International Journal of Information Technologies and Systems Approach
Volume 15 • Issue 1

Performance Measurement of a Rule-Based Ontology Framework (ROF) for Auto-Generation of Requirements Specification

Amarilis Putri Yanuarifiani, Multimedia University, Malaysia
Fang-Fang Chua, Multimedia University, Malaysia
Gaik-Yee Chan, Multimedia University, Malaysia

ABSTRACT

Documenting requirements specification requires a lot of effort from stakeholders and developers. Time and knowledge limitations are also obstacles in creating structured requirements documents. Previous works proposed a framework for automated generation of requirements specifications called rule-based ontology framework (ROF). The requirements documentation phase produces two outputs: process modeling according to the business process model and notation (BPMN) standard and software requirements specification (SRS) documents following the ISO/IEC/IEEE 29148:2018 standard. In this paper, the authors do performance measurement of ROF in the IS project case study which includes validating ROF prototype by performing user acceptance test (UAT), measuring effectiveness by calculating notation error and requirements error, and measuring efficiency by calculating the time spent in producing documents. The efficiency and effectiveness of both are measured by comparing BPMN graph and SRS document generated by ROF with BPMN graph and SRS document that are created manually by the stakeholders.

KEYWORDS

Auto-Generation of Requirements Documents, Performance Measurement, Requirements Engineering, Rule-Based Ontology, Software Requirements Specification

INTRODUCTION

Requirements Engineering is an important and crucial process in software development (Khan et al., 2015). It consists of four main phases: elicitation, documentation, validation, and management. Elicitation phase gathers the requirements from stakeholders and other sources to refine the requirements in greater detail. It is the most difficult task that may increase the risk of project failure (Nisar & Nawaz, 2015).

Requirements list from elicitation phase is written using natural language or conceptual models. A requirements document must be easily understood by both the stakeholders and technical person (Verma & Kass, 2008), so as to ensure the written requirements are matched with the organization's

DOI: 10.4018/IJITSA.289997

need. On the other hand, the requirements document must also meet the formality of the appropriate structure to ensure the developers can translate it correctly in the programming language. Potential errors or mismatch requirements which are found during code development process might increase project costs.

Our prior study (Yanuarifiani et al., 2019) proposed a Rule-based Ontology Framework (ROF). ROF covers two phases of Requirements Engineering which are elicitation and documentation. In elicitation phase, initial requirements are collected using the gap identification method that involves stakeholders and developers. By using Kano's model, we perform requirements prioritization to determine which requirements need to be eliminated and which requirements will be implemented. The final requirements are then stored in an ontology taxonomy called Requirements Ontology (RO). By using RO as input, the ROF automatically generates two types of requirements documents; a semi-formal modeling document in Business Process Model and Notation (BPMN) graph and a Software Requirements Specification (SRS) document in ISO/IEC/IEEE 29148:2018 template.

The application of ROF aims to increase effectiveness and efficiency of requirements documentation and minimize the risk of making human mistakes when preparing documents manually. All ROF phases are implemented as ROF prototype functionalities and used to develop Workload Management Applications as a case study. The aim of this paper is to ensure ROF objectives are achieved, by measuring ROF performance with three tasks. The first task, we validate the prototype ROF to ensure stakeholders receive all features and confirm the results by conducting a User Acceptance Test (UAT) by testing end-to-end functionality. The second task, measurement of effectiveness, is done by counting notation and requirements error. The third task, efficiency measurement is done by calculating the time needed to produce the requirements document. Last two tasks are measured by comparing the generated requirements document of ROF and the requirements documents created manually by the stakeholders.

Section two presents a literature review related to the concept of requirements document and the evaluation methods. Section three explains the concept, implementation, and output of ROF in a case study. Performance measurement for the prototype validation, effectiveness measurement and efficiency measurement are explained in Section four. A summary of the performance measurement results and analysis is discussed in Section five. Section six presents the conclusions of this paper.

RELATED WORKS

Requirements Specification

Specification of requirements, as a result of the elicitation process, needs to be written in the form of a requirements specification document. The document contains details of functional and non-functional features of the system to be built without mentioning technical knowledge. It can be written in natural or in other notation such as a graphical or mathematical system model (Sommerville, 2016).

BPMN is a common and widely used modeling language that provides process element notation independently (Braun, 2015). The main purpose of the construction of the BPMN by the Object Management Group (OMG) is to provide notation that is easily understood by all teams involved in Information System (IS) development projects starting from business analysts, technical developers and end users (Object Management Group, 2011). BPMN packages include BPMN core elements, process diagram, collaborative diagram and conversation diagram. In addition to fully supporting business modeling, BPMN is also one of the notation that supports semantic execution (Geiger et al., 2017) and integrated extension mechanism. Writing requirements specifications in BPMN notation can make it easier for business users to ensure system functionality is in accordance with company needs. As for technical developers, system specifications are clearer and can easily be translated into technical designs. Figure 1 is an example of using Collaborative Process in BPMN notation.