Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 637 – 641

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies

(ANT 2016)

Ontology Software Quality Model for Fuzzy Logic Evaluation Approach

Madjid Kara[a], Olfa Lamouchi[b], Amar Ramdane-Cherif [a,]*

[a] *LISV Laboratory, University of Versailles St-Quentin-en-Yveline, 10-12 avenue de l'Europe, 78140 Velizy, France*
[b] *Engineers National School of Carthage, 45 Rue des entrepreneurs, 2035, Tunisia*

Abstract

A software quality model is a very useful instrument for software quality evaluation. Researchers have studied the axis of modeling and evaluating software quality and several approaches have been proposed. These approaches were limited to a specific use field and did not offer a quality profile enabling us to evaluate a global software quality model. The evaluation based on International Organization for Standardization (ISO) models has emerged; however, these models do not guide us for their use in a global evaluation approach.
In this paper we treat extracted data from an ambient distributed system. Our work presents the following contributions:(i) creating a generic software quality model based on several existing software quality standards, (ii) proposing an instantiation algorithm to extract specified software evaluation model from generic software quality models, (iii) proposing a new global evaluation approach of the specified software evaluation model using measurement metrics and fuzzy logic.

Peer-review under responsibility of the Conference Program Chairs

Keywords: quality models; quality evaluation; quality measurement; metrics

1. Introduction

Software evaluation is an important step in ensuring sufficiency of software product quality. Reliable software quality model is based on precise, objective and calculable metrics, defined without any ambiguity to provide an incontestable evaluation quality[1]. The choice of data's appropriate representation is one of the most crucial tasks in the entire system development process[2]. The existing software quality models are generally hierarchical, grouping a set of factors, criteria and sub-criteria[3]. Several research works on software quality models have been completed and many classifications have been developed but the most important are: Mc Call et al-1977, Boehm et al-1978, FURPS Model-1992 and Dromey model-1995[4]. The first finding was that the proposed approaches were limited in their use fields and each researcher had its own criteria interpretation which led to have divergence

* Corresponding author. Tel.: (+33) 06.63.76.65.49; Fax: (+33) 01.39.25.49.85
*E-mail address:*madjid.kara@lisv.uvsq.fr

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.143

in criterion definition[5]. To group different software quality views, ISO/IEC 9126[6] standard was created in 2003. An update was established as ISO/IEC CD 25010[7] in 2007. It is used to establish software quality requirements and perform evaluations[8] using ISO/IEC 25023 standard[9] that contain a set of software quality basis measures. ISO/IEC 25010 has also been used as reference for its reuse or extension. Among these works we can mention: Al-Badareen-2011, Dubey-2012, Al-Qutaish-2010 and Samadhiya-2013[4]. Even if ISO models provide a solid theoretical basis, they present a lack of user guide in a global evaluation approach[10], and there is no explicit link between criteria and metrics[11] to evaluate all criteria.

The aim of our study is to evaluate a software quality model of an ambient distributed system. We have proposed a generic software quality models based on several quality standards as well as models proposed by others researchers. Equivalence relations will be established between criteria of these standards. We also propose a global evaluation approach to evaluate our instantiated software quality model using metrics and fuzzy logic[3, 12].

The rest of this article is organized as follows. In the next section, we present a global generation and evaluation approach of our specified software evaluation model. In section 3, we define the software evaluation model and its components then in section 4 we pass to the evaluation algorithm and tools used in our approach. In the final section, we provide an example of specified software evaluation model and then we will end up with a conclusion.

2. Proposed global approach

The objective of our approach is to evaluate a software quality model involved in the interaction process following the steps 1 to 6 (see Fig. 1). A specified software evaluation model will be instantiated from a generic software quality models, taking into account the extracted data from the interaction process. It consists of several factors, criteria and sub-criteria represented hierarchically. Each factor is composed of one or several criteria and each criterion is composed of one or more sub-criteria until reaching the measurable criteria called leaves. The evaluation approach will use fuzzy logic to evaluate the entire specified model and end up with a final numerical result.

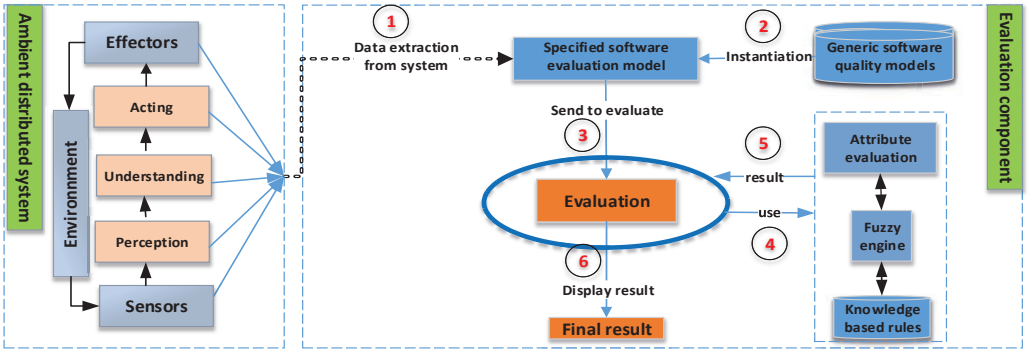

Fig. 1. Quality evaluation approach

3. Composition of software evaluation model (see Fig. 2):

We have software quality model, represented hierarchically where each factor is composed of one or more criteria, and each criterion is composed of one or several sub-criteria until reaching the lowest level. Criteria in the lowest level of the model are called leaves criteria. The metric model includes a set of metrics, representing measurement or evaluation procedures that assign numerical values for leaves criteria. Each metric uses one or more metrics variables. The interdependence model will support metrics variables variation to satisfy quality models, we will determine the impact of a possible variation level of one criterion on other's. The equivalence relations model defines the relations between metrics of different models. It helps to calculate metrics of a specific criterion that does not have metrics variables. Each equivalence relation contains the equivalents criteria and order of equivalence that indicate the search priority of criteria.

Fig .2. Composition of software evaluation model