The bicycle challenge in DMLA, where validation means correct modeling

Gergely Mezel, Dániel Urbán, Sándor Bäch (DAM, Budapest University of Technology and Economics, Budapest, Hungary)
Zoltán Theisz (evopro Systems Engineering Ltd., Hauszmann Alajos str. 2, Budapest, Hungary)

Step-by-step evolution of a concept

- Something
- Bicycle
- Mountain bike
- Model XZ362
- My "Black Thunder"

Core

- "The hardware"
- Data structure
  - ASM State: snapshot of the model
  - Model: nodes, edges, attributes (entities)
  - Labelled directed graph
  - Entities have labels - entities are 4-uplets
- Unique ID
- Meta: reference to meta element
- Value: concrete values
- Attributes: list of connected entities

Management

- Derived functions: obtaining information
- Shared functions: snapshot evolution

Bootstrap

- "The operating system"
- Set of entities - enabler of multi-level meta-modeling
- Flexible and scalable
- Multi-purpose meta hierarchy

Bootstrap2

- Constraints: constraints on model elements
- Domain: auralization

Bootstrap3

- Constraints: constraints on model elements
- Domain: auralization

DMLAScript

- "The programming language"
- Hides holes, more-user-friendly
- Entities & relations (Consider illusion for related tuples)
- Operations, User-friendly operation language
- Rule-based
- Full-modeling by 4-uplets
- Described by Abstract Syntax Tree (AST) built from entities (statements, expressions. if for variable, ...)
- OperationDefinition with signature constraints (return type, parameters, ...)
- OperationCall

Validation

- AI: entity support validation based on formula
- Alpha. meta against instance (e.g. type conformity)
- Beta: meta against entities (in-context check) (e.g. cardinality)
- Gamma: all instances of a certain entity (e.g. unique serial number)
- Validation mechanism:
  1. Loop over all entities
  2. Obtain meta by the meta ASM function
  3. Fetch all alpha = hierarchy chain up to base
  4. Combine alphas (meta, instance) by AND
  5. Combine entity variations by AND

Fluid multi-level metamodeling

- Fluid levels:
  - Stepwise refinement of domain concepts is decoupled
  - Meta - Instance is interpreted between two entities
  - Entities can refer to other entities on any levels
  - Consequence of real multi-level modeling
  - Entities are instantiated individually
- Partial instantiation
- Stepwise refinement: attributes - > also independent
- Some of the features remain untouched, others are concretized
- Entity instantiation = copy or concretize slots

Patterns

- Concretization: Concrete object
- Entity sketch only
- Notations

P0: Prohibition of features or their relation can be simply left out since validation will fail if disallowed entities occur in the model.
P1: Gradual type constrainning is supported by restricting constraints on slots.
P2: Create new slots by dividing general purpose slots, when new features are needed. Keep the original slot if adding new features may be required later or omit it otherwise.
P3: Mandatory slots are modeled by cardinality 1.1. They must be kept all along the whole instantiation chain. Optional slots are modeled by cardinality 0.1. They can be omitted on any level. Optional-mandatory slots are modeled by cardinality 0.1 and the MustIf*Oncce constraints can be omitted on any level supposing their value has been set earlier.
P4: Inheritance between entities is instantiated by instantiation

P01: Custom validation is driven by flags. If the flag is presented, the validation is turned off, if the flag is omitted, the validation is switched on.

DMLA Webpage:
https://www.daml.com/evopro/Content/VM/OMLA
The Bicycle challenge:
http://www.wald-ver.de/dresden/essen.de/MULT1018

P5: Enum and flag-like requirements are modeled as an enum type with its slot-less instances.
P6: Slots can represent signature-driven operation definitions at all concreteization levels.
P7: Alpha formula completes an entity's type semantics beyond meta-hierarchy.
P8: Global validation requirements are satisfied by gamma formulas.
P9: Soft validation, i.e. filtering features are supported by operations attached to the entities.