

Scenario Checker

An Event-B tool for validating
abstract models

Colin Snook,

Thai Son Hoang, Asieh Salehi Fathabadi, Dana Dghaym,
Michael Butler

Contact: cfs@soton.ac.uk

Verification versus Validation

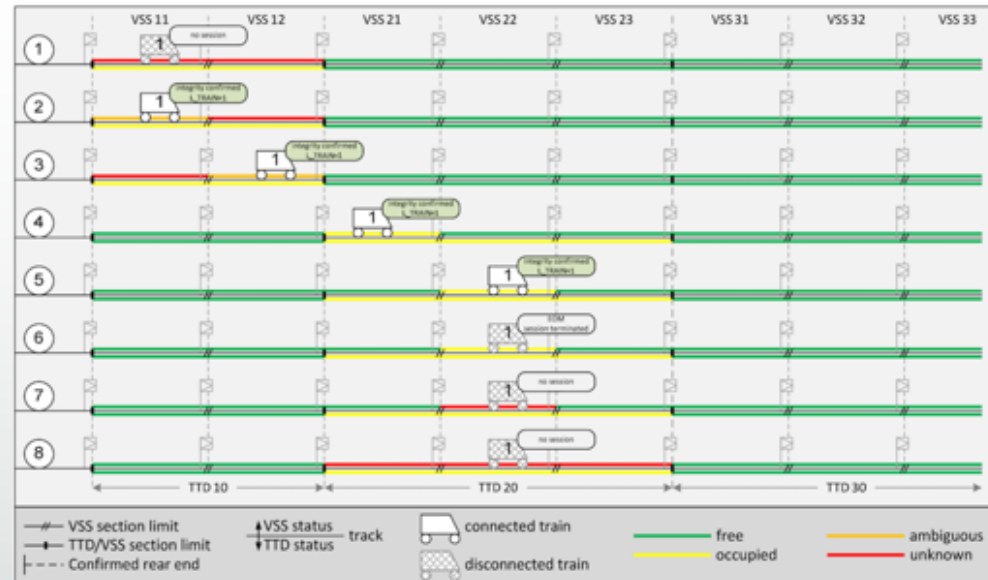
- Verification focuses on checking the model is well-defined and consistent.
 - E.g. satisfies the invariants and refines previous abstractions
 - Event-B has proof for this.
 - But verification does not tell us whether the model is what we want.
- Validation
 - Does it behave in a useful way.
 - Subjective assessment of domain experts
 - Demonstrate the behaviour of the model
 - Scenarios

Modelling ETCS Hybrid Level 3

- ABZ 2018 – Case study
 - Detailed specification
 - Explained by scenarios

Dghaym, Poppleton, Snook, (2018) Diagram-Led Formal Modelling Using iUML-B for Hybrid ERTMS Level 3. In *International Conference on Abstract State Machines, Alloy, B, TLA, VDM and Z (ABZ2018)*. LNCS Volume 10817 pp338-352.

- We found it difficult to understand the specification without looking at the scenarios.
- The scenarios helped us to test (Validate) the verified model
- BUT.. for abstract models we needed abstract scenarios



Behaviour driven model development

- A process for scenario driven model development.
- Abstract scenarios can be refined OR concrete scenarios abstracted
- DSL for scenarios (domain specific)
 - Based on gherkin/cucumber (from BDD)
- Scenario Execution:
 - Script – Regression testing
 - User driven - Acceptance testing
 - **SCENARIO CHECKER**

Snook, Hoang, Dghaym, Fathabadi, and Butler. (2021) Domain-specific scenarios for refinement-based methods. *Journal of SystemsArchitecture*, 112:101833, 2021.

```
1 Given Train1 stood at TTD10.VSS11
2 And Train1 is disconnected
3 And TTD10 is OCCUPIED
4 And TTD20 is FREE
5 When Train1 connects
6 Then Train1 connected
7 When Train1 enters TTD10.VSS12
8 When Train1 leaves TTD10.VSS11
9 When Train1 enters TTD10.VSS21
10 Then TTD20 is OCCUPIED
11 When Train1 leaves TTD10.VSS12
12 Then TTD10 is FREE
13 When Train1 enters TTD10.VSS22
14 When Train1 leaves TTD10.VSS21
15 When Train1 disconnects
16 Then Train1 disconnected
```

