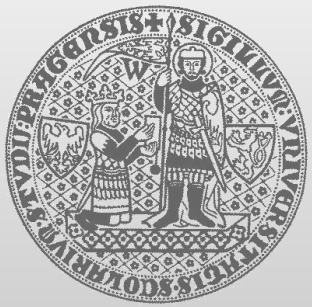
Model-Oriented Specifications & Language Z

http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems

Pavel Parízek



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

Model-oriented specifications

- Model: internal state & operations
- Implementation
  - Iterative refinement of specifications
  - “coding from scratch”
    - Tests generated from the model
- Target domain: complex state
  - databases, file systems
- Popular languages: Z, B, VDM
  - set theory + first-order logic + programming languages

Language Z

- Based on: sets, relations, predicates, and formulas in the first-order logic
- Specification
  - Informal description (plain text)
  - Formal chunks (graphical notation)
- Examples: bank account, file system

Schemas

- Structure: name, declarations, constraints

- System state schema
- Schema of operation

- Notation for variable names
  - input arguments (name?) versus output (name!)
  - pre-state (plain: a) versus post-state (primed: a’)

Schema calculus and composition

- Combining schemas
  - logic connectives (and, or, not)
- Including schemas
- Benefits: modularity & reuse

Process of creating specifications in Z

- Main steps
  - Informal description (plain text)
  - Schemas for the system state
    - Sets of necessary information about subject world
  - Schemas for operations
- Recommended approach
  - First standard control flow and valid inputs
  - Then incorrect inputs and handling errors
- Proving some claims
  - Using axioms and inference rules

Other features of Z

- Generic schemas
- Sequences

Tool support

- http://formalmethods.wikia.com/wiki/Z_notation#Tool_support
- http://czt.sourceforge.net/

Literature

- J.M. Spivey. The Z Notation: A Reference Manual. Oxford
  - http://spivey.oriel.ox.ac.uk/wiki2/files/zrm/zrm.pdf

- J. Bowen. Formal Specification and Documentation using Z: A Case Study Approach.