Comparison of Service and Software Product Family Modeling

Mikko Raatikainen, Varvana Myllärniemi, Tomi Männistö

Helsinki University of Technology
Finland
Agenda

- Problem
- Modeling in software product family
- Modeling in Services
- Comparison
- Conclusions
- Question specific to this paper
Problem

- Similarities in software product families and service oriented computing
  - Both aim at efficiently developing application from existing pieces of software
  - Both rely on models
- But also differences
  - Typically services are dynamic computational elements
  - Typically software product families deal with static elements

⇒ we discuss the similarities and differences in service oriented computing modeling and software product family modeling
Software Product Family Modeling

- Domain model including variability and product model to express the product of a software product family
- Several approaches to model variability
  - Specific approaches for variability modeling
  - Extension to existing approaches such as UML
  - Variability models to augment existing models
Service Oriented Computing Modeling

- Dominated by web service initiatives
- Basic concepts relatively mature such as WSDL
- Advanced concepts not as established
- Typically driven by different standards, such as WSDL and BPEL
Comparison

- No domain or variability modeling in services
- Service typically composition whereas software product family decompositional
  - No technical reason to do the opposite
- Both focus on architectural level concepts
  - Services typically dynamic elements whereas components static
- Composition, interfaces, and connections in both approaches

© Mikko Raatikainen, 2007
Comparison

- Notations in software product families typically graphical whereas in service XML-based
- Service modeling driven by standards, whereas in software product families plethora of approaches
- Services focus on dynamic aspect and stakeholders relevant to that whereas software product family adhere to different architectural viewpoints
Conclusions

- Feasibility of variability modeling in services
- Behavior modeling and analysis of services in software product families
- The actually needed concepts for modeling of services and software product families
  - Different viewpoints in services
- Unify variability modeling concepts
- Apply and reuse the modeling methods from other approaches
Question specific to this presentation

Are there detailed examples or a comparison of models, e.g. feature models vs. SDL/BPEL/BPMN

- To best of our knowledge modeling in the approaches has not been compared before
- We have tried to apply our software product family modeling tools (KumbangTools) to service composition
  - To some extended feasible
  - Not suitable for complex behavior
Thank you!

Questions?

mikko.raatikainen@tkk.fi