Scenario checker - concepts

- Model annotations
 - *Internal* events are fired automatically when enabled
 - *Private* variables are ignored when comparing scenario executions during playback
- Two modes
 - Manually **Record** a new scenario
 - Automatically **Playback** a previously recorded scenario
- Three Views
 - Control Panel – user controls and selection of *External* events
 - State View – display state of *Public* variables
 - Console – history of animation and user actions

Scenario Checker – Recording Mode

- User selects external events to fire
 - Internal events fire automatically until completion
- Scenario can be saved at any point
 - Sequence of External events fired
 - State of public variables after each big step.

Big Step

The screenshot displays the Scenario Checker software interface in recording mode. It is divided into three main panels:

- Control Panel (Left):** Contains buttons for 'Recording', 'Restart', 'Save', 'Big Step', 'SmI Step', and 'Run For 5'. Below these buttons is a list of external events to be fired:


```
close []
leaveInclave [Colin]
stealCard [Dana, c1]
stealCard [Son, c1]
stealCard [Ashih, c1]
stealCard [Colin, c2]
stealCard [Dana, c2]
stealCard [Son, c2]
timeout []
```
- State Table (Middle):** A table showing the current state of public variables.

Variable	Actual Value	Expected Value
door_latch	UNLOCKED	
inInclave	{Colin}	
UNLOCKED_stateMachin...	WAITING	
atDoor	0	
holds	{Colin=c1, Ashih=c...	
door_state	OPEN	
- History Log (Right):** A list of recorded 'Big Step' events, each followed by a sub-list of internal actions that occurred:


```
BigStep - fired [ext] enterInclave [Colin]
- Big step ran to completion
BigStep - fired [ext] stealCard [Ashih, c2]
- Big step ran to completion
BigStep - fired [ext] issueCard [Dana, c2]
- Big step ran to completion
BigStep - fired [ext] open []
- Big step ran to completion
BigStep - fired [ext] approachDoor [Colin]
- fired [int] unlock []
- Big step ran to completion
BigStep - fired [ext] issueCard [Colin, c1]
- Big step ran to completion
BigStep - fired [ext] INITIALISATION []
- Big step ran to completion
Checking m3
```

Scenario Checker – Playback Mode

- Scenario checker selects external events to fire according to the scenario
 - Internal events fire automatically until completion
- Change to Recording mode at any point
 - Allows new scenarios to be created efficiently
 - Same pre-amble with alternative endings

The screenshot displays the Scenario Checker software interface with four main panels:

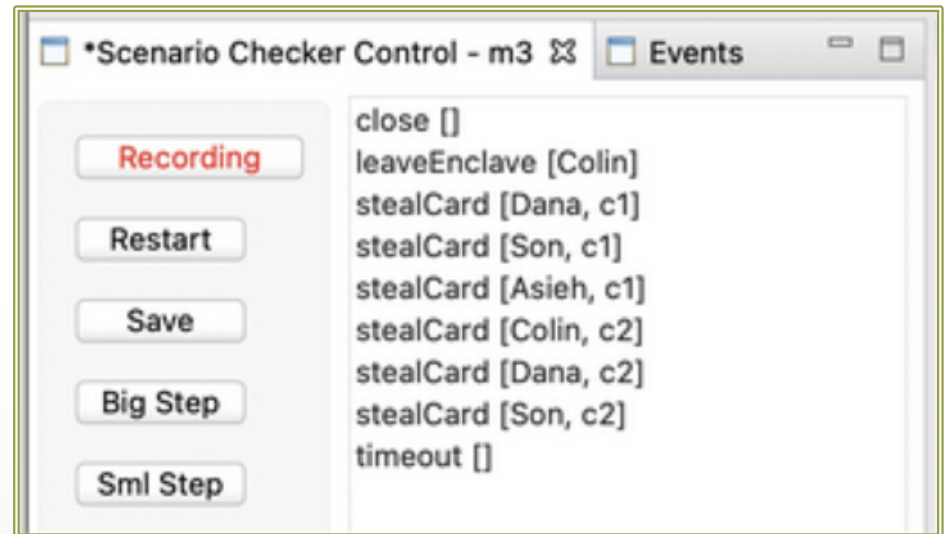
- Scenario Checker Control - m**: Contains playback controls: Playback, Restart, Save, Big Step, Sm1 Step, and Run For (set to 5).
- Events**: Lists external events: open [], issueCard [Dana, c2], stealCard [Dana, c1], stealCard [Son, c1], stealCard [Asieh, c1], leaveDoor [Colin], and timeout [].
- Scenario Checker State - m3**: A table showing the current state of variables.

