Information and Software Technology xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Shared service recommendations from requirement specifications: A hybrid syntactic and semantic toolkit

M. Brian Blake [a], Iman Saleh [a,*], Yi Wei [b], Ian D. Schlesinger [c], Alexander Yale-Loehr [d], Xuanzhe Liu [e]

[a] University of Miami, Coral Gables, FL, United States
[b] Microsoft Corporation, Seattle, WA, United States
[c] Georgetown University, Washington, DC, United States
[d] U.S. Army Corps of Engineers, Alexandria, VA, United States
[e] Peking University, Beijing, China

ARTICLE INFO

Article history:
Received 20 March 2013
Received in revised form 8 April 2014
Accepted 23 May 2014
Available online xxxx

Keywords:
Requirements engineering
Similarity analysis
Shared service discovery

ABSTRACT

Context: Software Requirement Specifications (SRSs) are central to software lifecycles. An SRS defines the functionalities and constraints of a desired software system, hence it often serves as reference for further development. Software lifecycles concerned with the conversion of traditional systems into more service-oriented infrastructures can benefit from understanding potential shared capabilities through the analysis of SRSs.

Objective: In this paper, we propose an automated approach capable of recommending shared software services from multiple text-based SRSs created by different organizations. Our goal is to facilitate the identification of overlapping requirements in these specifications and subsequently recommend shared components, which promotes software reuse. The shared components can be implemented as services that are invoked across different systems.

Method: Our approach leverages the syntactic similarity of the SRS text augmented with semantic information derived from the WordNet database. This work extends our earlier studies by introducing an algorithm that utilizes noun, verb, and predicate relations to enhance the discovery of equivalent requirements and the recommendation of reusable services. A prototype system is implemented to evaluate the approach and experimental results have shown effective recommendation of requirements and their realized shared services.

Results: Our automatic recommendation approach generates recommendations in few minutes compared to 9 h when services are manually inspected by developers. Our approach is also able to recommend services that are overlooked by the same developers, and to identify similarity between requirements even if these requirements are reworded.

Conclusion: We show through experimentation that we can efficiently recommend services by leveraging both the syntactical structure and the semantic information of a requirements document and that our approach is more effective than the manual selection of services by experts. We also show that our approach is effective in detecting similar requirements for a single system and hence discovering opportunities for software reuse.

1. Introduction

A significant and early step in software development processes is to identify and describe system features and boundaries [1]. Such descriptions are usually captured in text-based documents, known as Software Requirement Specifications (SRSs) [2,3]. Other visual approaches like the Unified Modeling Language (UML) can also be mapped into text-based descriptions by using XML or other techniques [4]. Although the emergence of agile development attempts to reduce the need for text-based specifications, still some form of descriptions will always be necessary for group collaborations. This phenomenon leads to a natural question, that is, *What if we could discover shared services from SRSs from peer organizations that may suggest feasible integration points?*

Commercial, academic, and government organizations alike may be able to expand their capabilities by sharing their underlying software services with complementing peers. The motivation

* Corresponding author. Tel.: +1 305 284 2658.
E-mail address: iman@miami.edu (I. Saleh).

http://dx.doi.org/10.1016/j.infsof.2014.05.019
0950-5849/

for this work was conceptualized during a consulting engagement. One of the authors served as a consultant for a consortium of national security agencies. As a part of the consolidation of information after the atrocities of 9/11, several of the national security agencies initiated software development projects to develop data sharing tools and portals. Many of these new software systems planned to utilize the modularity of the principles of *Service-Oriented Architecture* (*SOA*). Such SOA-based software systems suggest better interoperability with systems from peer organizations and promote an open environment where services are potentially universally-available across organizations. A challenge surrounding these new initiatives is that the underlying organizations developed their new capabilities without considering the activities of other peer security agencies. The consulting project was to lead a discussion between the architects of each of the new applications to determine if shared services could be developed that could be hosted at an inter-organizational enterprise service bus (ESB). These services could be reused and reduce repetition in software development across the organizations. Recommending shared service was an exercise where each of the architects met in a room for one-week attempting to associate their new services with those of other organizations. The architects used their system requirements and high-level software specifications to recollect potential services in their system and look for overlapping requirements with other organizations. This was a tedious and costly endeavor. The motivation of the work in this paper is to develop techniques and adaptive tools to automate this process. Such tools would be *immediately valuable in corporate environments* as a first-pass assessment toolkit for organizations that require an integration of their underlying software and services. The discovery of overlapping requirements (as introduced in this paper) could be the first step to identifying abstract services that may prove productive to multiple organizations. Automating such an approach allows organizational decision-makers to see functional and business overlaps that are not easily identified via traditional human inspection.

