Responses to Reviewers' Comments

for paper titled
Refinement of Statecharts with Run-to-Completion Semantics
by
K. Morris, C. Snook, T.S.Hoang, R. Armstrong, and M. Butler

Thanks We would like to thank the anonymous reviewers for their constructive comments and providing us with the chance to improve our paper. Below we summarise our major changes and address the problems and questions raised by the referees.

Reviewer #1:

```
Contribution:
Authors present an encoding scheme to be able to handle the
abstraction (and refinement) of statechart models. The contribution is
mainly focused onto the special encoding and the implementation. The
contribution is enough for a workshop paper and it could be a nice
piece of work to be shown in the workshop.
However, the approach seems not to scale well, so I would suggest the
authors evaluate their work using more complex (industrial)
verification problems.
```

Response: As discussed in the paper we believe it will scale well provided certain unusual styles of modelling are avoided. However, evaluation on larger models is definitely worthwhile. The following is added to conclusions.

> "We will also demonstrate the scalability of the translation on more realistic industrial examples. The Haemodialysis Machine case study(cite) from the ABZ 2016 conference would make a good test case since its highly sequential processes are natural for a statechart representation and results can be compared with existing iUML-B solutions(cite). The ERTMS Hybrid Level 3 case study(cite) from the ABZ 2018 conference is also an industrial example which would test the method. This case study would require lifting of the output models to a generalised set of instances using a model composition technique that we have been developing for this purpose."

Reviewer #1:

```
Typos and suggestions:

an intermediate refinement level by translation into Event-B --> an
intermediate refinement level by A translation into Event-B

Trigger events are queued when they are raised and then one is
dequeued --> Trigger events are queued when they are raised, and
then one is dequeued

This is repeated until no transitions are enabled and then the next
trigger --> This is repeated until no transitions are enabled, and
then the next trigger

The ASIC starts by initialising the buzzer, this involves sending
```

```
--> The ASIC starts by initialising the buzzer; this involves
sending

In practice we wish to leverage --> In practice, we wish to leverage
for the messages send between the system components --> for the
messages sent between the system components

right hand region --> right-hand region
```

Response: Done. Thank you very much.

Reviewer #2:

```
Nevertheless, there are some significant problems with
the work from my perspective, starting from motivation. There have
been many, many StateCharts semantics proposed over 20 years and any
proposal for new semantics much have a clear reason for existing.
The current approach, while having novelty in that it both supports
hierarchical refinement and run-to-completion behavior, needs a
stronger motivation in terms of solving design challenges faced by
real engineers.
```

Response: The following text is added to the 2nd last paragraph on page 2.

> "The motivation of the work is entirely driven by the industrial partner, who feels that the current sematics for Statecharts is insufficient for formal verification."

Reviewer #2:

```
The current semantics is one of non-determinism as to which events
will fire in a given microstep.  While this allows refinement, it
does not admit a deterministic execution strategy, which means that
it is not currently useful for generation of implementations.  In
addition, it appears that it is possible that events can be "lost"
and the system may choose not to execute them.  The authors must
provide additional justification that this is a sensible semantics;
I suspect that it will be surprising to most users who are familiar
with StateCharts.
```

Response: The non-determinism is removed in refinements before implementation. Abstraction often entails non-determinism. I.e., the possibility of losing an event is resolved one way or the other when we reach a decision on whether it should be fired or not. The following is added "The non-determinism is useful to allow abstractions which facilitate verification proofs but must be removed in refinements to reach a design suitable for implementation."

Reviewer #2:

```
When adding transitions at lower layers, it was
not clear to me how these new transitions fit into the cross-product
translation used to determine which set of transitions would fire.
Perhaps this is handled layer-by-layer, but then each layer would
need a scheduler and this was not discussed.
```

Response: Yes each layer makes a new cross product. The following is added "Each refinement produces a new set of events representing the (possibly extended) transition combinations that may occur at that level of refinement." (We already explain in the sentence before that these events replace the scheduler).

Reviewer #2:

```
The goal appears to be to try to create something that resembles
StateCharts but is bound to the semantics of Event-B.  In this case,
the authors should also justify why Event-B is a suitable language
for representing Statecharts behavior.  In this case, it appeared
that the target language made some aspects of behavior difficult to
formalize.
```

Response: We have added to the introduction section a couple of sentence regarding the advantages of using the Event-B within the Rodin tool. "The Event-B modelling method provides the logic and refinement theory required to formally analyse a system model. The open-source Rodin provides support for Event-B including automatic theorem provers and model checking capabilities. iUML-B augments the Event-B language with a graphical interface including state-machines."

Reviewer #2:

```
In addition, I would have liked a more formal description of the
translation, or at least a reference to a more formal description of
the translation than the prose description of different model
aspects.
```

Response: We feel that the current description of the translation is at the right level for understanding, supported by the running example. More formal/technical descriptions would be otherwise disruptive to the flow of the paper.

