Static Analysis and Family-based Model Checking with VMC

Maurice H. ter Beek
Franco Mazzanti
ISTI–CNR, Pisa, Italy

Ferruccio Damiani
Luca Paolini
Giordano Scarso
University of Turin, Italy

Michael Lienhardt
ONERA, Palaiseau, France

ABSTRACT

VMC is a research tool for on-the-fly model checking variability-rich behavioural models in the form of modal transition systems with variability constraints. In this tutorial, we will introduce a tool chain built around VMC that allows to perform an efficient kind of family-based model checking of such systems in absence of deadlocks. It accepts as input either such a modal transition system or a featured transition system. A featured transition system can efficiently be checked for ambiguities (dead or false optional transitions and hidden deadlock states) and possible ambiguities can be removed. A featured transition system without hidden deadlocks can be transformed into a modal transition system to make it amenable to the efficient kind of family-based model checking.

CCS CONCEPTS

• **Software and its engineering → Software product lines**; **Formal methods**; **Model checking**; **Automated static analysis**.

KEYWORDS

VMC, tool support, configurable systems, software product lines, variability models, featured transition systems, modal transition systems, static analysis, model checking

ACM Reference Format:
Maurice H. ter Beek, Franco Mazzanti, Ferruccio Damiani, Luca Paolini, Giordano Scarso, and Michael Lienhardt. 2021. Static Analysis and Family-based Model Checking with VMC. In *25th ACM International Systems and Software Product Lines Conference (SPLC'21), 06–11 September, 2021, Leicester, UK*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/...

1 TOPIC

Formal models of software product line (SPL) behaviour have been studied extensively throughout the last decade. Such variability-rich behavioural models are typically based on the superimposition in one single labelled transition system (LTS) equipped with feature-based variability (a family model) of multiple LTSs, each of which is a semantic representation of a different variant configuration (a product model). Arguably the best known models are featured transition systems (FTSs) [8] and modal transition systems with variability constraints (MTSυ) [3]. An FTS is an LTS whose action-labelled transitions are also labelled with feature expressions that condition the presence of transitions in product models. An MTS is an LTS that distinguishes admissible ('may'), necessary ('must'), and optional (may but not must) transitions, while an MTSυ is an MTS with an additional set of logical variability constraints that resemble the feature expressions of FTSs.

The automated analysis of variability models, such as detecting anomalies like dead or false optional features in feature diagrams, has been studied for decades [7]. In [1], an efficient automated static analysis of FTSs was introduced, capable of detecting behavioural counterparts of such anomalies, like dead transitions that cannot be executed in any product model or, on the contrary, false optional transitions that can be executed in all product models, but also hidden deadlock states that do not deadlock in the family model whereas they do in some product models. The criteria for such ambiguities are expressed as propositional formulae, thus reducing ambiguity detection to SAT solving. An FTS can be disambiguated to improve the clarity of the model and to facilitate a kind of family-based analysis.

VMC[1] [4, 6] is a research tool for explicit-state on-the-fly model checking of MTSυ properties expressed in a dedicated variability-aware action-based and state-based branching-time temporal logic v-ACTL [3], which is an action-based version of the well-known logic CTL. VMC offers both product-based model checking, upon explicit generation of the product models, and a kind of family-based model checking. The latter relies on the fact that for properties expressed in v-ACTLive$^{\Box}$, which is a rich fragment of v-ACTL interpreted on MTSs without deadlocks, validity for the family model guarantees validity of the property for all product models (cf. [3, Theorem 4]). FTS4VMC [5] is a recently developed front-end for VMC which allows to analyse an FTS for ambiguities (dead or false optional transitions and hidden deadlock states), disambiguate an ambiguous FTS, transform an FTS into an MTS, and perform an efficient kind of family-based model checking of an MTS obtained in this way from an FTS without hidden deadlock states, by calling VMC. The FTS4VMC implementation is publicly available from https://github.com/fts4vmc/FTS4VMC.

