DSM Workshop 2016
Discussion on Workbenches and Tools

Robert Hendriksen, Gabriel Konat, Riccardo Solmi, Andreas Wortmann, Enrico Persiani, Tanja Mayerhofer
Expertise and Interests in the Discussion Group

- Workbenches (each participant having own tooling):
  - EMF
  - GEMOC Studio
  - WHOLE-Platform
  - Spoofax
  - MontiCore
  - Supermodels (built on C#, graphical language workbench, proprietary)

- Tools:
  - Specification of abstract syntax of languages (grammars, schema, metamodel inference)
  - Specification of static semantics
  - Specification of operational semantics (interpretation)
  - Specification of translational semantics (compilation)
  - Specification of grammars
  - Model-to-model transformations
  - Code generators
  - Model simulators
  - Model debuggers
Future Challenges for Language Workbenches

• Composition of languages
  • For composing languages, we have to compose the different components of a modeling language (abstract syntax, concrete syntax, static semantics)
  • We also have to compose the complete tooling (code generators, editors, parsers, …)
  • How should composition of languages work? There are different techniques: subtyping, interfaces, references (might work differently for different language artifacts)
  • Language composition vs. integration vs. reuse

• Generalize semantics specifications to automatically generate V&V tools (debuggers, formal verification tools, testing tools, …), also code generators (they would be correct by construction)
  • Currently, different semantics have to be defined to generate different kinds of tools; or tools have to be developed manually