2016 Third International Conference on Information Retrieval and Knowledge Management

Integration of Use Case Formal Template using Mapping Rules

Swee Yin Wong[1], Edwin Mit[2], Jonathan Sidi[3]
Faculty Computer Science and Information Technology
Universiti Malaysia Sarawak,
94300 Kota Samarahan, Sarawak, Malaysia
[1]cyinwsy@gmail.com, [2]edwin@fit.unimas.my, [3] jonathan@fit.unimas.my

***Abstract*—The integration of use case and formal specification plays an essential role in addressing the issue of gaining the rigor and reliable software model such as formal model via easy and economic model such as object model. Although extensive research has been carried out on this integration, however there is a huge challenge on bridging the gaps between natural language used in use case scenario and the mathematics model used in formal model. This is mainly because of the differences in syntax and semantic of these two models. Natural language requirement is well documented that it is being inconsistent, inherently ambiguous, and incomplete even though natural language is universal, widespread, and flexible. As a consequence, it may lead to misunderstanding and produce an incorrect and inaccurate analysis and design model. Therefore, this paper aims to propose a use case formal template and define a new set of mapping rules that is used for formalizing UML use case by transforming use case scenarios which are written in natural language into VDM++ formal specification. The formal verification for the generated VDM++ formal specification can be further conducted by adopting the existing support tool of VDM++ (i.e. VDM++ ToolBox) to verify the correctness of the specification.**

Keywords-integration; use case; Vienna Development Method; formal specification; formal template; mapping rules

I. INTRODUCTION

Use case diagram in Unified Modeling Language (UML) is an essential tool for capturing user requirement. This is a high level model that bridges the gap between user and developer so that they can get a common understanding of the software model [1]. However, majority of the use case specifications are described in natural language such as English as semi-formal or informal structured text. It is well documented that natural language requirement is being inconsistent, inherently ambiguous, and incomplete even though natural language has the benefits of universal, widespread, and flexible [2]. In [3], the researchers also showed that the less accuracy in specification is main caused by the inherent ambiguity of natural language. Thus, the ambiguity and flexibility of natural language may lead to misunderstanding among developers, domain experts and end users of a system. As a result of this, it will produce incorrect and inaccurate analysis and design model and this will lead to software failures or rework cost. In particular, error cannot be accepted in critical and safety systems because any errors in those systems will involve catastrophic loss.

Besides that, software analysis model is often represented by using use case model during analysis phase, one of the early stages in Software Development Life Cycle (SDLC). All requirements should be produced in that phase as it is the most important phase in SDLC. Any error during the analysis phase will lead to the wrong analysis, design, and implementation models, which is the main reason for the software failure. Meanwhile, some critical decisions that should be done in analysis phase are also always deferred by developers until the design phase and implementation phase. As a consequence, a lot of re-design cost may be caused and the design time is lengthened.

In addition, based on Shen and Liu [4], they stated that a lot of misunderstanding and confusion among developers and end users can be reduced through formalism of use cases in a high level and such can be very beneficial in improving software quality. In work [5], Bakri et al. also stated that formal method is the optimum technique in error reduction especially at the earlier stages of software development. Furthermore, it is a well acknowledged fact that formal methods have been also quite successful in the area of uncovering ambiguities, incompleteness, and inconsistencies in requirements representation [6]. At the same time, by using formal specification, it can be very effective in improving system comprehensibility, reliability, and design time [7].

To deal with the ambiguity in requirements, UML use case has to be formalized into a rigor and reliable software model such as formal specification that use mathematical notation in describing software requirement description precisely with minimum confusion and ambiguity. In this research, VDM++ (Vienna Development Method ++) formal specification is chosen to formalize UML use case because it is executable and able to support both object-oriented concept and concurrency control. VDM has also a proven track record in industrial application [8], [9]. At this current stage, there is still a youthful and active Overture research community in supporting VDM [9]. Unfortunately, there is a formalism gap between UML use case and VDM++ due to their different syntax and semantics. This is a huge challenge to bridge the formalism gap. Hence, this work is to propose a use case formal template and define a new set of mapping rules for formalizing UML use case scenario written in natural language (i.e. English) into VDM++ formal specification. Through this formalization, a precise, complete, unambiguous, and consistent software specification is expected to be produced and such can improve the software quality, reliability, comprehensibility.