Variable	Actual Value	Expected Value
atDoor	(Colin)	(Colin)
door_latch	UNLOCKED	UNLOCKED
UNLOCKED_statemachin...	WAITING	WAITING
- Scenario Checker Console - m3**: A log of events and actions, including:
 - BigStep - fired [ext] approachDoor [Colin]
 - fired [int] unlock []
 - Big step ran to completion
 - BigStep - fired [ext] issueCard [Colin, c1]
 - Big step ran to completion
 - BigStep - fired [ext] INITIALISATION []
 - Big step ran to completion
 - Restarted
 - BigStep - fired [ext] open []
 - Big step ran to completion
 - BigStep - fired [ext] approachDoor [Colin]
 - fired [int] unlock []
 - Big step ran to completion

Scenario Checker Views:

Control Panel

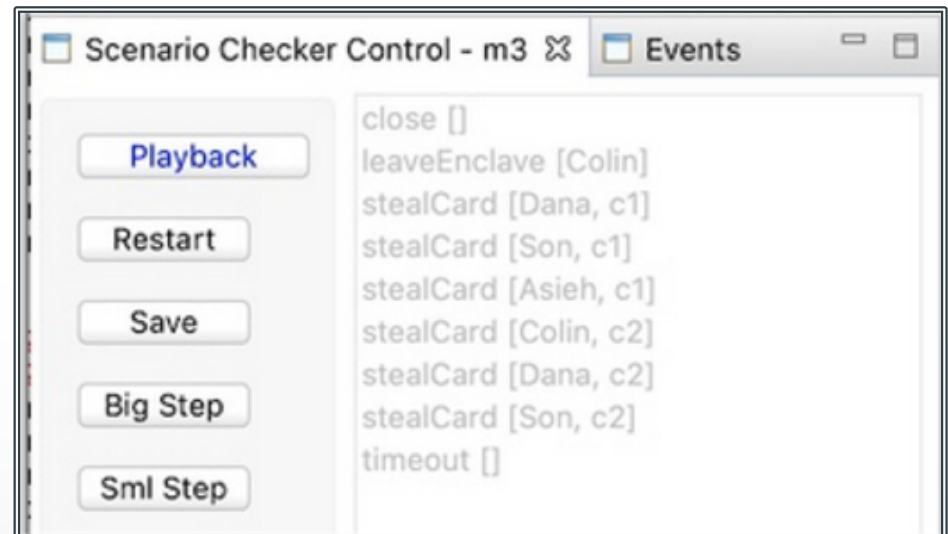
- Control Buttons
 - Change mode, restart, save
- Big Step
 - Fire Big-step starting with the selected external event
- Sml Step
 - Fire any enabled internal event or selected external event
- List of enabled External events
 - Select next external event
 - (or Double click to fire big step)



Scenario Checker Views:

