Information and Software Technology 56 (2014) 1144–1165

Contents lists available at ScienceDirect

Information and Software Technology

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

Toward automated feature model configuration with optimizing non-functional requirements

Mohsen Asadi [a,*], Samaneh Soltani [a], Dragan Gasevic [a,b], Marek Hatala [a], Ebrahim Bagheri [c]

[a] Simon Fraser University, 8888 University Dr, Burnaby, BC V5A 1S6, Canada
[b] Athabasca University, 1 University Drive, Athabasca, AB T9S 3A3, Canada
[c] Ryerson University, 350 Victoria St, Toronto, ON M5B 2K3, Canada

ARTICLE INFO

Article history:
Received 15 January 2013
Received in revised form 10 January 2014
Accepted 4 March 2014
Available online 28 March 2014

Keywords:
Software product lines
Feature model configuration
Stakeholders' preferences
Non-functional interdependencies

ABSTRACT

Context: A software product line is a family of software systems that share some common features but also have significant variabilities. A feature model is a variability modeling artifact, which represents differences among software products with respect to the variability relationships among their features. Having a feature model along with a reference model developed in the domain engineering lifecycle, a concrete product of the family is derived by binding the variation points in the feature model (called configuration process) and by instantiating the reference model.
Objective: In this work we address the feature model configuration problem and propose a framework to automatically select suitable features that satisfy both the functional and non-functional preferences and constraints of stakeholders. Additionally, interdependencies between various non-functional properties are taken into account in the framework.
Method: The proposed framework combines Analytical Hierarchy Process (AHP) and Fuzzy Cognitive Maps (FCM) to compute the non-functional properties weights based on stakeholders' preferences and interdependencies between non-functional properties. Afterwards, Hierarchical Task Network (HTN) planning is applied to find the optimal feature model configuration.
Result: Our approach improves state-of-art of feature model configuration by considering positive or negative impacts of the features on non-functional properties, the stakeholders' preferences, and non-functional interdependencies. The approach presented in this paper extends earlier work presented in [1] from several distinct perspectives including mechanisms handling interdependencies between non-functional properties, proposing a novel tooling architecture, and offering visualization and interaction techniques for representing functional and non-functional aspects of feature models.
Conclusion: our experiments show the scalability of our configuration approach when considering both functional and non-functional requirements of stakeholders.

1. Introduction

Software Product Lines Engineering (SPLE) aims at developing a set of software systems that share common features and satisfy the requirements of a specific domain [2]. SPLE decreases development costs and time to market, and improves software quality through strategic reuse of assets within a domain of interest. A technique adopted in SPLE for managing reusability is commonality and variability modeling through which common assets and their variabilities are formalized.

A software product line lifecycle encompasses a *domain engineering* process and an *application engineering* process. In the *domain engineering* process, a comprehensive formal representation of the products of the domain is developed. This includes a variability model and the core assets of the product family. *Feature models* are among the prevalent variability modeling techniques in SPLE and represent variability in terms of the differences between the features of the products that belong to a software family. A feature is a logical unit of behavior specified by a set of functional and non-functional requirements [3].

On the other hand, the *application engineering* process is responsible for capturing the requirements of the target application, deriving a concrete product from the variability model through a configuration process, and deploying the product into users'

* Corresponding author. Tel.: +1 7789958166.
E-mail addresses: masadi@sfu.ca, mohsen.asadi62@gmail.com (M. Asadi).

http://dx.doi.org/10.1016/j.infsof.2014.03.005
0950-5849/

environment [4]. Using feature models as variability modeling tools, the configuration process selects a suitable set of features to satisfy the stakeholders' requirements.

1.1. Open problem

The focus of existing research in software product lines has been mostly on modeling and managing product lines, while there are only a few works which in particular concentrate on the effective utilization of product lines (also known as the *configuration problem*) during application engineering. In the configuration problem, an application engineer receives a feature model and the target application requirements and attempts to select a subset of the features that optimize the requirements. Solving the configuration problem is challenging for an application engineer due to following reasons:

