CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Recently, the object-oriented methodology has been more popular and widely used. Its usage starts from the first level of software development lifecycle, known as the analysis level until the last level which is called the maintenance level. One of the object-oriented methodologies is called Unified Modeling Language or UML (Booch, et al. 1997), has now become a standard language in specifying object-oriented software.

The UML has its own notations and provides multiple views of a software system. It is a very useful modeling language in designing software. It acts as an interface between the natural language of user requirements and the programming language of implementation. It is also a communication medium between software designers and programmers. Under Object Management Group (OMG), UML has been released in many versions (Miller 2003). Even though OMG has produced a white paper on latest UML version which is version 2.0 in 2003, it seems that people still come out with different types of diagram when it comes to drawing a UML diagram for the same requirement (or case study). Hence,
this is the main point that gives an idea for this research to develop a tool in order to help users to draw a correct use case diagram.

Since the important part of UML is its facilities in drawing a use case diagram (Eriksson and Penker 1998), this research is only focused on the use case diagram. In UML, use case diagram plays an important role in the design level where it clearly visualizes all requirements that have been gathered. According to the UML specification (Object Management Group 2001), a use case diagram is a diagram that shows the relationship among actors and use cases within a system. A use case diagram is a diagram that helps analysts to discover the requirements of the target system from the user’s perspective. It describes what a system does from the standpoint of an external observer. Use cases are employed during the analysis phase of a software project to identify and partition system functionality.

The title of this project is named as “Development of a Text to Use Case Diagram Tool” or “T-UCD”. T-UCD is developed to help and guide students in identifying the elements of use case diagram from a short case study and subsequently draw a use case diagram based on the case study. T-UCD which acts as a tutor is designed to guide the student, where he or she is only required to follow all the steps to draw a correct use case diagram. It is hoped that the T-UCD will become a stepping stone to its next advance project to overcome problems in drawing a use case diagram.
1.2 PROBLEM STATEMENTS

Determining use case elements from a case study and drawing use case diagrams are not easy tasks for students who have limited knowledge in specifying and classifying requirements. They will face problems to draw a correct use case diagram. Thus, several important aspects in drawing use case diagram should be acknowledged and an in-depth research is needed in order to understand their problems. Based on a survey conducted through observations and research, below are the identified problems:

i. Students have problem in identifying a suitable actor and use case from the case study.

Based on the discussion with lecturers who teach the UML diagram subject, it is found that students are very weak in identifying use case elements. Although students do not have any problems with theoretical concepts of UML diagram, they often fail to draw a use case diagram correctly when a complex case study is given. Hence, lecturers concluded that the major problem of the students is due to vague understanding of the case study.

ii. Students always confuse in using “extend” and “include” relationship in drawing a use case diagram.

From the discussion, it is also found that students have a problem in using extend and include relationship. This situation happens because of different of explanations in various UML versions which results to confusion among students. For example, a number of UML versions used uses while other versions used include relationship (Ohst, et al. 2003).
iii. Current UML CASE tools are built only for expert users of UML

Many UML CASE tools have been developed for drawing UML diagram. A few examples are System Architect, Visual Paradigm, Smart Draw, Event Studio, and Argo UML. These systems provide helpful features to draw UML diagram for intermediate and expert users. However, until date not many systems are developed to help beginner users who have limited knowledge of UML diagram in order to use the system to its fullest capacity. Furthermore, most of these current tools are difficult to use and apply. These systems allow users to draw diagrams using the drag and drop method. Therefore, they are only suitable for expert users in drawing a use case diagram because requirements such actor, use case, and the relationships of their case study are already known. However, this is not the best technique for beginner users due to problems in identify suitable actors and use case from their case study. To overcome this matter, T-UCD is developed to identify the actor and the use case from the given case study.

iv. None of an efficient and effective system that provides guidelines and assists beginner students who are lack of knowledge in UML to draw a use case diagram.

