Journal of Logical and Algebraic Methods in Programming 84 (2015) 640–667

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Automatic promotional specialization, generalization and analysis of extended feature models with cardinalities in Alloy

Shohreh Ajoudanian [a,*], Seyed-Hassan Mirian Hosseinabadi [b]

[a] Department of Computer Engineering, Islamic Azad University Science and Research Branch, Tehran, Iran
[b] Department of Computer Science, Sharif University of Technology, Tehran, Iran

A R T I C L E   I N F O

Article history:
Received 5 February 2014
Received in revised form 15 November 2014
Accepted 29 November 2014
Available online 8 December 2014

Keywords:
Extended feature model with cardinality
Specialization and generalization of SPLS
Multiple multi-level promotions in Alloy

A B S T R A C T

Software product line engineering is a method of producing a set of related products that share more commonalities than variability in a cost-effective approach. Software product lines provide systematic reuse within a product family. Extended feature models with cardinalities are widely used for managing variability and commonality in the software product line domains. In this paper, we use promotion technique in Alloy to formalize constraint based extended feature models with cardinalities and their specialization and generalization. This technique has a significant influence on applying analysis operations on feature models. To show the benefits of the promotion technique, we calculate the reuse ratio of a feature in a large scale software product line. In the presented method, in addition to feature and group cardinalities, we consider different combinations of cardinalities with each other as well as feature cloning.

1. Introduction

Software product line (SPL) engineering is about producing a set of related products that share more commonalities than variability. Feature models are widely used for managing variability and commonality in software product lines [1]. Extended feature models with cardinalities (Ext-FM-WC) are models that comprise more information than features and relationships between them. These models include information about feature attributes and the relationships between features and attributes.

Analyzing this kind of models is an error-prone and tedious task, and is infeasible to carry out manually for large-scale feature models. As a result, the automated analysis of feature models is an active area of research in the SPL community [2]. In the scientific community, FMs are the de-facto standard representing the variability of an SPL, but in contrast, these models are rarely used in the industrial practice [32,33]. One reason is that feature models – as static models that only show the commonality and variability of the SPL – are not very beneficial in the industry. For modeling realistic SPLs, there must be a way of representing the evolution of the SPLs and their models over time. The features or the attributes of a model can be added to the model or deleted from it. The cardinalities can be updated in the model. We believe editing operations of a feature model are the alternative active area of research in the SPL community. Various industrial experiences have shown that one of the most efficient tools for making FMs intuitive and concise are feature attributes [1]. The number of features

* Correspondence to: Office number 317, Floor 3, Computer Engineering Department, Islamic Azad University of Najafabad, Najafabad, Isfahan, Iran. Tel.: +98 (0)31 42291111.

E-mail addresses: shajoudanian@pco.iaun.ac.ir (S. Ajoudanian), hmirian@sharif.edu (S.-H.M. Hosseinabadi).

http://dx.doi.org/10.1016/j.jlamp.2014.11.005
2352-2208/

have reduced with the use of attributes. For example, in [31] with the use of attributes, the number of features reduced from 189 to 59. In this paper, we use feature models with attributes.

The main contributions of this paper are:

1. Formalizing constraint based Ext-FM-WC with the help of the promotion technique in Alloy;
2. Formalizing specialization and generalization of Ext-FM-WC;
3. Increasing the speed of calculating process of some of the analysis operations of the feature models with the help of the promotion technique in Alloy.

The remainder of the paper is structured as follows: Section 2 presents an overview of related work. Section 3 presents basics of an Ext-FM-WC, editing operations, essential constructs of Alloy modeling language, and promotion pattern. In Section 4, we introduce the formalizing structure of our approach and provide an example of formalized Ext-FM-WC with our proposed promotional technique. Section 5 discusses the results obtained. Finally, Section 6 presents the conclusions.

2. Related work

An SPLS (Software product line system) provides a systematic reuse paradigm [7–9]. From common assets on which different programs of a domain can be assembled. SPLs and their feature models evolve over time. For example, a feature can be added to the model or an existing feature can be deleted from the model. The authors of [10] classified the evolution of feature models via a series of modifications such as refactoring, specialization, generalization, and arbitrary editing. They proposed an algorithm for reasoning about feature model modifications to help designers to determine how the program membership of the SPL has changed.

In [11], the traditional notion of refactoring has been extended in the SPL context. Authors have defined a feature model refactoring a transformation that improves the quality of a feature model by improving (maintaining or increasing) its configurability. In this paper, a set of refactoring operations are presented for FMs, for example, how to convert an *Alternative* relation to an *Or* relation in a feature model.

A feature assembly framework has been presented in [12], which is a new approach for creating feature models through the feature composition and the feature assembly. Furthermore, it promotes feature reusability by storing features in a so-called feature pool, which acts as a feature repository. They used a hybrid methodology that combines both a top down and a bottom up approach for modeling features in a product line to overcome the problem of scalability and abstraction mechanisms for feature models.

Czarnecki et al. [13] provided a formalization of cardinality-based feature modeling and its specialization. They translated cardinality-based feature models into context free grammar. However, the translated feature model did not support attributes, and global constraints.

The authors of MUSCLES [14] presented a formal model of multi-step SPL configuration. The MUSCLES framework used the simple feature model and its focus is on constraints optimization in a feature model.

To our knowledge, [15] is the first paper using Alloy for formalizing feature model and some very simple feature model's refactoring. It is a formalization of the simple feature model without attributes and cardinality and complex constraints.

In [34], the authors proposed a catalogue of 30 visual rules to merge FMs using graph transformations as a suitable technology to automate the merging of FMs. Some of the proposed rules can be used in refactoring operations. Their work does not include feature and group cardinality as an important part of each feature model. Their work proposes a viable approach for merging two FMs with the same parent feature, but their proposal does not provide any solution for editing operations.

In this paper, we present a formalization of the semantics of complex constraint-based extended feature models with cardinality (Ext-FM-WC) using the promotion technique in Alloy. It concentrates on specialization and generalization operations on the Ext-FM-WC. This formalism constructs the cardinalities for features, the cardinalities for groups, the attributes and the hierarchy of the feature model by promoting local state into global state. Constraints (basic and complex) are separately formalized.

Table 1 compares some approaches to feature model refactoring. The criteria are support for (i) attributes; (ii) feature and group cardinalities; (iii) basic constraints such as requires and excludes; (iv) complex constraints such as Boolean constraints involving values of attributes; (v) adding, deleting, or updating a feature to/ from/ in the model; (vi) adding, deleting, or updating an attribute to/ from/ in the model; (vii) adding, deleting, or updating cardinality to/ from/ in the model; (viii) formal semantics.

3. Background

In this section a brief introduction to Ext-FM-WC, editing operations on Ext-FM-WC, Alloy formal language and promotion technique in Alloy is provided.