Journal of Logical and Algebraic Methods in Programming 85 (2016) 287–315

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Modelling and analysing variability in product families: Model checking of modal transition systems with variability constraints

Maurice H. ter Beek [a,*], Alessandro Fantechi [a,b], Stefania Gnesi [a], Franco Mazzanti [a]

[a] Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", CNR, Via G. Moruzzi 1, 56124 Pisa, Italy
[b] Dipartimento di Sistemi e Informatica, Università di Firenze, Via S. Marta 3, 50139 Firenze, Italy

ARTICLE INFO

Article history:
Received 22 May 2015
Received in revised form 28 October 2015
Accepted 11 November 2015
Available online 30 November 2015

Keywords:
Model checking
Modal transition systems
Temporal logic
Product families
Variability

ABSTRACT

We present the formal underpinnings of a modelling and analysis framework for the specification and verification of variability in product families. We address variability at the behavioural level by modelling the family behaviour by means of a Modal Transition System (MTS) with an associated set of variability constraints expressed over action labels. An MTS is a Labelled Transition System (LTS) which distinguishes between optional and mandatory transitions. Steered by the variability constraints, the inclusion or exclusion of labelled transitions in an LTS refining the MTS determines the family's possible product behaviour. We formalise this as a special-purpose refinement relation for MTSs, which differs fundamentally from the classical one, and show how to use it for the definition and derivation of valid product behaviour starting from product family behaviour. We also present a variability-aware action-based branching-time modal temporal logic to express properties over MTSs, and demonstrate a number of results regarding the preservation of logical properties from family to product behaviour. These results pave the way for the more efficient family-based analyses of MTSs, limiting the need for product-by-product analyses of LTSs. Finally, we define a high-level modal process algebra for the specification of MTSs. The complete framework is implemented in a model-checking tool: given the behaviour of a product family modelled as an MTS with an additional set of variability constraints, it allows the explicit generation of valid product behaviour as well as the efficient on-the-fly verification of logical properties over family and product behaviour alike.

1. Introduction

Software Product Line Engineering (SPLE) [31,61] is by now a full-fledged software engineering approach aimed at developing, in a cost-effective manner, a family of software-intensive systems by systematic reuse. Individual products share an overall reference model or architecture of the product family, but they differ with respect to specific features. A feature is

* Corresponding author.
E-mail addresses: maurice.terbeek@isti.cnr.it (M.H. ter Beek), alessandro.fantechi@unifi.it (A. Fantechi), stefania.gnesi@isti.cnr.it (S. Gnesi), franco.mazzanti@isti.cnr.it (F. Mazzanti).

http://dx.doi.org/10.1016/j.jlamp.2015.11.006
2352-2208/

a (user-visible) increment in functionality of a product. SPLE reduces time-to-market, increases product quality, and lowers production costs. The common and variable parts of products are defined in terms of features and managing variability is about identifying the variation in a shared family model to encode exactly which combinations of features constitute valid products. The actual configuration of products during application engineering is then reduced to selecting desired options in the variability model. Hence, the overall production process is organised so as to maximise commonalities and at the same time minimise the cost of variations.

Variability modelling and analysis of software-intensive systems traditionally focuss on structural rather than behavioural properties and constraints. It is important to model and analyse their variability also at the behavioural level, in order to provide a form of quality assurance. This was first perceived in the context of UML 2.0 [46,69]. Consequently, Modal Transition Systems (MTSs) were recognised in [40] as a promising model to describe in a compact way all possible operational behaviour of the products of a product family. In a nutshell, an MTS [1] is an LTS that distinguishes between admissible ('may') and necessary ('must') transitions. Following [40], variants and extensions of MTSs were studied in order to elaborate a suitable formal modelling structure to describe variability in terms of behaviour, including modal I/O automata [52], variable I/O automata [55], and MTSs with logical variability constraints [4,5,36]. This triggered a growing interest in modelling behavioural variability, which led to the application of formal models different from MTSs, but still with a transition system semantics, including first and foremost the highly elaborated framework based on featured transition systems [26,27], but also process-algebraic approaches [12–14,42,45,67], Petri nets [59], and finite state machines [57]. As a result, behavioural analysis techniques like model checking [6,23] became available for the verification of (temporal) logic properties of product families, resulting in special-purpose model checkers [15,16,25,32].

In this paper, we focus on one such approach. We present the full formal underpinnings of a modelling and analysis framework, some of whose aspects were introduced in [4,5,15,16]. It is based on a specific subset of MTSs, whose elements are equipped with an additional set of logical variability constraints expressed over actions. These constraints allow one to capture all common variability notions known from feature models [43,49,63], since it is well-known that plain MTSs cannot efficiently (in a compact way) model, e.g., the notions of alternative and mutually exclusive features. Considering the may transitions as optional and the must transitions as mandatory, an MTS can be interpreted as a family of LTSs such that each family member corresponds to a specific selection of optional transitions. In this way, a single MTS can model a product family since it allows a compact representation of the family's behaviour, by means of states and actions, shared by all products, and variation points, by means of may and must transitions, used to differentiate among products. A specific selection of labelled transitions respecting the variability constraints expressed over action labels, determines possible product behaviour (modelled as LTSs). In this paper, we formalise this for the first time as a special-purpose refinement relation for MTSs, which differs fundamentally from classical MTS refinement, and we show how to use it to formally define and derive valid (i.e., configurable) product behaviour starting from the behaviour of a product family.

Subsequently, we recall v-ACTL, a variability-aware action-based branching-time modal temporal logic that was introduced with the sole purpose of reasoning over the syntactic structure of MTSs [15]. To this aim, it provides novel interpretations of some classical modal and temporal operators. In this paper, we demonstrate a number of results regarding the preservation of specific v-ACTL properties from family to product behaviour. These results pave the way for family-based analyses of MTSs, limiting the need for enumerative product-by-product analyses of LTSs. Based on model-checking techniques for v-ACTL, we updated the Variability Model Checker VMC [15,16] in such a way that it now allows one to perform two kinds of behavioural variability analysis on a product family modelled as an MTS with additional variability constraints:

1. The actual set of valid product behaviour can explicitly be generated from the MTS, after which the MTS and the resulting LTSs can independently be verified against a logical property;
2. A logical property can be verified directly over the MTS, relying on the fact that under certain syntactic conditions, validity over the MTS guarantees validity of the same property for all the family's valid products (LTSs).

Finally, we formally define the modal process algebra that VMC accepts as specification of an MTS and we illustrate the applicability of the overall framework and its associated tool by means of an example family of coffee machines.

The paper is based on previous publications (in particular, [4,5,15,16]), but it contains new material and results. The formal definition of refinement of MTSs has been completely revisited for the specific purpose of defining and deriving LTSs modelling valid product behaviour, which has required the introduction of new notions like consistent and valid products. The preservation by refinement of v-ACTL formulae has been formally defined and proved. The full syntax and semantics of the high-level modal process algebra used to specify MTSs in VMC has been defined and the associated set of variability constraints may now contain any Boolean expression over the action labels.

The paper is organised as follows. After presenting a running product family example in Section 2, we introduce the modelling framework of MTSs with additional variability constraints in Section 3. In Section 4, we define the action-based branching-time modal temporal logic, v-ACTL, for the formulation and analysis of behavioural variability in MTSs. The associated model-checking tool, VMC, for the specification and verification of behavioural variability in product families modelled as MTSs is described in Section 5. Section 6 discusses related models and tools and, finally, Section 7 contains some concluding remarks.