our proposed research work the behavior of UML Use case diagram is abstracted in a Table form.

In paper [3], the author represented the entire system of programs as control flow and data flow graph in form of table format which depicts the relationship between the attributes. In this paper we abstract the relationship of UML use case and represented them in form of a table.

In paper [4], the author developed a reengineering methodology that automatically abstracts the interrelationships between group of attributes and actor's interface, functional dependences etc. In this paper we abstract relationships between UML usecases and actor's in form of a table. In the [5] paper, the author proposed a process of transforming user stories into use cases. This transformation facilitates the extraction of actors, usecases and their associations relationship. In this paper we abstract the associations and relationships of UML use case diagram in a table form. In the [6] paper, the author presented an approach that takes brief description of the requirements in terms of UML use case diagram, and semi automatically generates the drafts of the detailed description of the requirements. In this paper we propose a semiautomatic tool that retrieves usecase relationships and associations from the UML usecase diagram which are similar to a usecase and represented in a table form. In paper [7], the author presented an approach to compare the effect of different UML diagrams properties in accessing the similarity between software designs. In our proposed research work we developed a semiautomatic tool that abstracts the similarity of UML use case diagram in the form of a table.

III. TERMINOLOGY

Usecase: Usecase is a collection of related success and failure scenarios that describes actors using a system to support a common goal.
Usecase Model: This is the set of all usecases that represents the systems functionality and its environment.
Usecase Instance: It is a particular story of using a system consists of sequence of actions and interactions between the actors and the system.
Scenario: A scenario is a step by step interaction in a time order between the actor and system.
Actor: An actor is a person or a system which either consumes the data or generates the data.

IV. METHODOLOGY

In this paper we have made an attempt to translate the UML use case diagram in a table format. The sequence of activities involved in the translation is depicted in figure 4.1 below. The various relationships such as association, extend, include, generalization is explicitly represented in the table.
In this paper we are using readily available tool called White Star UML tool for the retrieval of the Actors, use cases various Relationships such as includes, extends and generalizations. The proposed tool proposes the input data and generates the output which is presented in excel sheet for further processing. The sequence of steps during the process of translation are explained below.

4.1 Converting the UML UseCase diagrams to its equivalent XML document.

The UML UseCase diagram is drawn using white Star UML tool a rational Rose approach and is exported to XMI format. The generated XMI for use case diagram is manually validated for correctness and completeness. The proposed step of converting the UML usecase diagram into XMI format is discussed in section 5.

4.2 Finding and mapping the elements of XMI document to appropriate field in a table.

In this step we have proposed a tool which takes the output of step 1 as input and it abstracts relevant contents from the XMI document and then represented in the form of a table.

4.3 Block diagram for Representing UML UseCase Diagram to Table Form

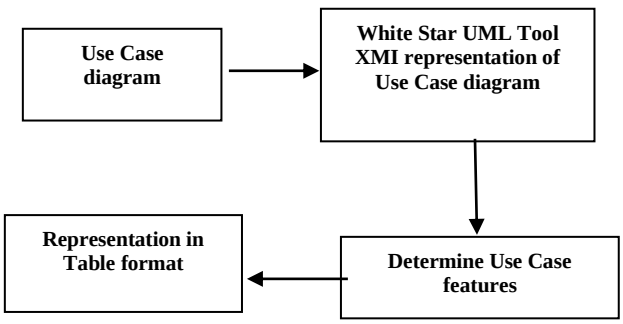


Figure 4.1: Block diagram to convert UseCase diagram to Table.

The above block diagram shows the procedure to convert UML UseCase diagram to Table form. Initially a UseCase diagram is drawn using White star UML tool and is exported to XMI form. This equivalent XMI Use Case form is given as input to our proposed tool (Use Case Table Generator). Our proposed tool validates the XMI tags and identifies the Use Case features. The Use Case features such as Actors, use cases and Relationships such as include, extend and generalizations are abstracted. The abstracted information of Use Case is mapped and is displayed to a Three column Table form. In this Use case Table, the association between Actor and usecase is represented as D-Usecase. The Include relationship between UseCases is represented as I-Usecase and the Extend relationship between UseCases is represented as E-UseCase. The Generalization for usecase is represented as GU-Usecase and GA-Actor for actors.

