# Quantitative Variability Modeling and Analysis

QSPL panel proposal

Maurice H. ter Beek
ISTI–CNR, Pisa, Italy
maurice.terbeek@isti.cnr.it

Axel Legay
Université Catholique de Louvain, Belgium
axel.legay@uclouvain.be

## ABSTRACT

The explicit management of variability in the development cycle of software-intensive systems has led to a plethora of modeling and analysis techniques tailored to deal with behavioral validation of such configurable systems. Most of the work, however, focusses on qualitative (i.e. functional) requirements. Recently, there is growing interest in variability modeling and analysis techniques that do explicitly consider quantitative (i.e. non-functional) requirements, such as dependability, energy consumption, security, and cost.

Today's software is embedded in a variety of smart and critical systems that run in environments where events occur randomly and affect the system, and to which it needs to adapt. Therefore, quantitative modeling and analysis is currently a hot topic.

We propose a panel on Quantitative Variability Modeling and Analysis (QSPL) at VaMoS to discuss the latest quantitative modeling and analysis techniques and how these can be applied to variability modeling and analysis of software-intensive systems.

## CCS CONCEPTS

• **Software and its engineering → Formal methods**; **Extra-functional properties**; **Software product lines**;

## KEYWORDS

Variability, Quantitative modeling, Quantitative analysis, QSPL

**ACM Reference Format:**
Maurice H. ter Beek and Axel Legay. 2019. Quantitative Variability Modeling and Analysis: QSPL panel proposal. In *13th International Workshop on Variability modeling of Software-Intensive Systems, February 6–8, 2019, Leuven, Belgium,* Gilles Perrouin and Danny Weyns (Eds.). ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/

## 1 BACKGROUND AND MOTIVATION

Software Product Line Engineering (SPLE) concerns engineering, in a *cost*-effective and *time*-efficient manner, *configurable* software-intensive systems. The key issue is to manage *variability* among the products (or variants) of a Software Product Line (SPL), typically expressed in terms of *features.* Products share a set of core features, but differ with respect to product-specific features, meaning that a product can be seen as a set of features. A set of products (or family) is defined by a *feature model.*

The explicit management of variability in the software development cycle causes complexity in SPL *modeling and analysis.* An important example concerns *behavioral validation*, i.e., guaranteeing that each product of the family satisfies a series of behavioral requirements. Variants of this problem include computing a set of products that do not satisfy the requirements together with a justification. A common approach to assess such requirements consists of building the system and testing it. However, if the system should fail to meet the requirements, then costly interactions are needed to improve it. This problem is amplified in SPLE, where the number of products typically grows exponentially with the number of features. In fact, it is generally acknowledged that two major challenges with respect to behavioral validation of SPLs are i) to offer a compact and efficient way of modeling the behavior of families of products, and ii) to offer efficient analysis algorithms to exploit such models [36].

Throughout the last decade, we have witnessed numerous efforts on lifting well-known formal specification languages and formal verification techniques from (single system) software engineering to SPLE (or configurable systems), cf., e.g., [3–5, 7, 10, 14, 16–19, 23, 26, 37, 38] and the more exhaustive references in [6, 42]. However, the vast majority of work on variability modeling and analysis focusses on *qualitative* (i.e., functional) requirements. Only recently, we are witnessing a growing interest in variability modeling and analysis techniques that explicitly consider *quantitative* (i.e. non-functional) requirements, like dependability, which encompasses attributes like availability and reliability, but also energy consumption, security, and cost [8, 9, 12, 13, 20–22, 24, 25, 29, 30, 32, 35, 39, 40].

Today's software is embedded in a wide variety of *smart* and *critical* systems (e.g., aircraft, railways, automotive, and medical devices) that run in environments where events occur *randomly* and affect the system (e.g., think of failures) and to which it needs to *adapt.* For these reasons, quantitative modeling and analysis (e.g., through probabilistic systems and probabilistic or statistical model checking) is nowadays receiving a lot of attention. In the specific setting of configurable software-intensive systems, this requires modeling and analysis techniques able to cope with the complexity of systems stemming from behavior, variability, and randomness.

