Model-Based Specification in VDM

Pavel Parízek

http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems

CHARLES UNIVERSITY IN PRAGUE
faculty of mathematics and physics
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)
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 \ pre-\ f(p)  
post \ post-\ f(p,r)  

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

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

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

Proof obligations

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