CHAPTER 1

Introduction

Since 1980s and into the 1990s, much research in software development problems has been done. The researchers have focused on automated software engineering tools, which are called Computer Aided Software Engineering (CASE) tools [1]. A CASE tool is a software used to support system development effort such as requirements capture, organization of documents, code, testing, and etc. According to Vessey [1], the CASE tools will completely replace the software developers in the system development process. In a more realistic view, these tools will help system developers in the process of specifying, designing, modifying and constructing the software systems.

In this thesis, the adaptation of meta-modeling to develop a CASE tool of UML (Unified Modeling Language) will be discussed. UML is recognized as a visual modeling language used to specify, construct, visualize and document the artifacts of a software system. It provides a full specification of the meta-classes that is needed for creating a modeling language which is appropriate for Object-Oriented Analysis and Design (OOAD) approach. This UML meta-model is in turn founded on a higher level modeling language and has an associate concept structure [2]. This concept level is called meta-metamodel level.

A meta-metamodel defines the language for specifying metamodels. A meta-model is an instance of meta-metamodel. The main role of the meta-model is to define the language for specifying a model. A model is an instance of a meta-model. It defines a language to describe an information domain.

The meta-metamodeling can be described as a process of creating a model at a higher level of abstraction than the things that had being modeled [3]. Based on the concept of meta-metamodeling, all the techniques of UML will be defined. A technique is a defined way of manipulating the concepts of a model. As such, these techniques will be defined and automated

by using the meta-modeling approach. The techniques of UML that will be defined are, *Use Case diagram, Class diagram, Sequence diagram, Component diagram, Deployment diagram, State diagram, Activity diagram, and Collaboration diagram.*

This documentation is organized in five chapters comprising the overview in Chapter 1 and followed by the definition and literature review which are related to meta-modeling in Chapter 2. The notation, syntax and semantics of each UML technique will be discussed and elaborated in Chapter 3. In Chapter 4, the discussion is about the meta-modeling of UML techniques using the concept of meta-model. The final chapter will discuss the evaluation and performance of UMLCASE, the conclusion of the project and suggestions for future enhancement.

1.1 Objectives

The main objectives of this research are as follows:

- To develop a CASE tool for UML by using the meta-metamodeling approach. The CASE tool will be able to support eight techniques of UML.
- To have a full understanding of the concepts of meta-metamodel. The concept will be used in defining the meta-model of UML.
- To be able to identify the syntax and semantics of the UML techniques and then apply the concepts of meta-metamodel to model these techniques.

1.2 Scope

The scope of this research project covers two parts as follows:

- The first part is about defining and developing a CASE tool called UMLCASE to support the UML techniques using a metaCASE tool, called MetaEdit Personal 1.2. The meta-modeling of these techniques is based on the OPRR (Object, Property, Role and

Relationship) concepts of the meta-metamodel of MetaEdit 1.2. These techniques are Use Case Diagram, Class Diagram, Activity Diagram, State Diagram, Collaboration Diagram, Sequence Diagram, Component Diagram and Deployment Diagram.

- The second part is about upgrading the techniques of UML into the MetaEdit+ environment that is a higher and more advanced version of MetaEdit. The meta-modeling of these techniques is based on GOPRR (Graphic, Object, Property, Role and Relationship) concepts of the MetaEdit+.