Until the time of writing, there have not been any current tools developed to teach the beginner users or students with limited knowledge in drawing use case diagram. To draw a use case diagram, a number of guidelines must be followed to achieve a correct diagram. Normally, students would not spend a lot of time to explore these guidelines in text books and they would also assume that all of these guidelines are explained in these system.
1.3 PROJECT SIGNIFICANCE

The project will provide a tool as a solution to minimize problems in drawing a use case diagram as previously explained in section 1.2 (Problem Statement). It will benefit students who have limited knowledge in UML diagram during learning process. Students, who have problems in identifying suitable use case elements from a case study, will find this tool very practical for them. It will also ensure that correct elements of the use case diagram are being identified correctly.

Apart from that, this project is designed to help students with limited UML knowledge capacity to learn more on use case diagram. It provides some notes, and guidelines in drawing a use case diagram along with a detail explanation on differences of include and extend relationship. In addition, it also provides an example of situation in using this relationship.

1.4 PROJECT OBJECTIVES

The main objective of this project is to provide a tool as a solution to help students in identifying use case elements and help them to draw a correct use case diagram. In order to accomplish the objective, three sub objectives are considered.

i. To analyze problems in drawing a use case diagram among students.

Drawing a use case diagram cause a problem for those who have limited knowledge in UML and requirements specification. To draw a complete use case diagram, elements of the use case diagram which are actor, use case, and relationship should be initially identified correctly from the case study. To achieve this project objective, a discussion has
been conducted with Madam Siti Hafizah Ab Hamid. She is a lecturer of the UML subject in the Faculty of Computer Science and Information Technology University of Malaya. The discussion is based on students’ case study answers, where it is found that the students encounter trouble in drawing a correct use case diagram.

**ii. To implement a tool that is capable to help users in identifying use case elements from a case study.**

Use case diagram has four major elements which are: the *actors* that the described system user interacts with; the *system* itself; the *use cases*, or services that the system knows how to perform, and the lines that represent *relationship* between these elements. These elements are needs to be correctly identified from the case study before use case diagram is drawn. From study and analysis on problems in drawing use case diagram among students have found that most of the students have a problem in identifying suitable elements of use case diagram if a complex case study is given by their lecturer.

**iii. To develop a tool that is capable to help users in drawing a use case diagram.**

T-UCD is a system that is capable to help users draw a use case diagram. T-UCD is very different with current UML CASE tools (SmartDraw, System Architect, Event Studio, ArgoUML) because the main focus for T-UCD for beginner users of UML. T-UCD will draw a use case diagram based on the use case elements which is identified from the case study entered by users. T-UCD also acts as a tutor for the user. Notes and guidelines are provided in T-UCD to help users draw a use case diagram.
1.5 RESEARCH QUESTIONS

In order to achieve the objectives defined in section 1.4, some research questions have been identified. This research questions are included in the questionnaire survey. A survey was conducted where it involved 200 respondents. The research questions of this work are:

i. Among students, what are the current problems in drawing use case diagram?

ii. What are the features and concepts of the new system that will help users in drawing use case diagram?

iii. What can the new system contribute to beginner users in drawing and handling problems in relation to use case diagram?

1.6 RESEARCH METHODOLOGY

This research uses V-Shaped Model as a process model for the system development. The project problem statements, research questions, objectives, and scope of the project have been defined to obtain the project definition. In order to get the feedback, this research uses triangulation, a mixed method of quantitative and qualitative approaches. In qualitative method, focus group discussions had been conducted while for quantitative method, findings from the questionnaire designed in collaboration with the focus group had been explored. In order to design the questionnaire, Goal Question Metric approach has been referred. In combination of the questionnaire survey analysis and the focus group discussions, the study aims arrived at a more nuanced understanding of the nature of the use case diagram tools among students and system developer.
During preliminary design, entity relationship diagram, class diagram, interaction overview diagram, data flow diagram and graphical user interface (GUI) for the system were produced. In detailed design, a system module which consists of the notes modules and tutorial modules were developed. In the implementation phase, modules of the system were coded.

In order to achieve the project objective, testing was done on T-UCD. In this phase, all defects and fault found were corrected. During the component integration, real time test was conducted on the full version system and finally, in the system integration phase, a black box testing was carried out. For the system measurement, Line of Code and object-oriented metrics which are Weighted Method Class, Depth of Inheritance Tree, and Number of Children have been measured.

