Chapter 2
I-Way:
A Cloud-Based Recommendation System for Software Requirement Reusability

Chetna Gupta
Jaypee Institute of Information Technology, Noida, India

Surbhi Singhal
Jaypee Institute of Information Technology, Noida, India

Astha Kumari
Jaypee Institute of Information Technology, Noida, India

ABSTRACT

This study addresses the problem of effectively searching and selecting relevant requirements for reuse meeting stakeholders' objectives through knowledge discovery and data mining techniques maintained over a cloud platform. Knowledge extraction of similar requirement(s) is performed on data and meta-data stored in central repository using a novel intersective way method (i-way), which uses intersection results of two machine learning algorithm namely, K-nearest neighbors (KNN) and term frequency-inverse document frequency (TF-IDF). I-way is a two-level extraction framework which represents win-win situation by considering intersective results of two different approaches to ensure that selection is progressing towards desired requirement for reuse consideration. The validity and effectiveness of results of proposed framework are evaluated on requirement dataset, which show that proposed approach can significantly help in reducing effort by selecting similar requirements of interest for reuse.

DOI: 10.4018/978-1-5225-9659-2.ch002

INTRODUCTION

Requirements reuse is a deliberate methodology which provides organizations with the novel ability to share a requirement crosswise over projects without absorbing unnecessary duplication of artifacts, thereby reducing development cost and accelerating time to market delivery. In last few years researchers and IT practitioners have attracted towards the concept of requirement reusability with the objective to decide on a set/subset of ready to use requirements ("as-is" or with some modifications). Though requirement reusability supports resource optimization during development and helps in reduction of errors but it is a challenging task that requires careful decision making and planning to provide desired functionality to the user. An early start preferably during requirement elicitation in this direction is most beneficial form of software reuse to save cost, time and accelerate time to market delivery. Decision on which requirement to reuse and to what extend depends upon the project context and situation at hand. Existence of a particular requirement does not guarantee that it is reusable in its present form. In the light of reusability, requirements are tuned to specific needs in order to increase their value to the customer and adaptability for the project at hand. Software developers should not only focus on the context in which an existing requirement can be used rather they should analyze other aspects of a requirement like, dependency between two or more requirements, related requirements that go well with the chosen one and might be reused along in order to add more value to the system, use cases, tests, attributes and hierarchy. In other words, a well written (at the right level of abstraction and scope) requirement should only be considered for reuse as comparison to generic requirements which might not save cost and time of software developers due to non availability of complete description of a requirement. Though it has received little attention (Kotonya & Sommerville, 1997; Lamsweerde, 2000), reusing early software products and processes can impact the life cycle from two basic points of view: improving the requirements engineering process (Cybulski, 1998; Sutcliffe et al., 1998), and supporting the development with reuse (Bellinzona et al., 1993; Bellinzona et al., 1992) process. Several tools and approaches for example, (Chou et al., 1996; Mannion et al., 1999; Mobasher & Cleland-Huang, 2009; Pacheco et al., 2017; Bakar et al., 2015; Schmitt & Liggesmeyer, 2015; Paydar & Kahani, 2015) etc. have been proposed in the literature to support requirement reuse as part of their functionality which can assist in finding a better set of software requirements according to a set of goals and constraints. Majority of them uses structuring and matching of requirements as a method. Other approaches use requirement specifications with similar behaviour or reuse, based on the assumption that specifications that exhibit similar behaviour are appropriate for reuse for the system which is under development.