In this tutorial, the participants are introduced to the field of behavioural variability modelling and analysis; they are made familiar with FTSs, MTSs, and family-based analysis of such models. After explaining ambiguities in FTSs, the participants are introduced to SAT solving and they are shown how to use it for static analysis of FTSs to detect ambiguities. Finally, a tool chain built around VMC is presented and it is demonstrated how to use VMC to perform an efficient kind of family-based model checking of deadlock-free FTSs and MTSs. It is also demonstrated how to use FTS4VMC to detect and remove ambiguities in an FTS, and to transform the resulting FTS into an MTS to make it amenable for analysis with VMC.

[1] http://fmt.isti.cnr.it/vmc

SPLC'21, 06–11 September, 2021, Leicester, UK

https://doi.org/10.1145/...

This tutorial is targeted towards an academic audience interested in the analysis of behavioural variability models; no specific prior experience is required.

As an outcome, the participants of this tutorial will understand the capabilities of the tool chain built around VMC and gain a basic practical understanding of how to use it to specify and analyse behavioural variability models, ranging from the detection of ambiguities by means of SAT solving to family-based model checking.

2 PLAN

This half-day tutorial will cover the following topics:

- Behavioural variability modelling and analysis (1 hour)
  - Featured Transition System (FTS)
  - Modal Transition System with variability constraints (MTSv)
  - Family-based model checking, temporal logic properties, VMC
- Automated static analysis of FTSs (1 hour)
  - Ambiguities: dead/false optional transitions, hidden deadlocks
  - Detecting ambiguities: criteria, SAT solving, implementation
  - Disambiguating FTSs: benchmark experiments
- Demonstrations of tool chain at work (1 hour)
  - VMC
  - FTS4VMC

This tutorial will be conducted by a combination of lectures and tool demonstrations. The material will be specifically prepared in case of acceptance, based on existing lecturing material (e.g., on behavioural variability modelling and analysis, SAT solving, and model checking and temporal logic) in combination with new material based on [1, 2, 5].

3 PRESENTERS' BACKGROUNDS

Maurice ter Beek is a researcher at ISTI–CNR (Pisa, Italy) and head of the Formal Methods and Tools lab. He obtained a Ph.D. at Leiden University (The Netherlands). He authored over 150 peer-reviewed papers, edited over 25 special issues of journals and proceedings, and serves on the editorial board of the Journal of Logical and Algebraic Methods in Programming, Science of Computer Programming, PeerJ Computer Science and ERCIM News. He works on formal methods and model-checking tools for the specification and verification of safety-critical software systems and communication protocols, focussing in particular on applications in service-oriented computing, software product line engineering and railway systems. He is member of the Steering Committees of the FMICS, SPLC and VaMoS conference series, and regular PC member of the FM, FMICS, FormaliSE, SEFM, SPIN, SPLC and VaMoS conference series.

Franco Mazzanti is senior researcher at CNR–ISTI since 2006, and member of the Formal Methods and Tools group. He is the main designer and author of the KandISTI formal verification framework, which includes the explicit, on-the-fly model-checking tools CMC, FMC, UMC and VMC. Author of more than 40 papers in the field of formal methods, his research focusses i) on the design and support of state- and event-based branching-time temporal logics for the specification and the evaluation of system requirements, and ii) on the exploitation of formal methods and tools diversity for the analysis of concurrent, asynchronous systems. He has applied his research in many EU projects, among which AGILE, SENSORIA, ASTRAIL and 4SECURAIL.

Ferruccio Damiani is associate professor at the Computer Science Department of the University of Torino. There he founded and coordinates the MoVeRe (System Modelling, Verification, and Reuse) research group. He is member of the board of the European Association for Programming Languages and Systems (EAPLS) and of the steering committee of the Integrated Formal Methods (iFM) conference series. His current research interests include: computational models and languages; concurrent, distributed, and mobile systems; variability modelling and software product lines.