1.7 PROJECT SCOPE

On completion, this project does not only need to meet and accomplish the objectives but the project scope must also satisfy the requirements to avoid unexpected outcomes of the selected topic. The main scope of this project is to implement a tool that can convert a paragraph of requirements to use case diagram. The application will cover the steps involved in drawing a use case diagram. The applications are:

i. Provide notes and guideline in relation to draw a use case diagram.

ii. Identify use case elements from the case study entered by user.

iii. Automatically draws a use case diagram based on the text entered by user.

iv. Allow user to save the diagrams.
1.8 PROJECT LIMITATIONS

T-UCD only focuses on the use case diagram. Other UML diagrams such as class diagram, sequence diagram, collaboration diagram, and state chart diagram are not applied in this system but will be include in the future work.

1.9 TARGET USER

The target users for this system are college or university students who are learning UML subject, system designers and system developers. Those who are in the process of developing software will benefit a lot from this system. Although they may have experienced UML either manually or by using the CASE tools, this system would provide a better and faster approach for this process.

Students can use this system to help them design their software. The system is of immense help for the students during the development of a system for their thesis. In most cases, students working their thesis have not learned much on UML. With that little knowledge and experience, T-UCD will save development duration compared to having to learn how to use the available CASE tools.

1.10 EXPECTED RESEARCH OUTCOMES

The main contribution of this project is to provide a use case diagram tool that can help students in drawing a correct use case diagram. The purpose is to decrease the problem among students in identifying a use case diagram’s elements. The tool will draw a diagram based on requirements (case study) entered by the user. The expected project outcomes are:
To produce a report that presents the results of use case diagram among university students and system developers.

To produce a system that can guide user step-by-step in how to draw a correct use case diagram from a case study given by their lecturer.

1.11 TERMINOLOGY

Below are terms used throughout the project that needs to be understood prior to reading this dissertation.

i. **UML**

The UML, an object-oriented modeling language is an acronym for Unified Modeling Language. The UML provides a standard notation for specifying software systems. It provides notations for modeling static, dynamic, use case, activity, interaction and other views.

ii. **Use Case Diagram**

Use case diagram describes what a system does from the standpoint of an external observer. The emphasis is on what a system does rather than how a system functions. Use case diagram depicts use case, actor, associations and system boundary.

iii. **Use Case**

A use case describes a sequence of actions that provide something of measurable value to an actor.
iv. **Actor**

An actor is person, organization or external system that plays a role in one or more interactions with the system.

v. **Associations**

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

vi. **Relationship**

A group of data items organized in columns and rows, also known as a table is known as relationship.

vii. **Aggregation**

A whole-part association between two or more objects, where one represents the whole and the others parts of that whole.

viii. **Generalization**

The abstraction of common features among elements (for example, classes) by the creation of a hierarchy of more general elements (for example, super classes) that encapsulate the common features.

ix. **System Boundary**

System boundary is the rectangle box around the use cases. It is to indicate the scope of the system. Anything within the box represents functionality within scope and anything outside the box is not.
1. Use Case Description

Use case descriptions are simply less formal descriptions of the use cases that contain all the information needed to produce the formal use case diagrams.

11. User Requirement

Something that users require a software system to do (functional requirement); alternatively, a standard for the performance of a system (non-functional requirement).

12. T-UCD

T-UCD stands for Development of a Text to Use Case Diagram Tool. It is the name given for this project.

1.12 PROJECT SCHEDULE

Having a simple methodology to follow in this project development was not enough to complete T-UCD sufficiently. In order to build a complex system, many software engineering tasks occurred in parallel, and the results of work performed during one task might affect on work to be conducted in another task. These interdependencies are very difficult to understand without a schedule. It is also virtually impossible to assess progress on a moderate or large software project without detailed schedule. To achieve this milestone, a project schedule was planned and analyzed carefully based on the scope of the work.
In this project, a timeline chart or a Gantt chart was generated. Figure 1.1 illustrates the T-UCD Gantt chart.

