­­­­­­­

ACM TOSEM

Run to completion semantics with Refinement using Event-B

Background

SCXML 2 iUML-B past

iUML-B 2 Event-B

R2C Semantics (e.g. SCXML)

Event-B Semantics

Refinement

Representing the operational semantics of SCXML in Event-B

Basis – Engine

Allowing for refinements – guard strengthening

Limitations: we do not support the full ordering of actions

Early attempts –

1. Simple next step – based on negated guards – negation is weakened
2. Engine – improves the R2C semantics but still suffers from the negated guards problem (refinement of the user model kinda works – simulations at the user level).

The solution:

1. Transition combinations approach – works because there is always one event to completion. Invariants as well as simulations.
   1. Explosion – mitigated by mutual exclusions… same trigger – different state-chart regions.

What can we do with it?

Why is RTC needed?

Refinement in RTC

An example

Choose example to illustrate need for RTC – i.e. something that wouldn’t work in Event-B

Discussion

Compare the behaviour of a visually similar statechart in iUML-B and SCXML

Based on their Event-B translations

Using the theorem prover to test whether they are equivalent (they are not)

Using the model checker to compare traces

Using LTL to show that certain temporal properties are not achieved by the iUML-B version.

Conclusion

Note: Things to highlight with the choice of example

1. Look at a system that is better model with SCXML run to completion semantics than iUML-B semantics
2. Look at how you can check for violations of refinement in a SCXML model construction
3. Look at the sort of invariant properties you can verify about a SCXML model