International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 5
ISSN: 2088-5334

Knowledge Representation Framework for Software Requirement Specification

L. Jelai[a,1], E. Mit[a,2], S. Samson Juan[a,3]

[a]*Faculty of Computer Science and Information Technology,Universiti Malaysia Sarawak (UNIMAS), 94300, Malaysia*
E-mail: [1]lilyjelai@gmail.com; [2]edwin@unimas.my; [3]sjsflora@unimas.my

***Abstract*— The need to extract correct information has become one of the main issues when analyzing the software requirement specification (SRS) documentation. The amount of gathered knowledge depends on the size of the information. However, the complexity of software systems is continuously increasing. As software systems change to more complicated systems, the information from the SRS documents may not be easily comprehended. For example, each annotation requirements tasks target the different types of information, and these tasks require the availability of experts specialized in the field. Large scale annotation tasks require multiple experts and very costly. If the number of experts is limited, annotation tasks may overwhelm the experts. The organization would not complete their objectives if they failed to manage their data because poor knowledge management affects many operations within the organization. To extract such vast information and turn it to useful knowledge, a company needs top quality software. This technology should able to input, store, and access systematically. This paper will discuss a framework based on the knowledge-based method, an attempt to improve knowledge representation. In this approach, WordNet 2.1 would be used as the knowledge source used to identify concepts represented by each word in a text from the SRS document.**

***Keywords*— knowledge-based; software requirement specification; WordNet.**

I. INTRODUCTION

Typically, the development process will include a software requirement specification (SRS) documentation. SRS documents portray a complete system behavior as it elaborates on the functional requirements, non-functional requirements, and other aspects of software systems such as business processes [1]. Thus, the achievement of a software project mostly relies on the quality of SRS documentation. SRS document helps as an input during the earlier phase, coding phase, and testing phase.

As software frameworks have been developed, software engineers ought to deal with a developing amount of data and information. According to Antunes, Gomez, and Seco [2], creating new supporting tools to support knowledge management amid software development and maintenance is essential within the software industry. This is due to the overwhelming knowledge obtain during the software development process. However, this overwhelming knowledge can be an asset for a software company. Therefore, how to make this knowledge valuable?

Antunes, Gomez, and Seco [2] recommended that every company must know how to utilize the knowledge for future reuse fully. Thus, companies should build components that can implement contexts characterization and data classification. One of the suggested ideas is to exploit the knowledge representation languages and turn it to domain conceptualizations, such as ontologies. Apart from that, these components must come out with solutions that oversee any access and exchange of important data [2].

SRS documents are frequently found to be corrupted. Most of the sentences used a full of ambiguity because it is written in an unrestricted natural language. Due to that reason, an expert must recognize and resolve any vague information manually [3]. The SRS documents can also be found in unstructured form. This situation would need additional efforts from the experts as they must extract significant information about the software. Most of these experts stated that they usually found the sentences describing functional requirements in other sections containing non-functional requirements and vice versa. To understand the differences between non-functional requirements and functional requirements, according to Hussain, Ormandjieva, and Kosseim [4], the non-functional requirement is a software requirement that articulates the quality requirements and the constraints over the related behavior of the system. For example:

> *"All the mandatory attributes cannot be empty, and the budget amounts cannot be negative"* [4].

Meanwhile, a functional requirement is defined as a software requirement that articulates the required behavior