

Domain Specific Metamodelling Languages

Domain-Specific Metamodelling Languages for Software Language Engineering

Steffen Zschaler, Dimitrios S. Kolovos, Nikolaos Drivalos, Richard F. Paige, Awais Rashid



- Sometimes, languages developed multiple times in a new context
 - => Recurring patterns and variations

- Concept of domain specific meta modelling language

Some (trivial) definitions

- DSL: custom- and purpose-built languages that target a specific domain
 - ↔ GPL
 - Implemented on top of metamodeling infrastructure (ex. EMF/Ecore)
- Families of DSLs: set of languages targeting a concrete instance of a common problem domain

Presentation outline

- Challenges for developing families of DSLs
- Domain specific metamodelling languages
- Examples
 - TML
 - VML*

Challenges for Developing Families of DSLs

- DSL
 - Close to problem domain
 - Requires a lot of expertise and effort

- Sometimes: similar DSL concepts in different domains
 - ex. traceability between model artefacts
 - notion of TraceLink
 - ex. variability in product-line development

- Alternative: develop generic language instead

Challenges

1. Systematic reuse of DSL concepts and infrastructure elements
2. Systematic support for specifying variability between the DSLs
 - Addition/removal of concepts
 - Integration with different additional languages



Domain Specific Metamodelling Languages (DSM2L)

More definitions

1. **Family of languages:** common core + set of types of variation
2. **Domain-Specific Metamodelling Language (DSM2L):** used to define syntax, semantics and tooling aspects of languages to a specific family of languages
 - more concise and efficient than GPL
 - not capable of capturing every metamodel

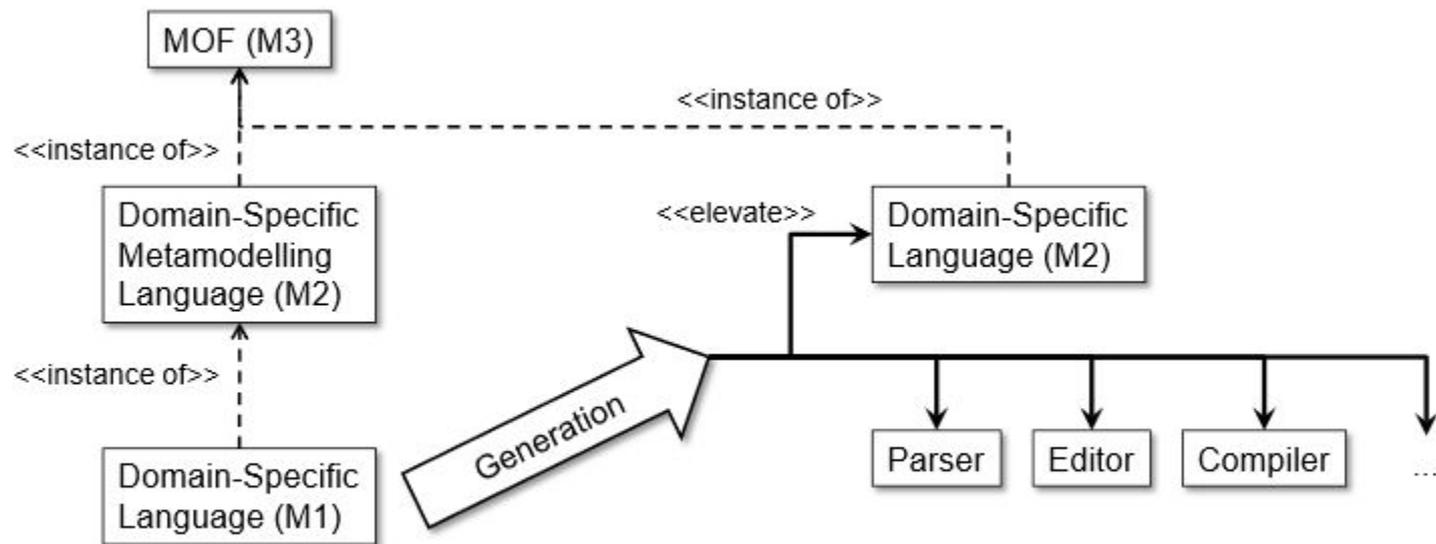
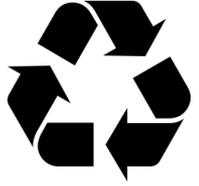


