

Applying SOFL to Construct Requirements Specification for Examination Monitoring System

A.R. Mat, M.A. Khairuddin, A.B.Masli, M.N. Jambli, and E. Mit

*Faculty of Comp. Science & IT, University Malaysia Sarawak
Kota Samarahan, MALAYSIA*

{marahman, kmasyraf, bmazman, jmnazim, edwin}@fit.unimas.my

Abstract—SOFL approach to the construction of the formal specifications has been proposed and applied in information systems, but its effectiveness and applicability have not been demonstrated aggressively. In this paper, we describe an application of the SOFL approach to the construction of a specification during requirements analysis. This approach requires two steps: informal and semi-formal steps. In order to describe on how this approach can be applied to capture requirements using SOFL easily, we use a case study to develop an examination monitoring system for construct abstract requirements. This case study demonstrates the suitability of SOFL to capture detail requirements and provides us with an insight into the knowledge of how SOFL approach can be effectively supported.

Keywords—SOFL; specification; requirements analysis; formal engineering method; examination monitoring system;

I. INTRODUCTION

It is hardly to understand the specification if it is not well presented to a client. It can be interpreted wrongly and might produce incorrect design specification. The analyst also needs to know how to get the essential information from the client. The question is how to capture a requirement effectively in a specification. One way is by using a traditional method such as Waterfall model [1], iterative model [2], UML [3], and spiral model [4]. Another possible way is by using formal method approach, such as a VDM [5], B Method [6], and Z method [7]. System analyst might also use Structured Object-Oriented Formal Language (SOFL) [8], to represent the specification to the client. SOFL, which is a formal engineering method, provides a comprehensible language and method for both requirements and design specifications to develop software systems. In this paper, we concentrate on the approach for capturing an abstract requirement specification.

SOFL has been developed and applied in information systems over the last eighteen years. In particular, to construct a specification for requirements analysis, two steps are required: informal and semi-formal. While the informal step is to gather all the required functions, data resources, and constraints into one specification in natural language, the semi-formal step is to group and transform the expression into SOFL language except logical expressions (i.e. type invariants, pre- and post-conditions). In this paper, we present a case study applying SOFL to construct a requirement specification for an

examination monitoring system. This paper will focus on the description of how to write an informal specification and how to transform it into a semi-formal specification.

The rest of this paper is organized as follows. Section 2 gives a short overview of the examination monitoring system. This is followed in Section 3 by giving a short overview of the SOFL method, in particular a two-step approach for an abstract requirements specification. A brief overview of related work is given in Section 4. Section 5 discusses how the SOFL approach is applied in writing the specifications for the examination monitoring system. Section 6 describes our experience and lesson learnt from this project. Finally, in Section 7 we conclude the paper and provide several suggestions for future research.

II. EXAMINATION MONITORING SYSTEM

Software Engineering Programme (SEP), for a Department of Computing & Software Engineering, is a one programme offered at the Faculty of Computer Science & Information Technology, University Malaysia Sarawak. This programme is led by a Programme Coordinator (PC). He is responsible to assign faculty members to be a *Setter* for drafting a final examination paper and appoints a *Vetter* for reviewing the exam paper, and monitors all the process of academic flow for SEP throughout the semester.

A. System Mechanism

The examination monitoring system (as shown in Fig. 1) contains three distinct types of objects: *Person*, *Courses*, and *TestDraft*. There are three kinds of *Person*: *ProgrammeCoordinator*, *Setter*, and *Vetter*. The *ProgrammeCoordinator* is responsible to assign a *Setter* for preparing an examination question for the *Course* and appoints a *Vetter* to review the *TestDraft* in which has been prepared by the *Setter*. Two *Vetters* will be assigned: *Vetter1* and *Vetter2*. Two possible options that the *Vetters* (*Vetter1* and *Vetter2*) can review the *TestDraft*: (i) sequentially vet- in which the *Vetter1* will vet the *TestDraft* based on *CheckParameters*. After giving any comment regarding the *TestDraft*, the *Vetter1* will pass the same *TestDraft* to the *Vetter2* for his/her comments; (ii) in parallel vet- in which both *Vetter1* and *Vetter2* vet the *TestDraft* in parallel.