

Scenario Checker: An Event-B tool for validating abstract models

Colin Snook, Thai Son Hoang, Asieh Salehi Fathabadi, Dana Dghaym, and Michael Butler

ECS, University of Southampton, Southampton, U.K.

{a.salehi-fathabadi, cfs, t.s.hoang, d.dghaym, m.j.butler}@soton.ac.uk

The Scenario Checker is a plugin tool for the Rodin platform for Event-B. It allows scenarios to be animated on Event-B models for validation purposes. Scenarios can be recorded, re-played and extended. The model can be annotated to distinguish and prioritise internal events and designate private variables. During *recording*, events that are designated as internal are automatically executed when enabled so that a form of ‘run to completion’ (or *big step*) is provided to represent the systems responses to changes in its environment. This allows the user to focus on developing the scenario in the environment while efficiently executing the response of the system. If necessary the user can take control of the response by executing internal events singly. This may be useful when the model still contains non-deterministic behaviour. The recorded sequence of external events and the values of non-private variables at each step can be saved in a scenario file. During *playback*, the sequence of external events to be executed is taken from the recorded scenario file while the internal events are again fired automatically. Hence the same scenario can be replayed after the model has been changed in order to test that scenario is still correctly executed in the new version of the model. At the end of each big step, critical (i.e. non-private) variables are compared with previous recorded values in order to highlight deviations. While replaying a scenario, the tool can be changed to record mode at any point so that an alternative ending to the scenario can be explored. This allows alternative scenarios to be efficiently developed from a common preamble.

The Scenario Checker is based on the ProB model checker and can be run in parallel with state visualisation tools such as BmotionStudio (which is included within ProB) and UML-B State-machine animation. The tool consists of the following views:

- Scenario Checker Control Panel view consisting of buttons to change the mode of the tool and to restart or save the recording. It also allows external events to be selected in record mode or shows the next external event to be executed in replay mode.
- Scenario Checker State view showing the state of non-private variables and, during replay mode, highlighting any differences.
- Scenario Checker Console view showing the execution of big step runs and other significant events.

The Scenario Checker is designed to allow abstract models to be validated. This implies that correspondingly abstract scenarios will be developed and refined (or

abstracted) in line with the model refinements. We have proposed a method for scenario based modelling (using the Scenario Checker) in [1]. Figure 1 shows the scenario checker in record mode (bottom 3 views) with BMotionStudio (top left) and Statemachine animation (top right). Figure 2 shows the scenario checker in playback mode, replaying the scenario recorded in Figure 1.

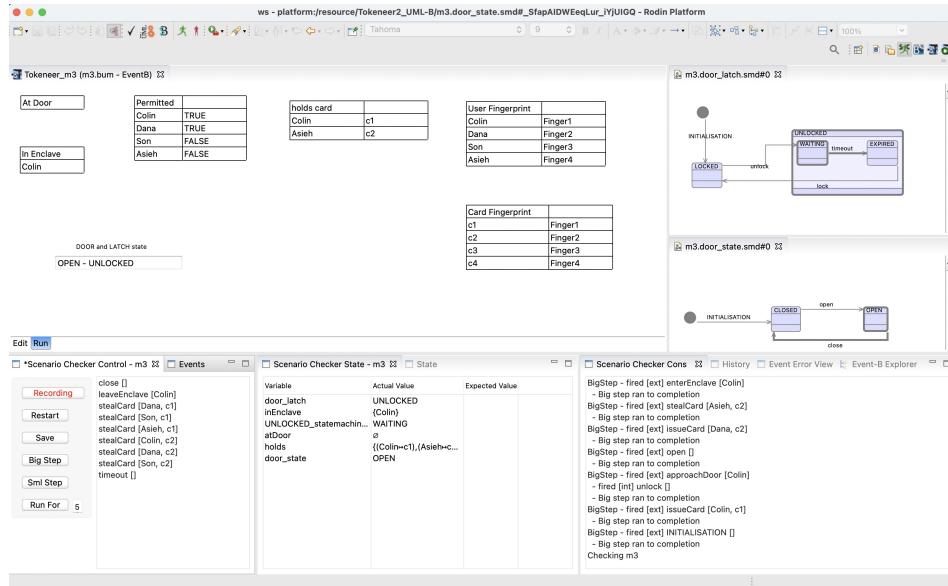


Fig. 1: Scenario Checker in recording mode

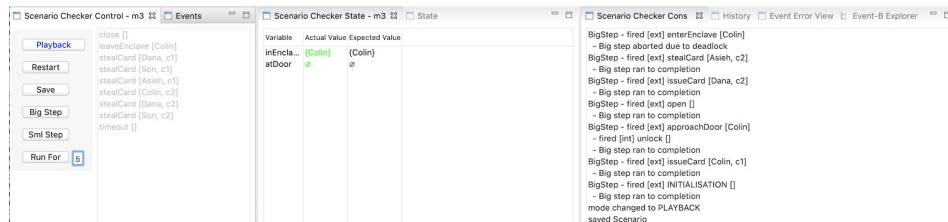


Fig. 2: Scenario Checker in playback mode

Acknowledgement: This work is supported by the following projects: - Hi-Class project (113213), which is part of the ATI Programme, a joint Government

and industry investment to maintain and grow the UK's competitive position in civil aerospace design and manufacture.

References

1. Colin Snook, Thai Son Hoang, Dana Dghaym, Asieh Salehi Fathabadi, and Michael Butler. Domain-specific scenarios for refinement-based methods. *Journal of Systems Architecture*, 112:101833, 2021.