Reviewer #3:

```
Page 1:

* Line -2, it says:

"While functional properties (usually) can be tested, safety,
security and reliability properties
(usually) must be proved formally."

"Can" ... and ... "must" ... why?
```

Response: We have revised the sentence. "While functional properties (usually) can be tested, the need for instantiation and state space explosion can make testing of safety, security and reliability properties intractable. Therefore, such properties must be proved formally."

Reviewer #3:

```
Page 2:

* Top: The first paragraph can be expressed more clearly. That is:
what is the approach taken? Is UML state charts mapped to iUML-B or
not?
```

Response: The following is added "The binding is facilitated by translating to iUML-B, a diagrammatic modelling notation for Event-B."

Reviewer #3:

```
Page 2:

* Second paragraph appears complicated writing.
```

Response: We feel it is well written and understandable and do not understand why the reviewer thinks it is complicated writing. Perhaps the changes made in other areas of the introduction help the reviewer.

Reviewer #3:

```
Page 3:

* Listing 1: I think I would name that variable "run2completion" differently.
```

Response: Changed "run2completion" to "completion".

Reviewer #3:

```
Page 4:

* Mid, it is not clear what this means:

"The diagrammatic models are contained within an Event-B machine"
```

Response: Agree - the containment is an unnecessary technical detail. Changed "are contained within" to "relate to".

Reviewer #3:

```
Page 4:
  and this as well:

 "while Event-B events are expected to already exist to represent the transitions."
```

Response: Agree - this is an unnecessary technical detail. Deleted "while Event-B events are expected to already exist to represent the transitions".

Reviewer #3:

```
Page 5:

* It would seem more natural to have one state variable ranging over
an enumerated type of states, rather than having a Boolean flag for
each state.
```

Response: In the interest of clarity we have added the following text: "iUML-B also provides the option of an alternative translation with a single state variable ranging over an enumerated type of states, however, the boolean representation of each state is more natural for a user to reference in SCXML guards and actions."

Reviewer #3:

```
* Section 3: be more clear about what this system does. Is it just a
buzz that goes off when a sensor senses something?
```

Response: The case study is about an enabling mechanism for a security system which is confidential and we feel that full details of the system will not bring any extra points to our contribution.

Reviewer #3:

```
* Mid:

"At this abstraction the spi_done triggered, which ..."
->
" At this abstraction there is only one event, spi_done, which ..."
```

Response: The text has been changed to: "At this abstraction the spi_done trigger, which indicates that the SPI system has finished, is an internal trigger that can be fired at any time."

Reviewer #3:

```
* Line -9: "on an Idle state" -> "in an Idle state".
```

Response: Fixed, thank you very much.

Reviewer #3:

```
* Line -6: "while SPI subsystem ..." -> "while the SPI subsystem ..."
```

Response: Fixed, thank you very much.

Reviewer #3:

```
Page 6:

* Figure 2: this looks like a way of modeling event refinement,
which seems different from state refinement. Perhaps comment on
this.
```

Response: To improve clarity we have revised the text to clearly state that the refinement used is superposition. "In a subsequent level of refinement, shown in Fig. 2b, the designer uses superposition refinement to add a parallel state representing the SPI subsystem."

Reviewer #3:

```
* Line -7: "This safety property is introduced ..." : where, when?
```

Response: Agree - Here we are talking about the problem not the model. Changed to "This safety property should hold at the first refinement and be preserved in all future refinements"

Reviewer #3:

```
* Figure 2(b): "last_byte_send" -> "last_byte_sent".
```

Response: Agree - We have modified the figure.

Reviewer #3:

```
Page 7:

* Mid: "Ancillary data, with corresponding actions to alter it, can
..." - something is wrong with this sentence.
```

Response: We have revised the sentence to: "Actions that modify ancillary data can be added to transitions."

Reviewer #3:

```
* Line -4: I don't understand: "Note that external triggers are always unguarded ...".
```

Response: We have revised the sentence for clarification. Changed to "External triggers represent inputs to the model. If no restrictions are imposed on the inputs then the events that raise external trigger are always unguarded and cannot be refined."

Reviewer #3:

```
Page 8:

* The next couple of pages is a lot of English without some structure to hang it up on.
```

Response: The section is an in-depth discussion about the reasoning behind the translation and we would prefer not to introduce gratuitous (sub-)headings.

Reviewer #3:

```
Page 9:

* The three bullets (refinement, invariant, guard): there is mention of the parent state. Not sure I
```

Response: By parent we mean the element that owns (or contains) the element that is being discussed. This is a very common analogy.

Reviewer #3:

```
Page 10:

* In general, the explanation of the example could be improved a lot.

Page 12:

* I had a hard time reading this.
```

Response: Without more specific details we are not sure what needs improving in these sections. Other reviewers commented that the paper is easy to follow so we have not changed anything in response to these comments.

Reviewer #3:

```
Page 14:

* Mid: "SCXML future..TransitionSet" - is this correct?
```

Response: It was intended as shorthand for the three events: 'SCXML_futureUntriggeredTransitionSet', 'SCXML_futureInternalTransitionSet', 'SCXML_futureExternalTransitionSet' but we have now listed them explicitly.