Applied Soft Computing 49 (2016) 1243–1255

Contents lists available at ScienceDirect

Applied Soft Computing

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

A hybrid approach to suggest software product line portfolios

Pedro de Alcântara dos Santos Neto [a,*], Ricardo Britto [a,b], Ricardo de Andrade Lira Rabêlo [a], Jonathas Jivago de Almeida Cruz [a], Werney Ayala Luz Lira [a]

[a] Laboratory of Software Engineering and Industrial Informatics, Federal University of Piauí, BR, 64049-550 Teresina, Brazil
[b] Department of Software Engineering, Blekinge Institute of Technology, SE 371-49 Karlskrona, Sweden

ARTICLE INFO

Article history:
Received 31 December 2015
Received in revised form 14 July 2016
Accepted 12 August 2016
Available online 17 August 2016

Keywords:
Software product lines
Search based software engineering
Search based feature model selection
Feature model selection problem
Product portfolio scoping
NSGA-II
Fuzzy inference systems

ABSTRACT

Software product line (SPL) development is a new approach to software engineering which aims at the development of a whole range of products. However, as long as SPL can be useful, there are many challenges regarding the use of that approach. One of the main problems which hinders the adoption of software product line (SPL) is the complexity regarding product management. In that context, we can remark the scoping problem. One of the existent ways to deal with scoping is the product portfolio scoping (PPS). PPS aims to define the products that should be developed as well as their key features. In general, that approach is driven by marketing aspects, like cost of the product and customer satisfaction. Defining a product portfolio by using the many different available aspects is a NP-hard problem. This work presents an improved hybrid approach to solve the feature model selection problem, aiming at supporting product portfolio scoping. The proposal is based in a hybrid approach not dependent on any particular algorithm/technology. We have evaluated the usefulness and scalability of our approach using one real SPL (ArgoUML-SPL) and synthetic SPLs. As per the evaluation results, our approach is both useful from a practitioner's perspective and scalable.

1. Introduction

A software product line (SPL) consists of a set of software products that share some core functionality – although differ in specific features – and are systematically developed using a collection of reusable assets [1–5]. A family of products associated with an SPL is described by a feature model, which defines the constraints between features (system functionality) and the valid products.

SPL engineering can lead to develop software with less effort and higher quality [6]. However, there are many challenges associated with this approach, such as problems related to feature model selection, feature model construction, SPL architectural improvement, and SPL testing [7].

A lot of attention has be given to the *feature model selection problem* (FMSP) [7,5], in which the objective is to select features and derive products from an SPL's feature model, optimizing different objectives (e.g. profitability), subjected to different types of restrictions (e.g. budget).

To maximize profitability, it is fundamental to define which products should be derived from an SPL and the features that each product should provide, which is know as *product portfolio scoping* (PPS) [8]. Note that PPS is dependent on feature model selection and therefore is affected by the FMSP.

In general, PPS is driven by aspects like product cost and customer satisfaction [6]. Customer satisfaction has been successfully analyzed in terms of segments [9]. Customer segmentation is a marketing strategy that divides a target market in terms of customers subsets (segments), which are made by clustering customers with similar needs.

When using customer segmentation, the defined customer clusters are used as one of the inputs of the feature selection process and the derivation of new products. The suitability of a product regarding a particular customer segment can be measured using the concept of customer satisfaction, which represents how good a product matches the needs of a particular customer segment. The bigger is the customers' satisfaction, the more suitable is the product regarding the customers' segment. Since it is not possible to address all customers' segments in a profitable way, product cost and attractiveness (selling potential) are also aspects that should

* Corresponding author.
E-mail addresses: pasn@ufpi.edu.br (P.d.A. dos Santos Neto), rbritto@ufpi.edu.br, rbr@bth.se (R. Britto), ricardoalr@ufpi.edu.br (R.d.A.L. Rabêlo), jonathas_jivago@hotmail.com (J.J.d.A. Cruz), werney.zero@gmail.com (W.A.L. Lira).

http://dx.doi.org/10.1016/j.asoc.2016.08.024
1568-4946/

be accounted for when performing PPS. The cost of a particular product is tightly related to the design of the whole SPL.

Defining a product portfolio (carrying out PPS) accounting for several different variables (e.g. customers satisfaction and product cost/attractiveness aspects) – which very often lead to conflicting interests – is an NP-hard problem [9]. This kind of problem has been recently addressed by research in a subarea of software engineering known as search-based software engineering (SBSE) [10]. SBSE refers to the usage of computational search to solve (optimize) software engineering problems.

In our previous work [11], we presented an initial proposal to solve the feature model selection problem – and thus supporting PPS – by combining NSGA-II algorithm [12] with Mamdani fuzzy inference systems [13] (hybrid approach). The proposal in our previous work had three main drawbacks: (i) The previous proposal has a severe associated construct validity threat, which is due to the difficulty to obtain from SPLs the required data to run our approach; (ii) We only used a "toy" example to validate our approach and only class level correlation between features and assets was used; (iii) We did not conducted any statistical analysis of the results returned by the NSGA-II algorithm, as suggested by Arcuri and Briand [14].