In [6], we coined the term *Quantitative Variability Modeling and Analysis* (QSPL) with the aim to cover the following kind of topics:

- Quantitative specification and verification techniques for systems with variability;
- Modeling and analysis of real-time, hybrid or probabilistic systems with variability;
- Analysis of safety, security or dependability properties of systems with variability;
- Modeling and analysis of dynamic, adaptive and (runtime) reconfigurable systems.

We believe that the time is ripe for a QSPL panel at VaMoS 2019 in Leuven, Belgium, so we have collected the following gender-balanced set of panelists from Belgium's four neighboring countries.


*VaMoS'19, February 6–8, 2019, Leuven, Belgium*

https://doi.org/10.1145/

## 2 CANDIDATE PANELISTS

Christel Baier (TU Dresden, Germany) is an expert in probabilistic model checking and quantitative analysis of stochastic systems and she has recently started applying this to SPLs [12, 13, 24, 25, 33].

Maxime Cordy (U Luxembourg) is an expert on behavioral modeling and analysis of SPLs, in particular based on featured transition systems, with affinity to quantitative approaches [14–16, 19–22, 40].

Uli Fahrenberg (École polytechnique, France) is an expert in quantitative extensions of transition systems, in particular weighted transition systems, and quantitative logics, including recent applications to behavioral SPL models [2, 11, 28–30, 39].

Mariëlle Stoelinga (U Twente, Netherlands) is an expert on quantitative modeling and analysis, in particular based on probabilistic, stochastic or timed automata models, and quantitative logics, but she has not yet been active in SPLE [1, 27, 31, 34, 41].

We have contacted these panelists and they have all expressed their intent to travel to VaMoS to participate in the panel in case our proposal gets accepted. Three out of four have never participated at VaMoS, so we believe this can lead to interesting discussions with our community on how to deal with quantitative notions in SPLE.

## REFERENCES


[1] A. Remke and M. Stoelinga (Eds.). 2014. *Stochastic Model Checking —Advanced Lectures of the International Autumn School on Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems (ROCKS'12)*. LNCS, Vol. 8453. Springer.

[2] S. Bauer, U. Fahrenberg, L. Juhl, K.G. Larsen, A. Legay, and C. Thrane. 2013. Weighted modal transition systems. *Form. Method. Sys. Design* 42 (2013), 193–220.

[3] M.H. ter Beek and E.P. de Vink. 2014. Using mCRL2 for the Analysis of Software Product Lines. In *Proceedings 2nd FME Workshop on Formal Methods in Software Engineering (FormaliSE@ICSE'14)*. ACM, 31–37.

[4] M.H. ter Beek, E.P. de Vink, and T.A.C. Willemse. 2017. Family-Based Model Checking with mCRL2. In *Proceedings 20th International Conference on Fundamental Approaches to Software Engineering (FASE'17) (LNCS)*, M. Huisman and J. Rubin (Eds.), Vol. 10202. Springer, 387–405.

[5] M.H. ter Beek, A. Fantechi, S. Gnesi, and F. Mazzanti. 2016. Modeling 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.

[6] M.H. ter Beek and A. Legay (Eds.). 2018. Special Section: Quantitative Variability Modelling and Analysis. *LNCS Trans. Found. Mastering Change* 2 (2018).

[7] M.H. ter Beek, A. Legay, A. Lluch Lafuente, and A. Vandin. 2016. Statistical Model Checking for Product Lines. In *Proceedings 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'16) (LNCS)*, T. Margaria and B. Steffen (Eds.), Vol. 9952. Springer, 114–133.

[8] M.H. ter Beek, A. Legay, A. Lluch Lafuente, and A. Vandin. 2018. A framework for quantitative modeling and analysis of highly (re)configurable systems. *IEEE Trans. Softw. Eng.* (2018).

[9] M.H. ter Beek, A. Legay, A. Lluch Lafuente, and A. Vandin. 2018. QFLan: A Tool for the Quantitative Analysis of Highly Reconfigurable Systems. In *Proceedings 22nd International Symposium on Formal Methods (FM'18) (LNCS)*, K. Havelund, J. Peleska, B. Roscoe, and E.P. de Vink (Eds.), Vol. 10951. Springer, 329–337.