Algorithm for Representing UseCase Diagram to Table Form

```
INPUT:  UseCase Diagram
OUTPUT: UseCase -Table
STR[1..n] ← File-Name.xml
for j UseCase ← 1 to UseCaseList do
includes ← IncludeUsecases.ToList(); includesCount++
extends ← ExtendUsecases.ToList();
extendsCount++
children ← ChildUsecases.ToList(); childrenCount++
parents ← ParentUsecases.ToList(); parentsCount++
if includes! = null && includesCount > 0
for j Includes ← 1 to includesCount do
print usecase.name ←"D-" + usecase.Name;
index ← x.Name
 if x.Name == inc.Name
   UseArray[index] ←inc.Name
   Print include ← "I-" + inc.Name;
endif
endfor
endif

if extends != null && extendsCount > 0
  for j extends ← 1 to extendsCount do
     index ← x.Name
     if x.Name == ext.Name
       UseArray[index] ←ext.Name
Print extend ← "E-" + ext.Name;
endif
endfor
endif
if children != null && childrenCount > 0
     for j child ← 1 to childrenCount do
       index ← x.Name
        if x.Name == child.Name
     UseArray[index] ←child.Name
      Print child ← "GU-" child.Name;
 endif
endfor
endif
if parents != null && parentsCount > 0
for j parent ← 1 to parentsCount do
 index ← x.Name
if x.Name == parent.Name
 UseArray[index]
←parent.Name
```

Print parent ← "GU-" parent.Name;
endif
endfor
endif
endfor

V. RESULT AND DISCUSSION

The proposed procedure is implemented for various types of UML UseCase diagram and the results we got are correct and complete. The sample UML UseCase diagram figure 4.2 is the input and Table 1 is the output of our methodology. In figure 4.2 the UML UseCase has two Actors, three UseCases and two extended UseCases. The two Actors are namely as Cellular Network and User and UseCases are named PlacePhoneCall, Receive PhoneCall and UserSchedule, and extend relationships are PlaceConferenceCall and Receive AdditionalCall.

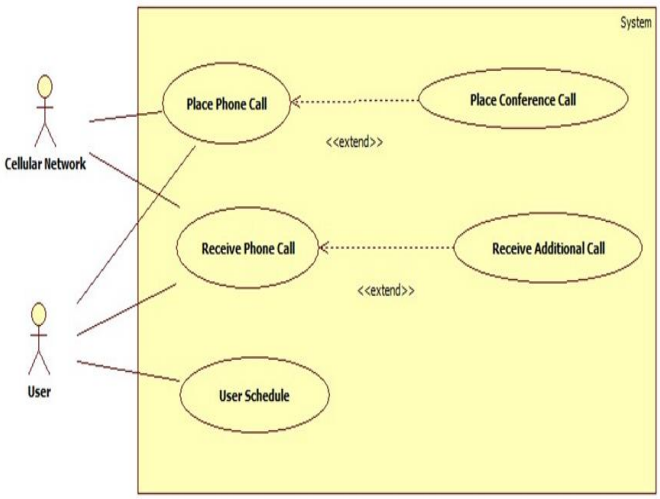


Figure 4. 2: UML Use Case diagram.

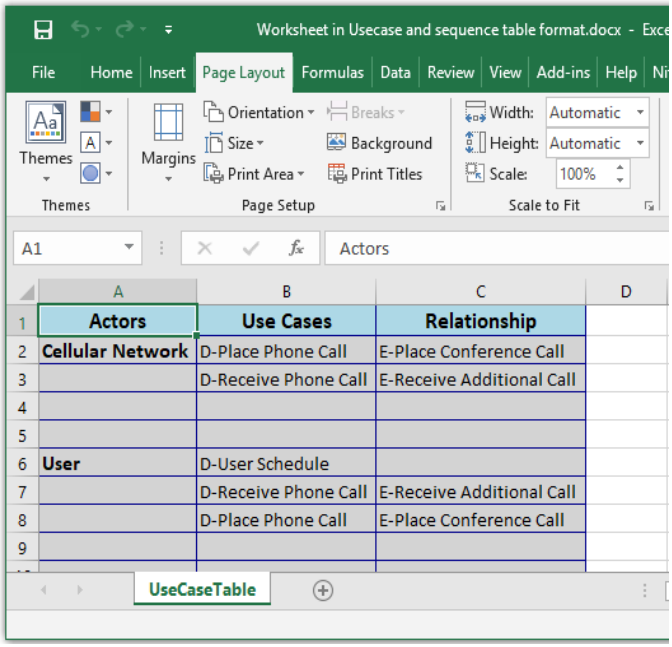
Step 1:
The first step of our proposed work is to export UML Usecase diagram to XMI form using White Star UML Tool. The following XMI code is generated by White Star UML for an UML Usecase diagram figure 4.2.

<UML:Model xmi.id="UMLProject.1">
<UML:UseCase xmi.id="UseCase.10" name="Place Phone Call"/>
<UML:UseCase xmi.id="UseCase.11" name="Receive Phone Call "/>
<UML:UseCase xmi.id="UseCase.12" name="User Schedule" />
<UML:UseCase xmi.id="UseCase.13" name="Place Conference Call "/>
<UML:UseCase xmi.id="UseCase.14" name="Receive Additional Call"/>
<UML:Actor xmi.id="Actor.17" name="Cellular Network"/>
<UML:Actor xmi.id="Actor.18" name="User"/>
<UML:UseCase xmi.id="UseCase.11" name="Receive Phone Call "/>
<UML:UseCase xmi.id="UseCase.12" name="User Schedule" />
<UML:UseCase xmi.id="UseCase.13" name="Place Conference Call "/>
<UML:UseCase xmi.id="UseCase.14" name="Receive Additional Call"/>
<UML:Actor xmi.id="Actor.17" name="Cellular Network"/>
<UML:Actor xmi.id="Actor.18" name="User"/>

Step 2:

In this step the required contents of an UML Usecase diagram are abstracted from the XMI format and represented in form of a Three column table by our proposed tool. Table 1 is generated output and is equivalent to figure 2 which contains all the Usecase features and is analogous to matrix.

