Summary

http://d3s.mff.cuni.cz

Pavel Parízek
How to create formal specifications

- Algebraic languages (Maude)
  - prototyping and testing
- Model-based languages: high-level design combined with
  - theorem proving (Z)
  - classic testing (VDM)
  - SAT analysis (Alloy)
- Concurrency and temporal behavior
  - Petri nets, TLA+
- Industry standards: UML and OCL
Experience

• Creating high-level formal specifications, models, and tools enables:
  ▪ thinking about the system at the domain level
    • ignore low-level implementation details (in Java or C)
  ▪ validation and early detection of design errors
  ▪ better understanding of the complex behaviors

• Usage of formal models allows easier
  ▪ validation of changes (what-if analysis)
    • especially when compared to implementation (code)