Research Challenges in Domain-specific Modeling

- Quality of DSMLs
- Composition of DSMLs
- Usability of DSML-related techniques
- Other challenges

Brainstormed by Faruk Caglar, Filipe Correira, Gökhan Kahraman, Jeff Gray, Philip Langer, and Timo Wegeler at DSM'13 in Indianapolis

Quality of DSMLs

- What are adequate quality characteristics of DSMLs?
  - Can we adopt existing quality characteristics for DSMLs?
    - E.g., adopting quality characteristics of software products (in general) or of programming languages
- How to evaluate the quality of DSMLs quantitatively and qualitatively?
  - What are adequate evaluation methods?
  - Parts of DSMLs should be evaluated in separation, but is this possible for DSMLs (e.g., evaluating the concrete syntax separately from the editor)?
- Quality is a matter of perspective
  - What are the stakeholders of DSMLs (domain expert, manager, developer, …)?
  - Which characteristics are the most important ones for which stakeholder?
- Gökhan reported on his recent work in this area
  - He proposes an assessment method, which adapts the quality characteristics of software products to DSMLs and which takes into account “evaluator profiles” (different perspectives of a DSML’s stakeholder)

Composition of DSMLs

- Composition of DSMLs is crucial for increasing reuse in DSM
- Composing DSMLs have to address their
  - Abstract syntax
  - Concrete syntax
  - Semantics
- How can concrete syntaxes of multiple DSMLs be composed?
- How can the different semantics of multiple DSMLs be composed?

- Jeff mentioned existing work of the GEMOC project

Usability of DSMLs and DSML-related Techniques

- One important goal of DSMLs is to enable domain experts to specify their systems
- However, are domain experts without a background in programming really able to fully benefit from DSMLs and related techniques (e.g., code generation, model transformation)?
- Often, generated code or interpreters raise (runtime) errors while the model is valid
  - Such errors can hardly be traced back to the original error in the model by non-programmers
- How can the development of DSMLs and DSML-related artifacts be simplified?
  - Recent work shows that demonstration-based approaches seem to be promising
    - E.g., model transformation by demonstration
    - Metamodel specification by demonstration
  - Still lot of work to do!

Other Challenges

- Explicit specification of the semantics of DSMLs
  - No commonly accepted and widely used explicit semantics specification technique
- Co-evolution of artifacts that depend on evolving models
  - Approaches exist for metamodel/model co-evolution
  - A few approaches are available to address other co-evolution problems
  - However, these approaches are very specific to the respective co-evolution problem
  - A general approach that is usable for every co-evolution problem is strongly needed
- …
  - (we are sure there are many other challenges that were not listed above ☺)