[10] M.H. ter Beek, F. Mazzanti, and A. Sulova. 2012. VMC: A Tool for Product Variability Analysis. In *Proceedings 18th International Symposium on Formal Methods (FM '12) (LNCS)*, D. Giannakopoulou and D. Méry (Eds.), Vol. 7436. Springer, 450–454.

[11] P. Bouyer, U. Fahrenberg, K.G. Larsen, N. Markey, J. Ouaknine, and J. Worrell. 2018. Model checking real-time systems. In *Handbook of Model Checking*, E.M. Clarke, T.A. Henzinger, H. Veith, and R. Bloem (Eds.). Springer, Chapter 29, 1001–1046.

[12] P. Chrszon, C. Dubslaff, S. Klüppelholz, and C. Baier. 2016. Family-Based Modeling and Analysis for Probabilistic Systems: Featuring ProFeat. In *Proceedings 19th International Conference on Fundamental Approaches to Software Engineering (FASE'16) (LNCS)*, P. Stevens and A. Wąsowski (Eds.), Vol. 9633. Springer, 287–304.

[13] P. Chrszon, C. Dubslaff, S. Klüppelholz, and C. Baier. 2018. ProFeat: feature-oriented engineering for family-based probabilistic model checking. *Form. Asp. Comp.* 30, 1 (2018), 45–75.

[14] A. Classen, M. Cordy, P. Heymans, A. Legay, and P.-Y. Schobbens. 2012. Model checking software product lines with SNIP. *Int. J. Softw. Tools Technol. Transf.* 14, 5 (2012), 589–612.

[15] A. Classen, M. Cordy, P. Heymans, A. Legay, and P.-Y. Schobbens. 2014. Formal semantics, modular specification, and symbolic verification of product-line behaviour. *Sci. Comput. Program.* 80, B (2014), 416–439.

[16] A. Classen, M. Cordy, P.-Y. Schobbens, P. Heymans, A. Legay, and J.-F. 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.

[17] A. Classen, P. Heymans, P.-Y. Schobbens, and A. Legay. 2011. Symbolic model checking of software product lines. In *Proceedings 33rd International Conference on Software Engineering (ICSE'11)*. ACM, 321–330.

[18] A. Classen, P. Heymans, P.-Y. Schobbens, A. Legay, and J.-F. Raskin. 2010. Model Checking Lots of Systems: Efficient Verification of Temporal Properties in Software Product Lines. In *Proceedings 32nd International Conference on Software Engineering (ICSE'10)*. ACM, 335–344.

[19] M. Cordy, A. Classen, P. Heymans, P.-Y. Schobbens, and A. Legay. 2013. ProVeLines: a product line of verifiers for software product lines. In *Proceedings 17th International Software Product Line Conference (SPLC'13)*, Vol. 2. ACM, 141–146.

[20] M. Cordy and A. Legay. 2018. Verification and Abstraction of Real-Time Variability-Intensive Systems. *Trans. Found. Mastering Change* 2 (2018).

[21] M. Cordy, P.-Y. Schobbens, P. Heymans, and A. Legay. 2012. Behavioural Modelling and Verification of Real-Time Software Product Lines. In *Proceedings 16th International Software Product Line Conference (SPLC'12)*. ACM, 66–75.

[22] M. Cordy, P.-Y. Schobbens, P. Heymans, and A. Legay. 2013. Beyond Boolean Product-Line Model Checking: Dealing with Feature Attributes and Multi-features. In *Proceedings 35th International Conference on Software Engineering (ICSE'13)*. IEEE, 472–481.

[23] A.S. Dimovski, A.S. Al-Sibahi, C. Brabrand, and A. Wąsowski. 2015. Family-Based Model Checking using Off-the-Shelf Model Checkers. In *Proceedings 19th International Software Product Line Conference (SPLC'15)*. ACM, 397.

[24] C. Dubslaff, C. Baier, and S. Klüppelholz. 2015. Probabilistic Model Checking for Feature-Oriented Systems. In *Transactions on Aspect-Oriented Software Development XII*, S. Chiba, É. Tanter, E. Ernst, and R. Hirschfeld (Eds.). LNCS, Vol. 8989. Springer, 180–220.