Luca Paolini is assistant professor at the Computer Science Department of the University of Torino. He is a member of the MoVeRe (System Modelling, Verification, and Reuse) research group. His current research interests include: formal analysis of software product lines, but also the development and foundation of programming languages and innovative and unconventional computing models.

Giordano Scarso is a M.Sc. student at the University of Turin, where he completed his B.Sc. degree in 2020 with a thesis titled *FTS4VMC: a tool for supporting disambiguation and family-based model checking of FTS* under the supervision of Ferruccio Damiani, Maurice ter Beek and Franco Mazzanti.

Michael Lienhardt is a research engineer in the field of applied formal methods in computer science, at the French Space Lab ONERA, where he works on different projects, including the design of a new generation of aerodynamics simulators. His research interests include concurrency, static analysis and software engineering.

4 TUTORIAL BACKGROUND

This tutorial has not been offered previously, but the theoretical underpinnings were presented at SPCL'19 [1], which resulted in a best paper award, and they are to be published in the special issue of *Empirical Software Engineering* on *Configurable Systems* dedicated to SPLC'19 [2]. The tool chain is presented in [5], which is under submission to the Demonstrations and Tools track of SPLC'21.

REFERENCES

[1] Maurice H. ter Beek, Ferruccio Damiani, Michael Lienhardt, Franco Mazzanti, and Luca Paolini. 2019. Static Analysis of Featured Transition Systems. In *Proceedings of the 23rd International Systems and Software Product Line Conference (SPLC'19)*. ACM, 39–51. https://doi.org/10.1145/3336294.3336295

[2] Maurice H. ter Beek, Ferruccio Damiani, Michael Lienhardt, Franco Mazzanti, and Luca Paolini. 2021. Efficient Static Analysis and Verification of Featured Transition Systems. *Empir. Softw. Eng.* (2021). https://doi.org/10.1007/s10664-020-09930-8

[3] Maurice H. ter Beek, Alessandro Fantechi, Stefania Gnesi, and Franco Mazzanti. 2016. Modelling and analysing variability in product families: Model checking of modal transition systems with variability constraints. *J. Log. Algebr. Meth. Program.* 85, 2 (2016), 287–315. https://doi.org/10.1016/j.jlamp.2015.11.006

[4] Maurice H. ter Beek and Franco Mazzanti. 2014. VMC: Recent Advances and Challenges Ahead. In *Proceedings of the 18th International Software Product Line Conference (SPLC'14)*, Vol. 2. ACM, 70–77. https://doi.org/10.1145/2647908.2655969

[5] Maurice H. ter Beek, Franco Mazzanti, Ferruccio Damiani, Luca Paolini, Giordano Scarso, Michele Valfrè, and Michael Lienhardt. 2021. Static Analysis and Family-based Model Checking of Featured Transition Systems with VMC. https://doi.org/10.5281/zenodo.4497888

[6] Maurice H. ter Beek, Franco Mazzanti, and Aldi Sulova. 2012. VMC: A Tool for Product Variability Analysis. In *Proceedings of the 18th International Symposium on Formal Methods (FM'12) (LNCS, Vol. 7436)*, Dimitra Giannakopoulou and Dominique Méry (Eds.). Springer, 450–454. https://doi.org/10.1007/978-3-642-32759-9_36

[7] David Benavides, Sergio Segura, and Antonio Ruiz-Cortés. 2010. Automated Analysis of Feature Models 20 Years Later: a Literature Review. *Inf. Syst.* 35, 6 (2010), 615–636. https://doi.org/10.1016/j.is.2010.01.001

[8] Andreas Classen, Maxime Cordy, Pierre-Yves Schobbens, Patrick Heymans, Axel Legay, and Jean-François Raskin. 2013. Featured Transition Systems: Foundations for Verifying Variability-Intensive Systems and Their Application to LTL Model Checking. *IEEE Trans. Softw. Eng.* 39, 8 (2013), 1069–1089. https://doi.org/10.1109/TSE.2012.86