Therefore, herein we put forward an extension of our previous work, which focused on improving the aforementioned issues. More specifically, the main differences between the two studies, and consequently the main contributions of this paper, are as follows:

- The previous version of our hybrid approach was dependent on the employed technologies and algorithms. Herein we reformulate our approach, detaching it from any particular technique/algorithm.
- We improved the data collection process associated with our approach, to facilitate its usage and to mitigate associated construct validity threats.
- We conducted a more comprehensive evaluation, which focused on show the real word usefulness and scalability of our hybrid approach. To validate the usefulness of our approach, we used the SPL associated with a tool broadly employed in both industrial and academic contexts (ArgoUML-SPL). We evaluated the scalability of our approach using a synthetic SPL. Note that in both cases we followed the guidelines by Arcuri and Briand [14] to evaluate the statistical significance of our results.

The remainder of this work has the following structure: Section 2 presents the related work, followed by relevant background information in Section 3. Section 4 states the problem addressed by this paper. Section 5 presents the research design and methodology employed herein. Section 6 presents the improved hybrid approach. Section 7 contains the results of the performed evaluation. Section 8 discusses the validity threats associated with this work. Finally, Section 9 presents the conclusions and directions for future work.

2. Related work

According to Harman et al. [7], the FMSP is isomorphic to a widely investigated software engineering (SE) problem know as the *next release problem* (NRP) [15], in which the objective is to select a subset of requirements optimizing aspects such as cost, value and stakeholder's importance for a company under constraints. We believe that literature on both problems are related to the proposal put forward herein. Thus, in the remainder of this section we discuss relevant work related to the aforementioned problems and also related to product portfolio scoping.

2.1. Feature model selection problem

Recently, a literature survey [7] and a systematic mapping study [5] on the state-of-the-art SPL-SBSE research were published, which show that from 2008 onwards there is a growing interest in this research area. We select some papers from these two secondary studies to be discussed herein.

The FMSP was first addressed by White et al. [16,17], who formulated the problem as the multidimensional/multiple choice knapsack problem; a constrained single objective formulation that was subjected to a budget constraint. They initially solved the problem using an exact optimization algorithm (Branch and Bound) and afterwards added a heuristic to reduce the associated search space.

Shi et al. [18] and Guo et al. [19] employed a formulation similar to the one employed by White et al. [17]. While the first used greedy search to solve the problem, the second employed genetic algorithms.

Muller [9] used a single objective formulation to optimize SPL's product portfolio in a value-based manner. To solve the problem, the employed the simulated annealing algorithm.

Sayyad et al. presented a first attempt to address the FMSP problem using a multiobjective formulation [20], in which the objective was to optimize total cost, defects, violation and total number of features offered by products. They employed IBEA algorithm to solve the problem. The authors later on improved the performance of their approach by introducing additional heuristics [21,22].

Olaechea et al. [23] employed a multiobjective formulation, aiming at optimize product's cost and associated resources. They solved the problem using an exact algorithm (GIA) and an approximate algorithm (IBEA), and compared the pros and cons of each approach.

2.2. Next release problem

The term next release problem was coined by Bagnall et al. [15] and refers to the problem of planning releases, which means selecting the requirements that are to be delivered in a particular release. The authors have used different metaheuristics algorithms to solve the problem, like greedy algorithms and simulated annealing. They used a single objective formulation to solve the problem, wherein the objective was to identify a stakeholders' subset whose requirements were to be satisfied, maximizing the stakeholders' importance to a company under resource constraints.

Ruhe et al. [24] approached the NRP in a different way, in which they aimed at optimizing software release planning by balancing the required and available resources, accounting for the stakeholders' priorities. They employed a single objective formulation, using genetic algorithms to address the problem.

Harman et al. [25] and Baker et al. [26] addressed the NRP by ranking and selecting candidate software components. They used a single objective formulation and applied greedy and simulated annealing algorithms to solve the problem.

Del Sagrado et al. [27] employed the ant colony algorithm to solve the NRP, optimizing the development effort associated to the requirements and customer's satisfaction.

A lot of attention has been given to address the NRP using multiobjective formulations (multiobjective next release problem – MONRP); it may be the case that single objective formulations achieve the maximization of one concern at the expense of the potential maximization of others [10].

Zhang et al. [28] presented a first attempt to solve the NRP using a multiobjective formulation. They employed the NSGA-II algorithm to optimize requirement's value and cost.

Finkelstein et al. [29] also employed a multiobjective formulation, but they focused on optimizing fairness in requirements assignments' results. They also used NSGA-II to solve the formulated problem.