Future Proof Dynamic Semantics for Overture (position statement)

Third Overture Workshop
Newcastle University
28 November 2006

Marcel Verhoef – CHESS
(with Jozef Hooman – ESI)

Theory versus Tool (1)

- VDM-SL has a formal dynamic semantics
- Was this semantics ever proven to be correct?
- $\text{VDMTools}_{\text{SL}}$ has an operational semantics
- Was this semantics ever proven to be correct?
- Does the operational semantics “implement” the dynamic semantics correctly?

- Informal conformance was demonstrated by means of testing

Theory versus Tool (2)

- VDM++ has *no* formal dynamic semantics
- It is argued that it is a superset of VDM-SL
- But this was never (formally) demonstrated
- $\text{VDMTools}_{\text{PP}}$ has an operational semantics
- Was this semantics ever proven correct?

- Again, validation by means of testing

Theory versus Tool (3)

- VDMTools has been notably successful, why?
- Validation approach (testing) was sufficient to guarantee conformance to the semantics

- And then came VICE
- Extends VDM++ with a notion of time
- Two-phase execution model (state, time step)
- Required another, much finer grain, operational semantics (a virtual machine)

Theory versus Tool (4)

- Concessions where made to e.g. scheduling strategies to keep tool performance acceptable
- This *may* cause phenomena in the observed behavior of the model that are *not* part of the model
- Again, testing was used to validate the operational semantics
- Lower abstraction level of the operational semantics makes this task *much harder*

Theory versus Tool (5)

- And then there was VICE++
- Adds multi-processing and asynchrony to VDM
- Substantial changes to the semantics required
- An abstract formal semantics given in [VLH06]
- Some questions arise:
  - Is it possible to validate the changes by testing?
  - Does the operational semantics implement the abstract formal semantics?
  - Tool optimization versus model accuracy?

conjecture

The current VICE++ operational semantics is not future proof.

Research Questions

- How to integrate probabilistic models?
  - e.g. `duration (x, y, z) ( .... )`
- Analysis instead of simulation?
  - Proof (our obvious short term objective)
  - Concurrency analysis with SMV, SPIN,...
  - Timing analysis with UPPAAL, KRONOS,...
  - Verification a la SDV (program abstraction, SAT)