Journal of Logical and Algebraic Methods in Programming 85 (2016) 1131–1153

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Input–output conformance testing for software product lines ☆

Harsh Beohar, Mohammad Reza Mousavi *,[1]

Centre for Research on Embedded Systems, School of Information Technology, Halmstad University, Sweden

ARTICLE INFO

Article history:
Received 1 April 2014
Received in revised form 23 September 2016
Accepted 24 September 2016
Available online 29 September 2016

Keywords:
Model based testing
Input–output conformance testing
Software product lines
Input–output featured transition systems

ABSTRACT

We extend the theory of input–output conformance (IOCO) testing to accommodate behavioral models of software product lines (SPLs). We present the notions of residual and spinal testing. These notions allow for structuring the test process for SPLs by taking variability into account and extracting separate test suites for common and specific features of an SPL. The introduced notions of residual and spinal test suites allow for focusing on the newly introduced behavior and avoiding unnecessary re-test of the old one. Residual test suites are very conservative in that they require retesting the old behavior that can reach to new behavior. However, spinal test suites more aggressively prune the old tests and only focus on those test sequences that are necessary in reaching the new behavior. We show that residual testing is complete but does not usually lead to much reduction in the test-suite. In contrast, spinal testing is not necessarily complete but does reduce the test-suite. We give sufficient conditions on the implementation to guarantee completeness of spinal testing. Finally, we specify and analyze an example regarding the Ceiling Speed Monitoring Function from the European Train Control System.

1. Introduction

1.1. Motivation

Software product lines (SPLs) have been proposed as a response to the ever-increasing demand for mass production and mass customization of software. Since their introduction, SPLs have gained popularity and have been increasingly used in the practice of software development. Briefly, an SPL consists of a variety of computer systems (products) that are built upon a common base (platform). The products share several core features, but also differ from each other in some features, commonly referred to as variability points.

Testing such SPLs is known to be very challenging due to the large spectrum of variability and the complexity of products. There have been several attempts to provide a structured discipline for testing SPLs. However, it appears from the recent

☆ This article integrates and extends the authors' ACM SAC SVT 2014 and MBT 2014 papers [1,2]. An extended abstract was presented at the 25th Nordic Workshop on Programming Theory, NWPT 2013, in Tallinn.

* Corresponding author.
E-mail addresses: harsh.beohar@hh.se (H. Beohar), m.r.mosuavi@hh.se (M.R. Mousavi).

[1] The work of Mohammad Reza Mousavi has been partially supported by the Swedish Research Council (Vetenskapsradet) award number: 621-2014-5057 (Effective Model-Based Testing of Concurrent Systems) and the Swedish Knowledge Foundation (Stiftelsen for Kunskaps- och Kompetensutveckling) in the context of the AUTO-CAAS HöG project (number: 20140312).

http://dx.doi.org/10.1016/j.jlamp.2016.09.007

surveys [3–7] that several fundamental approaches to model-based testing are not yet fully adapted to and adopted in this domain (also see Section 1.2 for a brief overview of the related work).

The theory of input–output conformance (IOCO) testing [8] is one such fundamental approach that uses labeled transition systems for model-based testing. The *testing hypothesis* of this approach is that the behavior of the implementation under test can be viewed as an (unknown) input–output labeled transition system that is input-enabled, i.e., can accept any input action. We are not aware of any prior work in adapting the theory of IOCO to cater for variability in SPLs. The present paper addresses this gap by extending IOCO to the setting of SPLs.