Control Panel

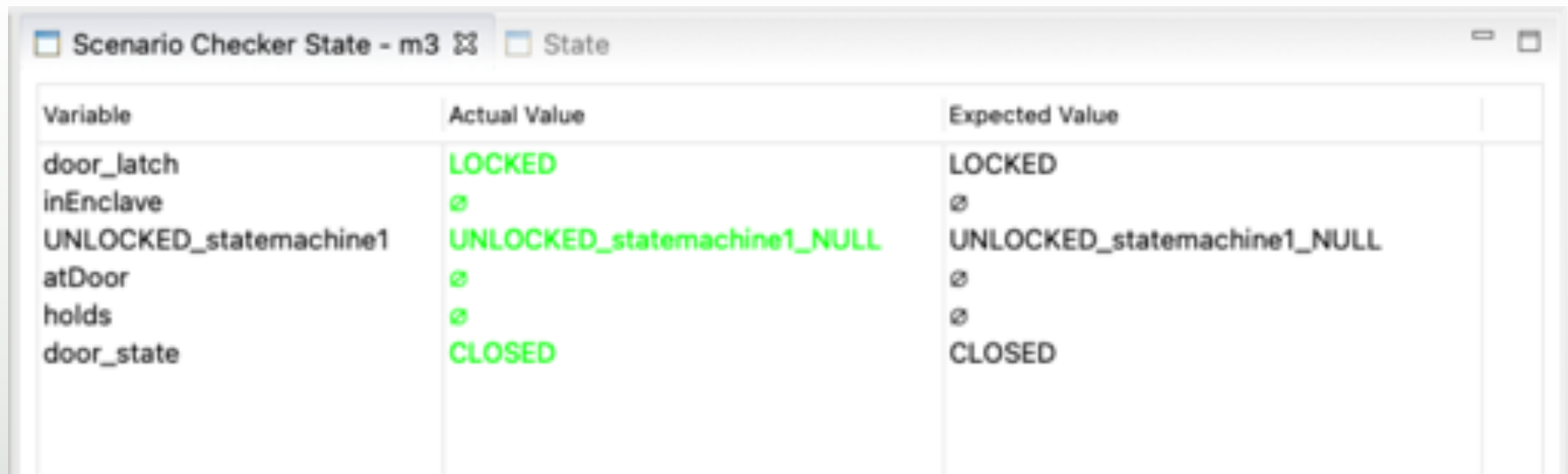
- Control Buttons
 - Change mode, restart, save
- Big Step
 - Fire big-step starting with the next external event in the recorded scenario
- Sml Step
 - Fire any enabled internal event or next external event



Scenario Checker Views:

State View

- Values of public variables
- Comparison with previously recorded values
 - (playback mode only)



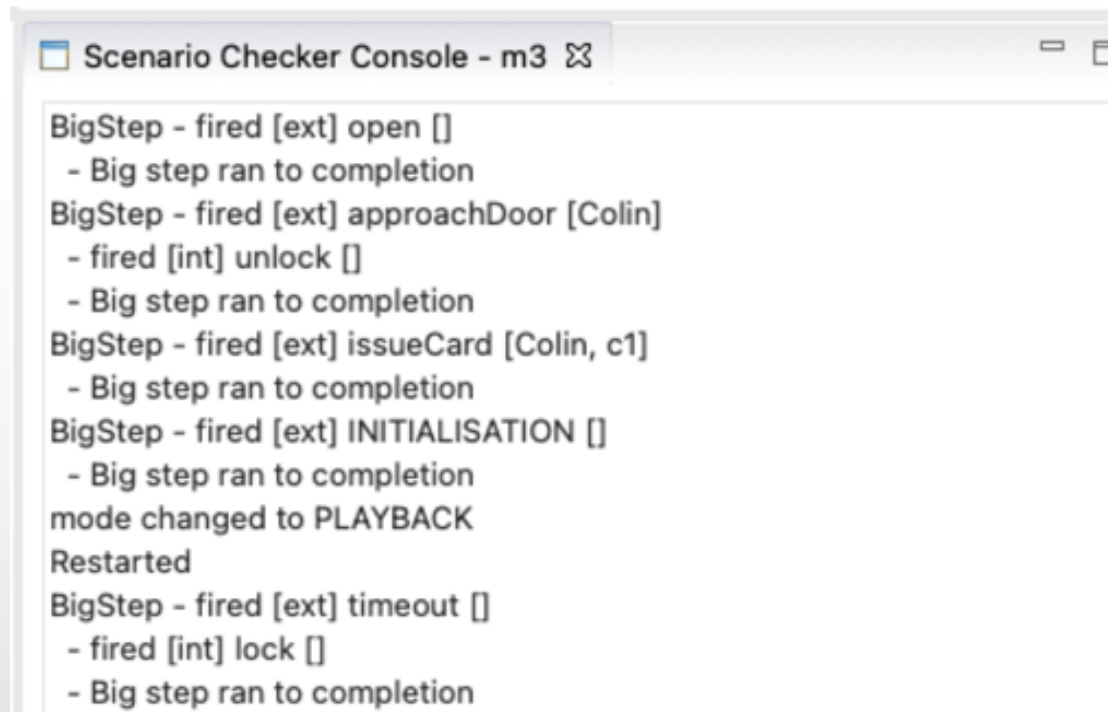
The screenshot shows a window titled "Scenario Checker State - m3" with a sub-tab "State". It displays a table comparing actual and expected values for several variables. The actual values are shown in green text.