Fig. 1. Generating language support infrastructures from a DSM2L

DSM2L: challenges



DSM2L:

1. Supports systematic reuse of DSL constructs
2. Systematic support for specifying variability

Example: TML

Traceability

- Embed in model ↔ distinct traceability model
- Generic nature
 - set of traceability concepts
 - represented in metamodel as metaclasses
- Case specific nature
 - more time and effort required
 - contain similar constructs

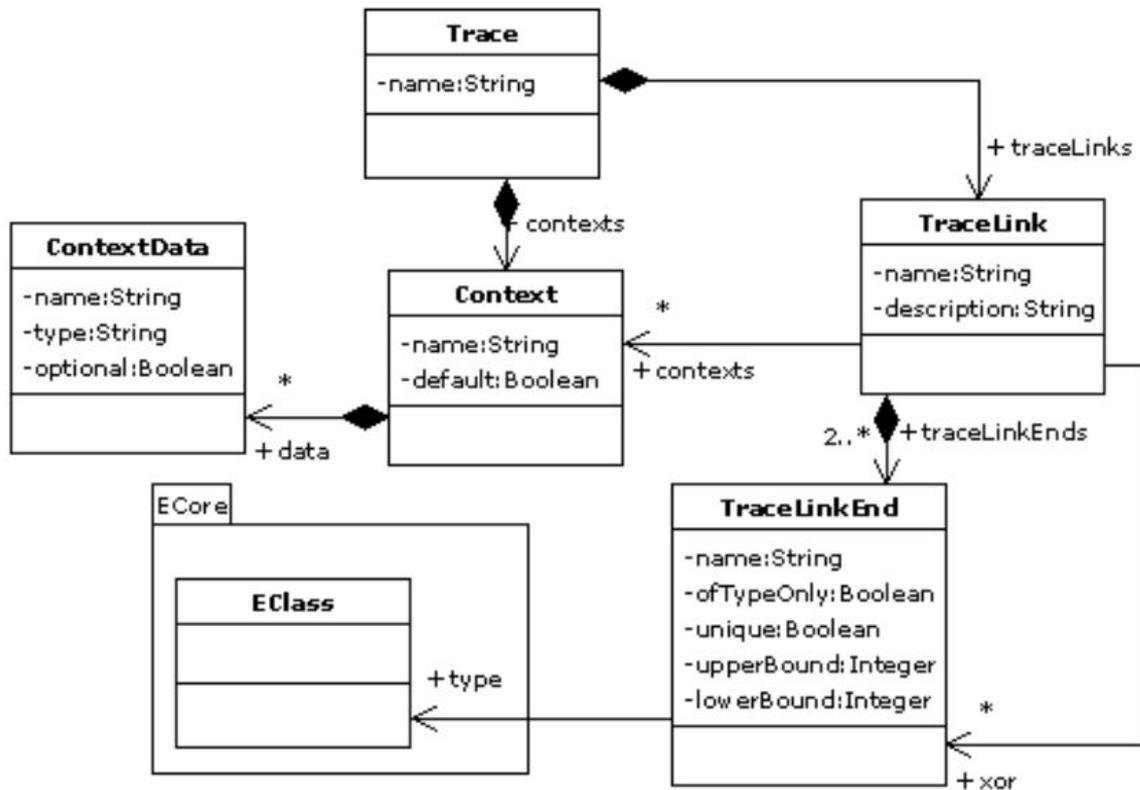
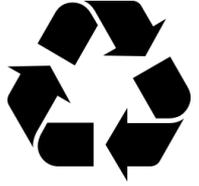


Fig. 2. Core of the TML (from [18])



TML: challenges

1. Systematic reuse:

by promoting recurring patterns to first class metamodels

2. Systematic support for variability:

different types of TraceLinks and TraceLinkEnds

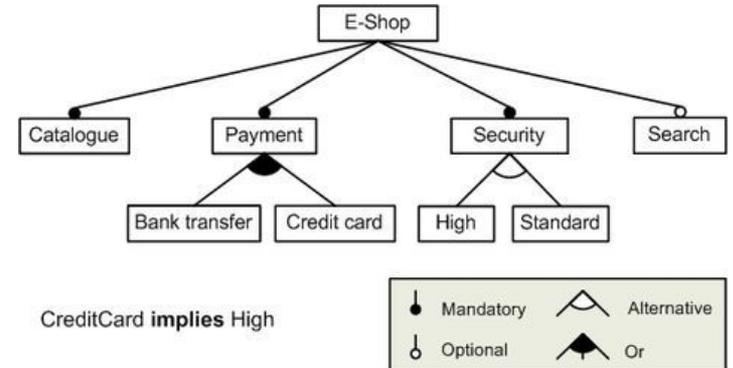
Some constraints should be specified manually