To this end, we propose input–output featured transition systems (IOFTSs) as simple yet expressive behavioral models of SPLs and adapt the traditional IOCO theory to allow for using IOFTSs (instead of plain input–output transition system models) as test models for model-based testing. Our approach preserves the testing hypothesis of IOCO; although we include more information in our test models to capture the structure of SPLs, the interaction with the system under test only goes via plain input and output actions and the internal structure of the product is not revealed during the test execution. We define the test suite and the test cases that are generated from an IOFTS, which can be used for checking conformance. Furthermore, we define two notions of refinement, one at the level of IOFTSs and another one at the level of test suites, which allow for focusing on particular sets of features and eventually on a particular product. We show that these two refinements interact nicely, in that they lead to the same set of test cases. The techniques proposed in this paper are rather generic and we believe these techniques can be adapted to other model-based testing theories (such as those proposed in [9–12]).

In addition, we take first step towards an efficient and coordinated test process for applying IOCO to SPLs. To this end, we develop a theoretical framework of residual and spinal test suites. Intuitively, both residual and spinal test suites are IOFTSs (whose underlying graph is tree-like), which allow one to test the common features once and for all, and subsequently, only focus on the specific features when moving from one product configuration to another. However, they differ in their testing power and efficiency: testing power refers to the possibility of rejecting non-conforming implementations (ideally a test suite is complete, i.e., it can reject each and every non-conforming implementation by generating at least one failing test case), and efficiency refers to the size of the test-suite. On one hand, spinal test suites have strictly less testing power than residual test suites; on the other hand, spinal test suites produce more compact test cases when compared to test cases produced by residual test suites. We show that residual test suites are complete, i.e., for each product it is always sufficient to use the residual test suite with respect to the features present in the afore-tested products, whereas spinal test suites are not necessarily complete. Lastly, we also show that spinal test suites are exhaustive, i.e., they reject each and every non-conforming implementation under test, when the implementation satisfies the *orthogonality criterion*. This is a rather mild criterion, which implies that old features are not capable of disabling any enabled behavior from the new features on their own and without involving any interaction with the new feature's components.

The proposed theory is the first step towards a feature-based analysis [13] of SPLs based on the IOCO theory. For example, once a feature (combination) selection criterion is fixed, one can use the spinal testing method to focus test those features (feature combinations) in a selection of concrete products.

1.2. Related work

Various attempts have been made regarding formal and informal modeling of SPLs, of which [14–18] provide comprehensive surveys. By and large, the literature can be classified into two categories: structural modeling and behavioral modeling techniques.

Structural models specify variability in terms of presence and absence of features (assets, artifacts) in various products and their mutual inter-relations. Behavioral models, however, concern the working of features and their possible interactions, mostly based on some form of finite state machines or labeled transition systems. The main focus in behavioral modeling of SPLs (cf. [19–25]) has been on formal specification of SPLs and adaptation of formal verification (mostly model checking) techniques to this new setting. Our notion of input–output featured transition system is a slight extension of featured transition system [21]. There are few alternatives to FTSs that could be used as behavioral test models for model-based testing [26]. Such models include the extensions of process algebras – [24,27] and Petri nets with features [28], modal transition systems [29,23] and higher-level models such as UML state- and sequence-diagrams [30–32].

In this paper, we assume a predefined structure of the SPL in terms of a feature diagram. The structural information in the feature diagram is used to annotate the behavioral model and steer the test process. An alternative approach to specifying and programming SPLs is the delta-oriented approach [33,11,34,12,35], where the SPL is specified in terms of additions to, removals from, or modifications of the core product. Although our work is based on input–output featured transition systems, we envisage that the ideas pursued in this paper can be adapted to other behavioral test models and to other conformance testing theories, such as those on finite state machines [10,9] and on delta-oriented methods. For example, recently, in [12,35] related techniques have been explored in the area of delta-oriented SPL models. For higher-level modeling frameworks, our input–output featured transition systems can serve as a semantic domain; this way, our techniques can be applied to higher-level modeling and specification languages (such as UML state diagrams or domain specific languages).

Several testing techniques have been adapted to SPLs, of which [5,6,3,4] provide recent overviews. Hitherto, most fundamental approaches to formal conformance testing [10] have not been adapted sufficiently to the SPL setting. The only