Variable	Actual Value	Expected Value
door_latch	LOCKED	LOCKED
inEnclave	⊘	⊘
UNLOCKED_statemachine1	UNLOCKED_statemachine1_NULL	UNLOCKED_statemachine1_NULL
atDoor	⊘	⊘
holds	⊘	⊘
door_state	CLOSED	CLOSED

Scenario Checker Views:

Console

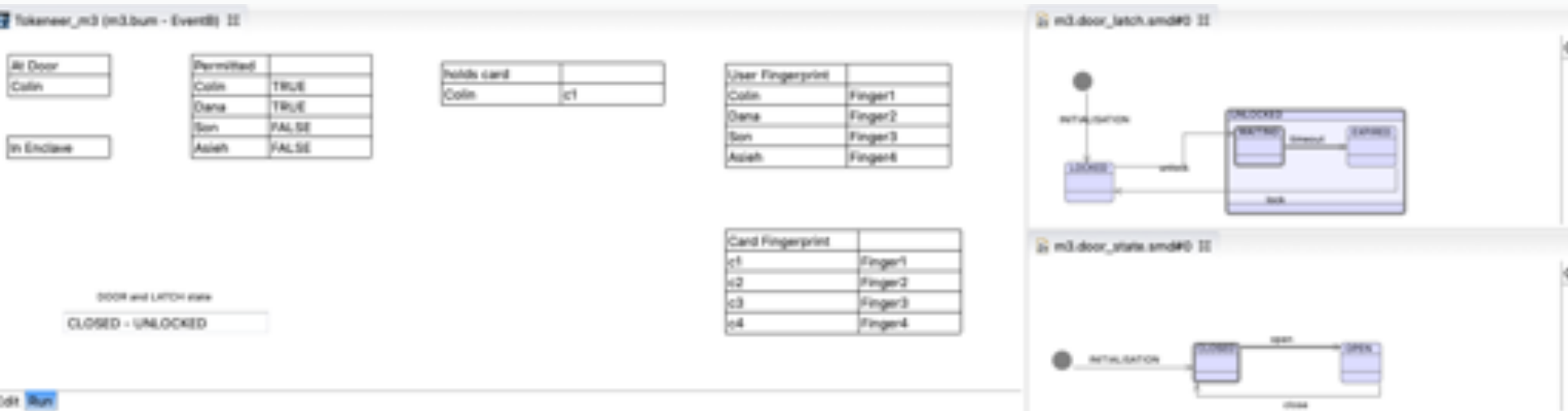
- History of execution (arranged in big steps)
- Other important events (restart, saved, mode changes etc.)



```
Scenario Checker Console - m3 ✕  
BigStep - fired [ext] open []  
  - Big step ran to completion  
BigStep - fired [ext] approachDoor [Colin]  
  - fired [int] unlock []  
  - Big step ran to completion  
BigStep - fired [ext] issueCard [Colin, c1]  
  - Big step ran to completion  
BigStep - fired [ext] INITIALISATION []  
  - Big step ran to completion  
mode changed to PLAYBACK  
Restarted  
BigStep - fired [ext] timeout []  
  - fired [int] lock []  
  - Big step ran to completion
```

State Visualisation

- State can be visualised using other tools
 - E.g.
 - UML-B Statemachine animation,
 - BMotionStudio
 - Animation is synchronised via our ProB interface plugin
 - Easy to add new synchronised animations
 - Easy to adapt to ProB API changes (e.g. new version)



Summary

- Validation is important
- Efficient tool for managing scenarios
 - Record helps create scenarios
 - Playback and extend helps create alternative scenarios
 - Big step (run to completion) saves a lot of time
- Acceptance testing of models
 - State visualization using existing visualization tools
- Future work
 - Script mode for regression testing

Thank you

Questions?