

Refinement of Reactive Systems*

Subtitle[†]

First1 Last1[‡]

Position1

Department1

Institution1

Street1 Address1

City1, State1 Post-Code1, Country1

first1.last1@inst1.edu

First2 Last2[§]

Position2a

Department2a

Institution2a

Street2a Address2a

City2a, State2a Post-Code2a, Country2a

first2.last2@inst2a.com

Position2b

Department2b

Institution2b

Street3b Address2b

City2b, State2b Post-Code2b, Country2b

first2.last2@inst2b.org

Abstract

Text of abstract

Keywords run-to-completion, refinement

1 Introduction

Possible outline:

- Motivation
- Background and Previous Work
- Description of Sample Application
- Statechart Refinements
- Verification of Safety Properties
- Results
- Conclusion

Ideas on content We prove properties in early refinements - because easier to see - (so far fully automatic - try proving battery inv in later refinement) although we put everything in the same model we envisage gradual construction we could also build a Sirius tool that can show views at each refinement.

We simulate the translation with the scenario checker. It automatically runs to completion (events are annotated as internal and prioritised) This shows the main intended R2C behaviour for a refinement level. Other behaviours are possible depending on future refinement - e.g. a completion can occur without taking enabled transitions if in future we

*with title note

[†]with subtitle note

[‡]with author1 note

[§]with author2 note

might strengthen a guard. these can be explored by single event stepping without the scenario checker big step.

proof forces us to finalise transitions.. .. i.e. to remove non-determinism that allows for some future refinement. (in usual event-b it is not built in to the 'engine' and is only checked when the next refinement is provided.. .. we only have this finalisation because we needed to model the run to completion queueing mechanisms, hence new transitions modify old variables (queues)) we have to limit what can be done in later refinements as some things could violate what is proved. e.g. a response transition must come next (to make the invariant before completion) so we cannot strengthen its guards any more. e.g. similarly a trigger raised as a response must be used in the way that preserves the invariant, so future transitions may not consume it.

Also section on how we arrived at the new translation. adding the invariant about only one dequeued trigger at a time - otherwise proof did not discharge about the future triggered transitions (because consuming another dequeued trigger would discard the one we are interested in) the drone model made us rethink the basis.. we had to dequeue triggers.. (why?)

2 Background and Previous Work

3 Run To Completion

Describe Run to completion semantic leading to our Event-B implementation in Basis.

4 Translation to Event-B

Describe how we enhance SCXML with 'umlb' attributes (inc. refinement levels) and translate into UML-B state-machines.

5 Description of Sample Application

6 Statechart Refinement

7 Verification of Safety Properties

In a statechart model we naturally wish to verify properties P that are expected to hold true in a particular state S . Hence, all of the safety properties that we consider are of the form: $S = \text{TRUE} \Rightarrow P$, where the antecedent is implicit from the containment of P within S .

There are two kinds of properties that we might want to verify in a statechart; 1) properties concerning the values of auxiliary data which is being maintained by the system and 2) constraints on the state of a parallel statechart region.

SCXML models represent components that react to received triggers and cannot be perfectly synchronised with changes to the monitored properties. Hence, if naively expressed, P may be temporarily violated until the system reacts by leaving the state S in which the property is expected to hold. To cater for this we express P in a modified form P' that allows time for the reaction to take place.

There are two forms of reaction that can be used to exit S ; a) an untriggered transition, or b) a transition that is triggered by an internally raised trigger. For a), the modified property P' is *untriggered transitions are not complete* or P , and for b) P' is *trigger t is in the internal queue or dequeued* or P (where t is the particular trigger needed to react to the violation of P).

For properties about the value of auxiliary data, untriggered transitions appear to be more suited because, in this case, there is unlikely to be a natural place to raise an internal trigger when the appropriate conditions arise. For properties about the state of a parallel region either reaction could be used depending on whether the system detects the violation in the state that contains P or the state that P refers to.

In this section we illustrate a typical example of the type of properties that we imagine could be verified in a reactive SCXML system. All of the proof obligations are automatically discharged for our sample application. Since our models are strictly structured and proof obligations will always have this common form, we expect that proofs will always discharge automatically. We model the safety property features at an early level of refinement where the models are relatively simple, so that the modeller has more chance of expressing valid verification conditions. Detail is then added in later refinements which is proven (automatically) to preserve the proven safety properties. In our example, some auxiliary data is monitored by one statechart and another statechart utilises this by referring to the state of the monitor. Hence the reaction consists of an untriggered transition in the monitoring statechart that sends an internal trigger to the other statechart when it leaves the desired monitoring state.

Now describe the battery charge verification in UML-B

8 Results

Text of paper ...

9 Conclusion

Text of paper ...

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. nnnnnnn and Grant No. mmmmmmm. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

A Appendix

Text of appendix ...