Table 1: Output of UML Usecase diagram figure 4. 2 generated by our proposed tool



	A	B	C	D
1	**Actors**	**Use Cases**	**Relationship**	
2	**Cellular Network**	D-Place Phone Call	E-Place Conference Call	
3		D-Receive Phone Call	E-Receive Additional Call	
4				
5				
6	**User**	D-User Schedule		
7		D-Receive Phone Call	E-Receive Additional Call	
8		D-Place Phone Call	E-Place Conference Call	
9				

VI. CONCLUSION

This paper proposes a semi-automatic tool which takes the UML usecase diagram as an input and then generates the output in a three-column table format. This Representation of UML usecase diagram is carried out by converting use case diagram into its equivalent XMI format and then abstracting the information from XMI and is presented in a table format. This tool orders all the UML use case standard features such as Actors, use cases, Includes, Extends and generalizations, which systematizes with no consequences. It helps to quickly visualize relationships and further it is used to abstract requirement specification.

REFERENCES

1. Martina Seidl "UML @ Classroom", Springer International Publishing Switzerland 2015, DOI 10.1007/978-3-319-12742-2.
2. Dr. R. N. Kulkarni, C. K. Srinivasa "An Ameliorated Approach to Represent UML Class Diagram in the Table Format "International Journal of Computer Applications (0975 – 8887) Volume 182 – No. 7, August 2018.

3. Dr. Shivanand M. Handigund, Dr. Rajkumar N. Kulkarni, "An Ameliorated Methodology for the Abstraction and Minimization of Functional Dependencies of legacy 'C' Program Elements ". International Journal of Computer Applications (0975 – 8887) Volume 16– No.3, February 2011.
4. Dr. Shivanand M. Handigund, Dr. Rajkumar N. Kulkarni "An Ameliorated Methodology for the design of Object Structures from legacy 'C' Program", ©2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 13.
5. Meryem Elallaoui, Khalid Nafil, Raja Touahni, "Automatic Transformation of User Stories into UML Use Case Diagrams using NLP Techniques", Elsevier, 8th International Conference on Ambient Systems, Networks and Technologies 2018.
6. S. Paydar and M. Kahani, "A semi-automated approach to adapt activity diagrams for new use cases," Elsevier, Information and Software Technology, vol. 57, pp. 543-570, 2015.
7. Alhassan Adamu, Wan Mohd Nazmee Wan Zainon. "Multiview Similarity Assessment Technique of UML Diagrams", Elsevier 4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia.
8. Dr. Shivanand M. Handigund A. M. Shivarama, An Ameliorated Methodology for the Abstraction of Object-Oriented Features from Software Requirements Specification. The 2015 International Conference on Soft Computing and Software Engineering (SCSE 2015).
9. Dr. R. N. Kulkarni and Padmapriya Patil, Abstraction of Information Flow Table from a Restructured Legacy 'C' Program to be amenable for Multicore Architecture. 5th International Conference on Innovations in Computer Science and Engineering (ICICSE) © Springer, LNCS 2018.
10. Doug Rosenberg and Matt Stephens "Use Case Driven Object Modeling with UML: Theory and Practice", Copyright © 2007 Distributed to the book trade worldwide by Springer-Verlag New York, ISBN-13 (pbk): 978-1-59059-774-3.
11. https://sourceforge.net/projects/whitestaruml/ referred on 11th Feb 2019.

AUTHORS PROFILE

Dr. R. N. Kulkarni holds a Ph.D in software Engineering from Visvesvaraya Technological University, Belagavi, Karnataka, India. He has more than 30 years of teaching experience and published nearly 65 publications in International journals/conferences and also national conferences. His area of research work are Software Engineering, DBMS, Soft Computing and Object Technology. He is presently working as Prof. & Head of Computer Science & Engineering Department at Ballari Institute of Technology & Management, Ballari,Karnataka, India.

C. K. Srinivasa has completed M.Tech in Computers Science and Engineering from Kuvempu University Karnataka, India. His teaching interests are in the areas of Computer Programming, computer graphics and Object-Oriented Analysis and Design. His research interests and work are in the areas of Software Engineering. He is Currently working as Associate Prof. in Computer Science & Engineering Department at Ballari Institute of Technology & Management, Ballari, Karnataka, India.