Information and Software Technology 61 (2015) 33–51

Contents lists available at ScienceDirect

Information and Software Technology

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

A systematic mapping study of search-based software engineering for software product lines

Roberto E. Lopez-Herrejon *, Lukas Linsbauer, Alexander Egyed

Institute for Software Systems Engineering, Johannes Kepler University Linz, Altenbergerstr. 69, 4040 Linz, Austria

ARTICLE INFO

Article history:
Received 28 August 2014
Received in revised form 16 January 2015
Accepted 17 January 2015
Available online 29 January 2015

Keywords:
Software product line
Systematic mapping study
Search based software engineering
Evolutionary algorithm
Metaheuristics

ABSTRACT

Context: Search-Based Software Engineering (SBSE) is an emerging discipline that focuses on the application of search-based optimization techniques to software engineering problems. Software Product Lines (SPLs) are families of related software systems whose members are distinguished by the set of features each one provides. SPL development practices have proven benefits such as improved software reuse, better customization, and faster time to market. A typical SPL usually involves a large number of systems and features, a fact that makes them attractive for the application of SBSE techniques which are able to tackle problems that involve large search spaces.
Objective: The main objective of our work is to identify the quantity and the type of research on the application of SBSE techniques to SPL problems. More concretely, the SBSE techniques that have been used and at what stage of the SPL life cycle, the type of case studies employed and their empirical analysis, and the fora where the research has been published.
Method: A systematic mapping study was conducted with five research questions and assessed 77 publications from 2001, when the term SBSE was coined, until 2014.
Results: The most common application of SBSE techniques found was testing followed by product configuration, with genetic algorithms and multi-objective evolutionary algorithms being the two most commonly used techniques. Our study identified the need to improve the robustness of the empirical evaluation of existing research, a lack of extensive and robust tool support, and multiple avenues worthy of further investigation.
Conclusions: Our study attested the great synergy existing between both fields, corroborated the increasing and ongoing interest in research on the subject, and revealed challenging open research questions.

Contents

* Corresponding author. Tel.: +43 732 2468 4380.
E-mail addresses: roberto.lopez@jku.at (R.E. Lopez-Herrejon), lukas.linsbauer@jku.at (L. Linsbauer), alexander.egyed@jku.at (A. Egyed).

http://dx.doi.org/10.1016/j.infsof.2015.01.008
0950-5849/

1. Introduction

Search Based Software Engineering (SBSE) is an established yet young discipline that focuses on the application of search-based optimization techniques to software engineering problems [1]. In this article, we follow Harman et al. who consider SBSE techniques to primarily include metaheuristic search based optimization techniques and classical operations research techniques [1,2]. Some examples of SBSE techniques are: evolutionary computation techniques[1] (e.g. genetic algorithms), basic local searches (e.g. hill climbing, simulated annealing or random search) [4], and integer programming [1].

Software Product Lines (SPLs) are families of related systems whose members are distinguished by the set of features they provide [5,6]. *Variability* is the capacity of software artifacts to vary and its effective management and realization lie at the core of successful SPL development [7]. *Feature models* are tree-like structures that establish the relations between features and have become the de facto standard for modeling variability [8,9]. Over the last decade, extensive research and practice both in academia and industry attest to the substantial benefits of applying SPL practices [6,10,11]. Among the benefits are better customization, improved software reuse, and faster time to market.

Typical SPLs have a large number of features that are combined in complex feature relations yielding a large number of individual software systems that must be effectively and efficiently designed, implemented and managed. Precisely this fact is what makes SPL-related problems suitable for the application of SBSE techniques which are generic, flexible, robust, and have been shown to scale to large search spaces such as those that typically characterize SPLs (e.g. [12]). This recent realization has sparked a surge of research and application at the intersection of SBSE and SPLs, which has manifested with an increasing number of articles at many of the publication outlets of both research communities. This is precisely what prompted us to perform a *systematic mapping study* to provide an overview of the research at the intersection of these two fields [13–15]. In contrast with a *systematic literature review* whose goal is primarily to identify best practice [13,15–17], our general goal was to identify the quantity and the type of research and results available, and thus highlight possible open research problems and opportunities, for both SBSE and SPL communities. More concretely we wanted to identify at what stages of the SPL development life cycle have SBSE techniques been used and which ones. We also wanted to find out the provenance, number and types of artifacts used as case studies as well as how they were empirically analyzed. And finally, which are the fora where the research work was published.

Our study corroborated the increasing and ongoing interest in applying SBSE techniques in SPLs. We found that the most common application is software testing, and the most common techniques are genetic algorithms and multi-objective evolutionary algorithms. Our study identified the need to improve the robustness of the empirical evaluation (e.g. more adequate statistical analysis) and the need for more extensive and robust tool support. We hope that this mapping study not only serves to highlight the main research topics at the intersection of SBSE and SPLs but that it also serves to encourage researchers to pursue work at the intersection of both areas.

The paper is structured as follows. Section 2 presents the process we followed for our systematic mapping study. It details the research questions addressed, how the search was performed, the classification scheme used, and how the data was extracted and analyzed. Section 3 presents the results we obtained for each research question. Section 4 contains a description of the results found along with open questions and avenues worth of further investigation. Section 5 summarizes the threats to validity we identified in our work and how they were addressed. Section 6 concisely describes the existing review studies and surveys of SPLs and SBSE. Section 7 summarizes the conclusions of our study and future work.

2. Systematic mapping study

Evidence-Based Software Engineering (EBSE) is an emerging software engineering area whose goal is *"to provide the means by which current best evidence from research can be integrated with practical experience and human values in the decision making process regarding the development and maintenance of software"* [18]. One of the approaches advocated by EBSE is systematic mapping studies whose goal is to provide an overview of the results available within an area by categorizing them along criteria such as type, forum, frequency, etc. [14]. For performing our mapping study, we followed the protocol proposed by Petersen et al. [14], whose main stages are shown in Fig. 1. Next we describe each of the processes

[1] Evolutionary computation is an area of computer science, artificial intelligence more concretely, that studies algorithms that follow Darwinian principles of evolution [3].