- There are several types of variability relations and integrity constraints between the features [2].
- The number of possible configurations has exponential growth. Even in small feature models the number of possible configurations can be very high. Industrial feature models may consist of hundreds of features which increases the complexity of feature model configuration [5].
- Features may have either positive or negative impact on the different business concerns of a product, and hence expose different quality attributes [6]. We refer to business concerns of a product (e.g., *security* and *customer satisfaction*) and quality attributes of a product (e.g., *performance* and *cost*) as non-functional properties (NFP). For example, a feature may have a negative impact on *security*, but a positive impact on *customer satisfaction* or it could have *high performance* but *low reliability*.
- In addition to functional requirements, stakeholders may have several constraints and preferences over non-functional properties during the product derivation. For example, one stakeholder may ask for a product with *high security*, *high customer satisfaction*, and specific *cost*; and can mention that *customer satisfaction* is more important than *security*; which would make the configuration process more difficult and complex [5,7].
- Finally, there are interdependency relations between non-functional properties such that an increase or decrease in the value of one non-functional property may lead to an increase or decrease in the value of another non-functional property. For example, increase in the value of *security* may decrease the value of *performance* for the stakeholders [8].

There are a number of algorithms for product line configuration [9–12], which aim at assisting application engineers in solving the configuration problem. However, to our knowledge, there has been little coverage for the consideration of non-functional requirements in the configuration algorithms. Some techniques have addressed the configuration problem by transforming the feature model configuration problem into a Constraints Satisfaction Problem (CSP), and have used CSP-solvers to build optimal configurations [5,13]. The main problem with these techniques is time inefficiency. Other techniques solve this problem by applying approximation algorithms, but their final configurations are only partially optimal [14,15]. To our knowledge, almost all these works except [13] only support limited types of NFPs (i.e., quantitative NFPs such as footprint and cost) and do not consider qualitative NFPs (e.g., security). Moreover, no work has considered the preferences of stakeholders in terms of the relative importance between non-functional properties in the process of feature model configuration; and relative importance varies depending on the stakeholders standpoint and application domain [16]. Relative importance of non-functional properties is especially important for the stakeholders and software designers who are able to define the relative importance among the available functional and non-functional options but have difficulty in deterministically picking their choice from those options [16]. Thus, a product line configuration technique should not only be able to operate over deterministic functional choices, but should also be able to operate when the relative importance between both functional and non-functional properties are given. Finally, to the best of our knowledge, no feature model configuration approach has taken interdependencies between non-functional properties into account.

1.2. Contribution and approach overview

Existing challenges in the configuration problem faced by application developers motivated us to develop an automated method for selecting a set of features that would fulfill both the stakeholders' functional and non-functional requirements and preferences. To this end, we adopted and integrated the *Analytical Hierarchy Process (AHP)* [17], *Fuzzy Cognitive Map (FCM)* [18], and *Preference-based Planning* techniques [19].

AHP is a well-known pairwise comparison method used to calculate the relative ranking of different options based on stakeholders' judgments [17,20]. FCM [18] is an extension of cognitive maps which incorporates fuzzy causal functions to represent fuzzy relations among objects in a complex system. FCM have been widely used several domains for modeling and decision making [21–23]. Hierarchical Task Network (HTN) planning is a popular planning technique, which is suited for domains with hierarchical task decomposition [24,19]. The HTN Planning technique generates plans from a developed hierarchical network of domain tasks and actions [25].

The general overview of the proposed approach is illustrated in Fig. 1. As shown in the figure, our approach captures functional requirements and non-functional requirements of the stakeholders for a final application. Non-functional requirements are captured in terms of relative importance of non-functional properties along with constraints over the non-functional properties (Section 3.3). Afterward, we employ an Analytical Hierarchy Process (AHP) [17] to calculate the local weights of non-functional properties. To incorporate the value related interdependencies during the feature selection, we employ Fuzzy Cognitive Map (FCM) to compute the overall influence between non-function properties and then calculate the global non-functional properties weights by integrating result of AHP and FCM (Section 4.1.1). Next, ranks of the features based on the importance of non-functional properties and their values assigned to the features is obtained via a utility function defined in Section 4.1.2.

We generate the HTN planning domain and problem from the feature model and stakeholders' requirements (Section 4.2). First, preprocessing steps are preformed to handle optional and OR relation for transformation and then we produce domain predicates, operators, and tasks according to the proposed transformation rules. We apply SHOP2 planner [26], an HTN-based planning system widely used for planning problems, to identify an optimal plan (i.e., a plan with the best overall cost). To produce the final configuration, the features chosen by the SHOP2 planner are selected and represented in the visual view of our tool.

The main contributions of this paper are as follows:

- Proposing a complete approach for configuring feature models which includes an easy-to-understand formalism for capturing the stakeholders' preferences over non-functional properties represented in terms of relative importance; utilizing the analytic hierarchy process and fuzzy cognitive maps to calculate the weightes of NFPs; transforming a feature model and