Example: VML*

VML*

- Variability Modelling Language
- Used in SPLs
- Feature model
- Mapping problem
 - each feature => identify and specify solution space models

- VML*: DSM2L for VML languages
 - common core metamodel



VML* core

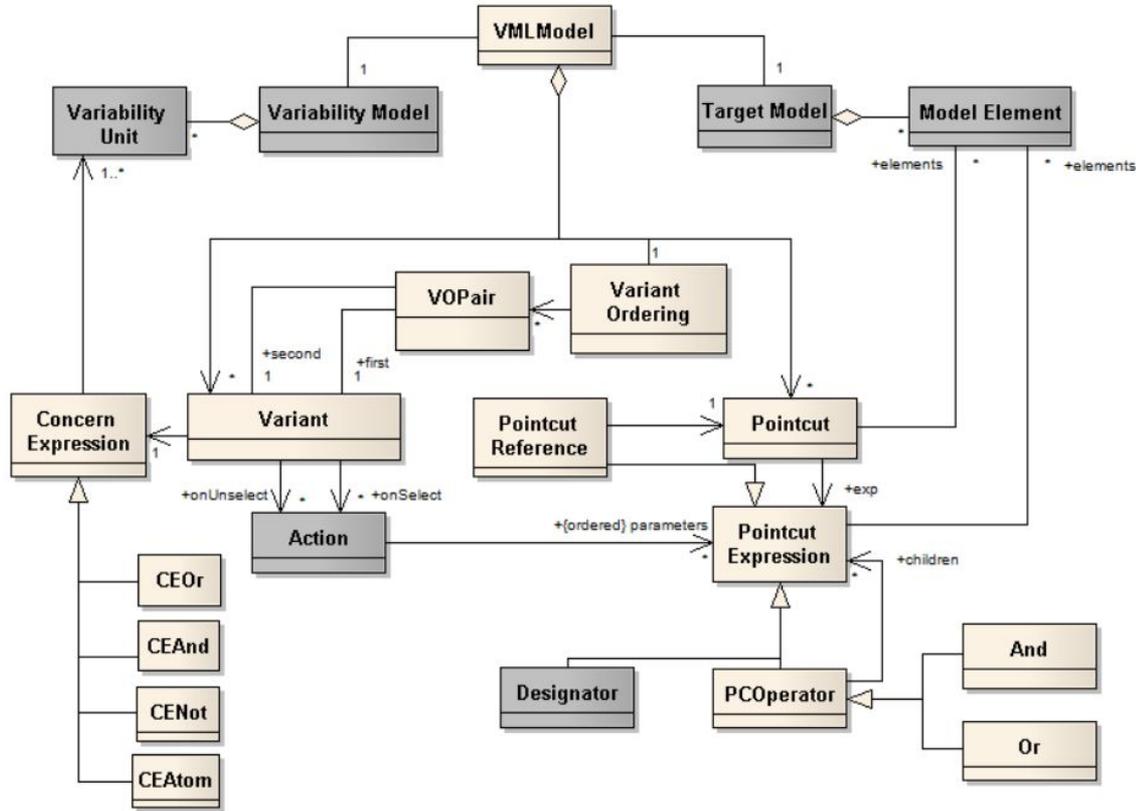
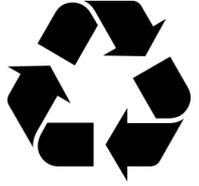


Fig. 3. Common core metamodel for VML languages (from [48])

VML*: challenges



1. Systematic reuse:
 - support to reuse all shared concepts
2. Systematic support for variability:
 - specify differences in one place

Conclusion

- DSM2L: try to solve issue of re-implementing recurring patterns in DSLs
- Most of the time: unable to specify all constraints for each model