Modelling the Evolution of Software Product Lines

Dr. Andreas Pleuss
Supervised by Dr. Goetz Botterweck
Research Strand Software Product Lines (SPL)

Model-driven SPL Engineering
- Leverage concepts from Model-driven Engineering:
  - Models
  - (Semi-) automated model transformations
  - Visualizations and interactive tools
- Missing so far: Support for Evolution of SPLs

Challenges
- Example automotive domain: management decisions arise frequently
  - “In 2011 we will add a navigation system”
  - “In 2012 we will provide a DVD player as additional media option”
  - “From 2013 CD players will be no longer supported”
- Goal here: Support for proactive long-term planning on feature level

Related Work
- Delta Models (Change Sets):
  - Model fragments specify delta to previous model (+ context)
  - Support add, remove, modify
- Change Operators:
  - Specify changes in terms of (complex) operations
  - Specific for a modelling concept (e.g., metamodels)
- Both concepts relative to previous models or to baseline
  - Limited use for incremental, order-independent planning

Modelling Evolution of Feature Models
- Abstract to Model Fragments which change/remain stable
- Change Operators for local property changes within fragments
- Feature Model concepts for relationships/constraints between them (called “EvolutionFeature Model”, EvoFM)
- A version of the evolving feature model corresponds to a configuration of the EvoFM
- Evolution Plan shows all configurations over time
- Rationale associated with one or more decisions in the Evolution Plan

Application & Current Results
Analysis of Evolution
- Model transformations automatically generate an EvoFM from given set of feature models
- Incremental approach → allows updating the EvoFM at any time
- Application to open-source project Eclipse (interpreted as SPL)
  - Example: Eclipse 3.4 (“Ganymede”), 1957 Features, 2017 Cross-tree constraints

Planning of Evolution
- Chosen approach allows incremental, order-independent long-term planning of evolution decisions in Evolution Plan
- Formal semantics of feature models allows constraint checking
- Model transformation automatically generates feature models from given EvoFM configurations
- Application to SPL in automotive domain (with semi-automated propagation of evolution changes to Matlab/Simulink models)