In this paper, we describe an automated approach that reasons about the overlaps between different text-based software specification documents. The proposed approach incorporates a customized information extraction algorithm that collects relevant data from those SRSs. The approach is hybrid as it combines an existing syntactical similarity approach [5,6] and a customized semantic-oriented algorithm that utilizes the WordNet [7] database.

We will try to answer the following research questions:

- RQ-1. *How effective is our approach in detecting similar requirements and recommending shared services?*
- RQ-2. *Can our approach detect similar requirements even when they use different words or assume different sentence structures but have the same underlying meaning?*
- RQ-3. *What is the performance of our automated approach on shared services recommendation compared with that of human analysts?*
- RQ-4. *Can grammar elements, such as noun, verb and predicate, be leveraged to enhance the precision of recommended shared services?*
- RQ-5. *Can shared services or similar requirements be detected in SRSs randomly retrieved from different organizations?*

The paper proceeds in Section 2 with a comparison of our approach with related work. Section 3 defines the structural model of an SRS document to facilitate the similarity comparison of requirements. Section 4 describes the proposed approach for service discovery and requirements overlap, followed by a description of our implemented prototype in Section 5. Our evaluation and experiment results on various SRS documents from multiple organizations are presented in Section 6. Finally in Section 7, we give a conclusion based on our evaluations as they address the aforementioned research questions. We also discuss the direction of our future research in Section 7.

2. Background and related work

The state of the art in automated requirements elicitation is surveyed in [8]. The authors categorize work in that field based on the tools, degree of automation, knowledge reuse, evaluation approach, and evaluation concepts. Using their categorization, our work falls belong to the fully automated approaches with an imported knowledge base (WordNet). As for the evaluation, the authors recognize several evaluation methodologies. Among these methodologies, we are following the *controlled experiment* evaluation method which involves studying the presented method or artifact in a controlled environment, which can be done e.g. by comparing the performance of an analyst using the artifact with the performance of an analyst devoid its support [8].

In [9], the authors propose a semi-supervised approach for the automatic detection of non-functional requirements. They use a learning set of pre-categorized requirements to train a classifier. Once a classifier is trained, it is used to categorize further unlabeled requirements. By comparison, our work is not limited to non-functional requirements and our goal is to detect similarity between in a requirements' document using syntactic and semantic techniques.

It has been proven that early identification of faults within an SRS can improve the entire software lifecycle. Ultimately, early identification reduces time in a later stage required to fix these flaws [10]. Similarly, the software lifecycle can also benefit from understanding shared and well defined services extracted from other SRSs. Just as we often leverage third party libraries in the implementation stage to avoid repetition, utilizing shared services in an SRS can save time and cost as well as improve the interoperability of the underlying software system. The approaches in our work do not advance the state-of-the-art in general text matching/document indexing, information retrieval area, whether syntactic or semantic [11,12]. But, rather, the unique contribution here relates to that application of general techniques to the requirements matching and service recommendation domain. As such, related research projects can be characterized by two categories, (1) discovery of similar/related requirements from different requirement specifications and (2) automatic extraction of meta- or semantic information that can be used to define/recommend open, but shared services.

With regards to (1), Spanoudakis et al. [13] developed elegant formal approaches for discovering overlaps across requirement specifications. Their intention is to find overlaps between different specifications to reveal inconsistencies. Their work requires the formalization of specification using first-order logic. Our approach presented here is more applied. As such, the analysis of specifications in our work is directly related to aspects of the specification documents and the underlying sentence structure. In our work, overlap is further utilized to infer shared services across multiple organizations.

Zachos et al. [14] developed a service recovery tool that can be integrated with RUP and UML to generate service queries. Their research focuses on assisting users to find the most suitable software service from a service database. Their approach is effective for reusing open source services. Alternatively, our approach uncovers the abstract capabilities across multiple SRSs that may be later developed into new services. In the area of Natural Language Processing (NLP), Hussain et al. [15] develop a text classifier aiming at splitting the Non-Functional Requirements (NFR) from requirement documents. Our approach identifies equivalent requirements within different requirement specifications as

Please cite this article in press as: M. Brian Blake et al., Shared service recommendations from requirement specifications: A hybrid syntactic and semantic toolkit, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.05.019