Quantitative Variability Modeling and Analysis

- 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

QSPL @ VaMoS
7 February, 2019
Leuven, Belgium

Panelists and moderators

- **Panelists**
  - Ina Schaefer (TU Braunschweig, DE)
  - Maxime Cordy (U Luxembourg, LU)
  - Pierre-Yves Schobbens (U Namur, BE)
  - Norbert Siegmund (BU Weimar, DE)

- **Moderators**
  - Maurice ter Beek (ISTI-CNR, Pisa, IT)
  - Axel Legay (UC Louvain, BE)

Behavioral validation of configurable systems

The explicit management of variability in software development causes complexity in SPL modeling and analysis, e.g. for behavioral validation:

- guarantee that each product of a family satisfies certain behavioral requirements
- compute a set of products that do not satisfy the requirements with a justification

These problems are amplified in SPLE because the number of products typically grows exponentially with the number of features, resulting in major challenges:

- offer compact and efficient way of modeling the behavior of families of products
- offer efficient analysis algorithms (e.g., model checking) to exploit such models

A decade of efforts on lifting well-known formal specification languages and verification techniques from single systems to configurable systems (SPLs):

- LTS ➔ FTS, MTS ➔ MTSv, CCS ➔ DeltaCCS, Petri nets ➔ Feature Nets
- SPIN (fPROMELA, fLTL), NuSMV (fSMV, fCTL), mCRL2 (feature μ-calculus)

Only recently explicitly consider quantitative (i.e., non-functional) requirements (e.g. dependability, energy consumption, cost, security)

- Featured timed/weighted/priced automata, configurable parametric timed automata, performance-annotated UML activity diagrams, variable UML sequence diagrams
- ProVeLines, PRISM / ProFeat, QFLan

Why QSPL?

“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.”

M.H. ter Beek & A. Legay (Eds.), Special Section: Quantitative Variability Modelling and Analysis. LNCS Trans. Found. Mastering Change 2 (2019)

Open questions (1/2)

- **How to relate quantities with variability?**
  - think of non-Boolean feature attributes (e.g., *real-time*, *cost*, and *rewards*) and quantitative constraints among them
  - feature attributes in these contexts likely represent more complex forms of variability, e.g., *data throughput*, *processor speed*, or *energy consumption*
- **How to incorporate quantities in (textual) languages for variability modeling?**
  - e.g. FAMILIAR, TVL, Clafer, some of which allow some *complex constraints*
  - how to remain conceptually close to *feature models* in the traditional sense?
- **How do quantities influence the (space-time) trade-off between product-based and family-based analyses and can (qualitative) abstractions be of help?**
  - in many cases, the overhead introduced by the *family-based* approach implies it is substantially slower than the *product-based* analysis
  - involves *more complex analysis* of the behavior that is shared among products
- **How about the robustness and sensitivity of quantitative variability modeling and analysis?**
  - assess how values of a property are affected by changes in *constituent values*
  - identify critical system parts with more significant impact by assigning a *degree of importance*, so as to optimise system parameters for more focused analyses

Open questions (2/2)

- **How to obtain quantities, which quantities do we really want to model?**
  - in case of several quantities, how to find an optimisation trade-off among them?
- **…**
  - …
- **…**
  - …

Possible solutions

- ...
  - ...
- ...
  - ...
- ...
  - ...