[25] C. Dubslaff, S. Klüppelholz, and C. Baier. 2014. Probabilistic Model Checking for Energy Analysis in Software Product Lines. In *Proceedings 13th International Conference on Modularity (MODULARITY'14)*. ACM, 169–180.

[26] M. Erwig and E. Walkingshaw. 2011. The Choice Calculus: A Representation for Software Variation. *ACM Trans. Softw. Eng. Methodol.* 21, 1, Article 6 (2011).

[27] M. Faella, A. Legay, and M. Stoelinga. 2008. Model checking quantitative linear time logic. *Electron. Notes Theor. Comput. Sci.* 220, 3 (2008), 61–77.

[28] U. Fahrenberg and A. Legay. 2014. The quantitative linear-time–branching-time spectrum. *Theoret. Comput. Sci.* 538 (2014), 54–69.

[29] U. Fahrenberg and A. Legay. 2017. Featured Weighted Automata. In *Proceedings 5th IEEE/ACM FME International Workshop on Formal Methods in Software Engineering (FormaliSE@ICSE'17)*. IEEE, 51–57.

[30] U. Fahrenberg and A. Legay. 2018. Quantitative Properties of Featured Automata. *Trans. Found. Mastering Change* 2 (2018).

[31] M. Gerhold and M. Stoelinga. 2018. Model-based testing of probabilistic systems. *Form. Asp. Comp.* 30, 1 (2018), 77–106.

[32] C. Ghezzi and A. Molzam Sharifloo. 2013. Model-based verification of quantitative non-functional properties for software product lines. *Inform. Softw. Technol.* 55, 3 (2013), 508–524.

[33] L. Herrmann, M. Küttler, T. Stumpf, C. Baier, H. Härtig, and S. Klüppelholz. 2018. Configuration of Inter-Process Communication with Probabilistic Model Checking. *Trans. Found. Mastering Change* 2 (2018).

[34] T. Hune, J. Romijn, M. Stoelinga, and F. Vaandrager. 2002. Linear parametric model checking of timed automata. *J. Log. Algebr. Program.* 52-53 (2002), 183–220.

[35] M. Kowal, I. Schaefer, and M. Tribastone. 2014. Family-Based Performance Analysis of Variant-Rich Software Systems. In *Proceedings 17th International Conference on Fundamental Approaches to Software Engineering (FASE'14) (LNCS)*, S. Gnesi and A. Rensink (Eds.), Vol. 8411. Springer, 94–108.

[36] A. Legay and G. Perrouin. 2017. On Quantitative Requirements for Product Lines. In *Proceedings 11th International Workshop on Variability Modelling of Software-intensive Systems (VaMoS'17)*. ACM, 2–4.

[37] M. Lochau, S. Mennicke, H. Baller, and L. Ribbeck. 2014. DeltaCCS: A Core Calculus for Behavioral Change. In *Proceedings 6th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'14) (LNCS)*, T. Margaria and B. Steffen (Eds.), Vol. 8802. Springer, 320–335.

[38] R. Muschevici, J. Proença, and D. Clarke. 2016. Feature Nets: behavioural modelling of software product lines. *Softw. Sys. Model.* 15, 4 (2016), 1181–1206.

[39] R. Olaechea, U. Fahrenberg, J.M. Atlee, and A. Legay. 2016. Long-term Average Cost in Featured Transition Systems. In *Proceedings 20th International Systems and Software Product Line Conference (SPLC'16)*. ACM, 109–118.

[40] G.N. Rodrigues, V. Alves, V. Nunes, A. Lanna, M. Cordy, P.-Y. Schobbens, A. Molzam Sharifloo, and A. Legay. 2015. Modeling and Verification for Probabilistic Properties in Software Product Lines. In *Proceedings 16th International Symposium on High-Assurance Systems Engineering (HASE'15)*. IEEE, 173–180.

[41] M. Stoelinga. 2002. An introduction to probabilistic automata. *Bulletin EATCS* 78 (2002), 176–198.

[42] T. Thüm, S. Apel, C. Kästner, I. Schaefer, and G. Saake. 2014. A Classification and Survey of Analysis Strategies for Software Product Lines. *ACM Comput. Surv.* 47, 1, Article 6 (2014).