Information and Software Technology 55 (2013) 1845–1859

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

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

Combining service-orientation and software product line engineering: A systematic mapping study

Bardia Mohabbati [a,*], Mohsen Asadi [a], Dragan Gašević [a,b], Marek Hatala [a], Hausi A. Müller [c]

[a] Simon Fraser University, 13450 102 Ave. Surrey, BC V3T 5X3, Canada
[b] Athabasca University, 1 University Drive Athabasca, AB T9S 3A3, Canada
[c] University of Victoria, STN CSC Victoria, BC V8W 3P6, P.O. Box 3055, Canada

A R T I C L E I N F O

Article history:
Received 17 December 2012
Received in revised form 24 May 2013
Accepted 24 May 2013
Available online 13 June 2013

Keywords:
Systematic mapping
Service-oriented architecture
Software product lines

A B S T R A C T

Context: Service-Orientation (SO) is a rapidly emerging paradigm for the design and development of adaptive and dynamic software systems. Software Product Line Engineering (SPLE) has also gained attention as a promising and successful software reuse development paradigm over the last decade and proven to provide effective solutions to deal with managing the growing complexity of software systems.
Objective: This study aims at characterizing and identifying the existing research on employing and leveraging SO and SPLE.
Method: We conducted a systematic mapping study to identify and analyze related literature. We identified 81 primary studies, dated from 2000–2011 and classified them with respect to research focus, types of research and contribution.
Result: The mapping synthesizes the available evidence about combining the synergy points and integration of SO and SPLE. The analysis shows that the majority of studies focus on service variability modeling and adaptive systems by employing SPLE principles and approaches.

Result: In particular, SPLE approaches, especially feature-oriented approaches for variability modeling, have been applied to the design and development of service-oriented systems. While SO is employed in software product line contexts for the realization of product lines to reconcile the flexibility, scalability and dynamism in product derivations thereby creating dynamic software product lines.
Conclusion: Our study summarizes and characterizes the SO and SPLE topics researchers have investigated over the past decade and identifies promising research directions as due to the synergy generated by integrating methods and techniques from these two areas.
Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.

Contents

* Corresponding author. Tel.: +1 778 782 7431.
E-mail addresses: mohabbati@sfu.ca (B. Mohabbati), masadi@sfu.ca (M. Asadi), dgasevic@sfu.ca (D. Gašević), mhatala@sfu.ca (M. Hatala), hausi@cs.uvic.ca (H.A. Müller).

0950-5849/$ - see front matter Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2013.05.006

1. Introduction

Service-Orientation (SO), which is manifested in an architectural style that is commonly referred to as Service-Oriented Architecture (SOA), has substantially contributed to changing the perspective of todays software development. Service-oriented computing is gaining momentum as a means to develop scalable and flexible distributed applications. This momentum aims at addressing various challenges such as application integration, reusability, modularity and interoperability [38].

SO focuses on creating and developing application solutions by utilizing services as building blocks of software that encapsulate functionality and provide flexibility through dynamic binding. The visionary promise of SO is that applications are composed seamlessly by loosely-coupled services in order to create adaptive and dynamic systems. Services are characterized as a set of autonomous, platform-independent computational units which can be described, published, discovered, and dynamically composed and assembled [38]. SOA provides the architectural underpinnings to support software reuse and enables variability at both design- and run-time. Variability refers to the ability of a software system to be configured, customized and adopted to meet particular requirements [45]. However, SOA lacks support for modeling and managing variability that promotes configurability (i.e., customization) and systematically-managed reuse [3,48]. SOA artifacts, including service specifications and process models, are not generally designed with variability for planned and enforced reuse requirements in mind.

Variability is an important concept, and variability analysis, modeling and management have been core research subjects in software product line research. Software Product Line Engineering (SPLE) addresses the issues of engineering and developing software-intensive systems and supports large-scale reuse in the course of development. SPLE offer effective methods and techniques for variability modeling and systematic reuse in software development in order to (i) support configurable software architectures and (ii) enable for mass customization of software-intensive systems [41].

Combining and integrating SO and SPLE have been a subject of considerable research interest in recent years, observed through the literature (e.g., [2,9,11,15,20,22,33]), and dedicated workshop series [12,31,32]. Hence, we need to synthesize the evidence regarding the usefulness of combination. Therefore, we conducted a systematic mapping study, based on our preliminary results of our previous work [34], to analyze the existing research and relevant literature published on this topic. Systematic mapping studies (a.k.a. scoping studies) are designed and performed to provide a wide overview of a research area [28]. In software engineering, systematic mapping studies have been recommended when a research topic is new or not mature enough to comprise a set of comparable empirical studies [28,27]. The aim is to “map out” the research undertaken instead of answering detailed research questions in contrast to systematic literature reviews (SLRs), which derive very specific research questions [6,28,40]. Thus, a mapping study as a part of evidence-based software engineering is conducted if research evidence exists on a topic and provides an indication of the quantity of the evidence [28].

This paper presents a systematic mapping study that aims at collecting evidence about how SO and SPLE are combined or integrated with the aim to identify the research trends and categorize studies at a higher granularity level. Our goal is to provide a map of existing research and synthesize current evidence on the integration of two paradigms. Moreover, the outcomes of our mapping study can help to identify research challenges and to direct ongoing research.

The rest of paper is constructed as follows: Sections 2 and 3 describe the methodology followed in our conducted mapping study and classification scheme. Section 4 reports results. The threats to validity are given in Section 5. Section 6 presents a discussion. Section 7 conclude this work and highlights the direction of future work.

2. Systematic mapping and research method

Efforts in software engineering research are dedicated toward developing a standard methodology for conducting mapping studies [7,28,40]. Peterson et al. [40] describe methods for conducting mapping studies and discuss differences between systematic mapping studies and systematic literature reviews. Moreover, they provide guidelines for a broader set of situations where either or both systematic mapping studies and systematic literature reviews are appropriate and required to be conducted. A systematic literature review is conducted as a means of identifying, evaluating, interpreting, and comparing all available researches that are relevant to a particular research question and relative merits of competing technologies. In contrast, a systematic mapping study provides a systematic and objective procedure to determine the nature and extent of the empirical-study data to answer a particular research question [7].

Mapping studies often use the same basic methodology as systematic literature reviews; however, they aim at identifying and classifying all related research into a broad software engineering topic.

The software community has been working to towards well-defined methods to conduct mapping studies [40]. The procedure of systematic mapping study presented in this paper combines a well-organized set of proper practices both to undertake mapping study and to systematically review guidelines in the context of software engineering [5,26,40]. The combination of guidelines help us to leverage both systematic mapping and literature review techniques. The major steps of our systematic mapping process comprised: (1) definition of a protocol of the study and research questions; (2) exploratory search and data collection; and (3) analysis of the collected data and reporting. Fig. 1 summarizes a process followed in our systematic mapping study. The details of each step are described in the following subsections.

Download English Version:

https://daneshyari.com/en/article/10366582

Download Persian Version:

https://daneshyari.com/article/10366582

Daneshyari.com