Translating SCXML Statecharts to iUML-B State-machines

Karla Morris[1] and Colin Snook[2]

[1] Sandia National Laboratories, Livermore, California, U.S.A.
knmorri@sandia.gov
[2] University of Southampton, Southampton, United Kingdom
cfs@ecs.soton.ac.uk

To facilitate automatic proof, the Event-B notation is restricted to a simple guarded action behaviour. While iUML-B goes some way to provide an intuitive state-transition representation of Event-B models, its notation follows the semantics of Event-B. Engineers that are used to the richer semantics of Harel style statecharts may find these restrictions difficult to accept. Conversely, Event-B has features such as refinement and invariant properties that are not considered in most statechart notations. It may be cumbersome for engineers to re-model existing systems into iUML-B for verification.

In order to explore the feasibility of this model transformation, we have developed a translation from a statechart representation, *State Chart XML: State Machine Notation for Control Abstraction* (SCXML) [3], into iUML-B. For this initial work we do not support features of SCXML associated with more problematic areas of the semantic mismatch, such as 'run to completion' transition sequencing. Nevertheless, the translation provides an interesting first step towards interchange between the two notations.

SCXML is an XML notation for Harel (hence UML) style statecharts extended with a general purpose action language. The concrete syntax for SCXML is based on XML and includes a data modelling facility and an action language. An example of SCXML syntax is shown in figure 1.

To facilitate Event-B formal verification, extensions to the SCXML modelling notation are necessary so that additional modelling features required by Event-B can be integrated with the SCXML model. The SCXML schema allows extension elements and attributes belonging to a different namespace to be added. The SCXML tooling provides fallback mechanisms so that these extensions are supported without the need for syntactic definition. We define a new namespace, *iumlb* and add two new elements, *iumlb:invariant* and *iumlb:guard* as well as a number of new attributes. Invariants are not supported in SCXML and SCXML transitions only have a single *cond* attribute whereas we may need to introduce conjuncts of a transition condition at various refinement steps. The concept of refinement does not exist in SCXML. We introduce a new integer valued attribute, *iumlb:refinement*, which may be attached to any element of either namespace in order to specify the refinement level of that element.

The iUML-B tools are based on the *Eclipse Modelling Framework* (EMF) [2]. It is beneficial to load the SCXML model into EMF so that our existing model transformation technology can be used to implement the SCXML to iUML-B

```
<iumlb:invariant iumlb:refinement="1" predicate="TRUE = TRUE" name="inv_top_level"/>
<datamodel iumlb:refinement="2">
  <data expr="false" id="Gate_In.Block" iumlb:type="BOOL"/>
</datamodel>
<!-- Other model details -->
<state id="BLOCKED">
  <transition cond="[On_In.CardAccept==true]" target="UNBLOCKED">
    <iumlb:guard name="gd1" predicate="On_In.CardAccept==true" refinement="2"/>
    <assign expr="true" location="Gate_In.Block" iumlb:refinement="3"/>
  </transition>
  <onentry>
    <assign expr="true" location="Gate_In.Block"/>
    <assign expr="false" location="On_In.Reset"/>
  </onentry>
  <onexit>
    <assign expr="false" location="Gate_In.Block"/>
  </onexit>
  <iumlb:invariant predicate="Gate_In.Block == TRUE" name="GateCondition"/>
</state>
```

Fig. 1. Part of SCXML model including iumlb extension elements

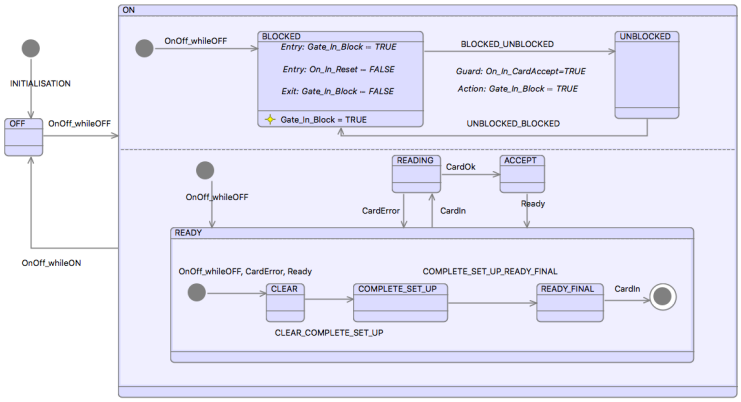

Fig. 2. State-machine diagram in iUML-B at refinement level 3 (partially annotated with guards and actions)

translation. An EMF meta-model for SCXML is available from the Sirius [1] project. It supports generic model loading capabilities for new namespace extensions. Hierarchical nested state charts are translated to similar corresponding state-machine structures in iUML-B in a series of refinement levels as directed in the SCXML *iumlb* extensions.

References

1. Eclipse Foundation. Sirius project website. https://eclipse.org/sirius/overview.html, 2016.
2. D. Steinberg, F. Budinsky, and E. Merks. *EMF: Eclipse Modeling Framework*. Eclipse (Addison-Wesley). Addison-Wesley, 2009.
3. W3C. SCXML specification website. http://www.w3.org/TR/scxml/, 2015.