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Title review</td>
<td>14 days</td>
</tr>
<tr>
<td>2 Information listings</td>
<td>102 days</td>
</tr>
<tr>
<td>3 Project registration</td>
<td>2 days</td>
</tr>
<tr>
<td>4 Proposal preparation</td>
<td>35 days</td>
</tr>
<tr>
<td>5 Literature review</td>
<td>120 days</td>
</tr>
<tr>
<td>6 Methodology</td>
<td>21 days</td>
</tr>
<tr>
<td>7 Information analysis</td>
<td>90 days</td>
</tr>
<tr>
<td>8 System requirement analysis</td>
<td>37 days</td>
</tr>
<tr>
<td>9 System analysis</td>
<td>53 days</td>
</tr>
<tr>
<td>10 UI Design</td>
<td>31 days</td>
</tr>
<tr>
<td>11 Coding</td>
<td>110 days</td>
</tr>
<tr>
<td>12 System testing &amp; evaluation</td>
<td>22 days</td>
</tr>
<tr>
<td>13 System modification</td>
<td>30 days</td>
</tr>
<tr>
<td>14 System installation</td>
<td>7 days</td>
</tr>
<tr>
<td>15 Documentation</td>
<td>500 days</td>
</tr>
<tr>
<td>16 Report writing</td>
<td>4 days</td>
</tr>
</tbody>
</table>

**Figure 1.1: Gantt chart**

There are fifteen tasks within six groups of activities in system schedule. The six groups are system planning, system analysis, system design, system development, system evaluation and system finish.

In the system planning activities, a project review is important to have a good understanding on the system to be developed. The understanding ensures project development is always within the project scope and on track way to achieve project objectives. In the system analysis activities, the collected information is used as a data to initiate system development. In this stage, questionnaires have been distributed and focus group discussion was conducted. After gathering the information, a number of important analyses were made on the data collected. At this level the use case diagram is constructed and it will be a reference to rely the system design level.
In system design activities, much focus relies on data design, architectural design, component design and interface design. During this process, entity relationship diagram, class diagram, data flow diagram, interaction overview diagram, component diagram were developed. All of these diagrams are used for implementation in the next level, which is system development level.

System development or code generating level is an activity that translates what had been written or designed in the system design level into machine-readable form or coding. At this stage the actual system was developed. System testing is an activity where the system is tested by logical test and functional test. Activity in system finish only begins after all subsystems are fully accomplished. The activities included final testing and installer making.

1.13 ORGANIZATION OF PROJECT REPORT

This report is organized into eight related chapters. Chapter 1 presents the introduction of this project. It starts by presenting the project definition, project motivation, project objectives, project scope and limitations, target audience, project expected outcome, terminology and organization of this project. The last section is the conclusion of this project.

Chapter 2 provides literature review on UML diagram, use case diagram and user education in drawing use case diagram. It also includes other project initiated by other researcher with similar approach and methods but with varying objectives. Discussion and comparison with current UML CASE tools in drawing use case diagram are also covered in this chapter.
Chapter 3 describes the system development strategies, which include the methodologies used in the research and development of the system. The V-Shape model is employed to develop this system. This chapter also focuses on the research techniques used to capture user requirements, results and analysis carried out during this research.

Chapter 4 describes the system requirements which include functional requirements and non-functional requirements which are adopted for this research. This chapter also describes the development tools specification which includes related software requirements and hardware requirements.

Chapter 5 describes the design issues involved in developing the T-UCD, which include database design, system design and system model using the UML, database design and the GUI of the system.

Chapter 6 discusses the system implementation, which comprise of the coding algorithm for functional requirements and the explanation module used in the system.

Chapter 7 contains testing for T-UCD. It consists of unit testing, integration testing, system testing and acceptance testing. These several tests are conducted to ensure that T-UCD is error-free.

Chapter 8 discusses the strength of T-UCD, limitations and future enhancement suggestions for this project. In addition, objectives achieved, system strengths before a conclusion is made are discussed.
1.14 SUMMARY

There is a great need to produce more research on issues involved in drawing use case diagram. From the research, it is found that many issues arise in drawing use case diagram which occur to both within students and system developers. A big challenge in confronting these issues is to make the user understand use case diagram. A lot of effort is required in educating these users on theoretical and practical issues of the use case diagram. Thus, this project is one of the possible approaches used to educate user on drawing use case diagram and to produce a system that is capable to guide user in the drawing process. Besides, the system can also help the user in identifying the use case elements from a given case study. It is important to help user in identifying the use case elements so that they know how to draw a correct use case diagram.