Model-Based Specification in VDM

http://d3s.mff.cuni.cz

Pavel Parízek

Department of Distributed and Dependable Systems

FACULTY OF MATHEMATICS AND PHYSICS
Charles University
Vienna Development Method (VDM)

- Formal specification languages
  - VDM-SL
  - VDM++

- Combination: model-based + algebraic
  - Abstract modeling (data + contracts)
  - Executable subset (prototyping implementation)

- Tools
  - validation, analysis, testing
  - code generation (Java, C++)
VDM-SL

• Syntax
  - ASCII text, graphical

• Features
  - Basic types: numeric, character, token, quote
  - Collections: set, sequence, map
  - Type constructors: union, cartesian product, record (composite)
  - Functions (pure, no side effects)
  - Operations (modify global state)
Example

- Management system for public transport

- Key concepts
  - Modules (import, export)
  - Implicit definition of functions/operations
    - Contracts (precondition, postcondition)
  - Explicit definition of functions/operations
    - Prototype implementation (algorithm)
  - Control-flow structures
    - imperative, functional
Proving correctness

**Implicit definition**

\[
f(p: T_p) r: T_r \\
pre \quad \text{pre-} f(p) \\
post \quad \text{post-} f(p, r)
\]

**Explicit definition**

\[
f: T_p \rightarrow T_r \\
f(p) = \ldots
\]

**Proof obligation**

\[
\text{forall } p: T_p \cdot \text{pre-} f(p) \Rightarrow f(p): T_r \text{ and post-} f(p, f(p))
\]
Refinement – another perspective

- Abstract data representation AR
- New concrete data representation CR
- Abstraction function $\alpha : \text{CR} \rightarrow \text{AR}$

Proof obligations

- $\forall a: \text{AR} \cdot \exists c: \text{CR} \land a = \alpha(c)$
- $\forall c: \text{CR} \cdot \text{pre-OpA}(\alpha(c)) \Rightarrow \text{pre-OpC}(c)$
- $\forall c^\sim, c: \text{CR} \cdot \text{pre-OpA}(\alpha(c^\sim)) \land \text{post-OpC}(c^\sim, c) \Rightarrow \text{post-OpA}(\alpha(c^\sim), \alpha(c))$
Case studies

- International conference on Rigorous State Based Methods: ABZ
  - [https://abz2021.uni-ulm.de/](https://abz2021.uni-ulm.de/)
  - [https://www.southampton.ac.uk/abz2018/participants/programme.page](https://www.southampton.ac.uk/abz2018/participants/programme.page)
Tools

- **VDMTools**
  - [http://fmvdm.org/vdmtools/](http://fmvdm.org/vdmtools/)
  - Checks syntax, types, integrity
  - Interpreter (debugger)
  - Code generation (Java, C++)

- **Overture**
  - [http://overturetool.org/](http://overturetool.org/)

- https://dl.acm.org/citation.cfm?id=94062