Demo: Graphical Hybrid Automata with Simulink and Stateflow

Akshay Rajhans
MathWorks
Natick, MA
akshay.rajhans@mathworks.com

Srinath Avadhanula
MathWorks
Natick, MA
srinath.avadhanula@mathworks.com

Alongkrit Chutinan
MathWorks
Natick, MA
alongkrit.chutinan@mathworks.com

Pieter J. Mosterman
MathWorks
Natick, MA
pieter.mosterman@mathworks.com

Fu Zhang
MathWorks
Natick, MA
fu.zhang@mathworks.com

ACM Reference Format:
Akshay Rajhans, Srinath Avadhanula, Alongkrit Chutinan, Pieter J. Mosterman, and Fu Zhang. 2018. Demo: Graphical Hybrid Automata with Simulink and Stateflow. In *HSCC '18: 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week), April 11–13, 2018, Porto, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3178126.3187007

1 INTRODUCTION

Simulink® and Stateflow® are tools for Model-Based Design that support a variety of mechanisms for modeling hybrid dynamics. Each of these tools has different strengths. The HSCC 2018 tool paper [1] presents a new modeling construct, called graphical hybrid automaton, which combines these strengths to enable graphical modeling of hybrid dynamics within a single Stateflow chart. This demo is intended to demonstrate the new modeling construct, the user interface, as well as various modeling workflows. Interested participants would get to see key features of this new formalism in action via the examples from the companion paper [1], as well as try the new functionality themselves.

2 REVIEW EXISTING ALTERNATIVES

Alternatives to the graphical hybrid automaton and their tradeoffs are discussed.

2.1 Existing alternatives

Three types of approaches to modeling hybrid dynamics in Simulink and Stateflow have been available.

(1) **Type I: Only Simulink.** Simulink provides a variety of blocks for modeling mode switches using (i) explicit mechanisms such as switch blocks and conditionally-executed subsystems and (ii) implicit mechanisms such as saturation and external reset.
(2) **Type II: Only Stateflow.** Since release R2007b, a textual syntax for modeling continuous dynamics in the 'during' actions of Stateflow states allows the modeling of hybrid automata.
(3) **Type III: Mixed Simulink and Stateflow.** This approach uses Simulink blocks kept outside the Stateflow chart to model continuous dynamics. Signal lines between Simulink and Stateflow are used for (i) guard conditions in Stateflow that depend on state(s) of one or more Simulink blocks, (ii) control outputs from Stateflow that are used to drive the subsystem corresponding to the active mode, and (iii) reset values computed in Stateflow to drive the external initial condition (EIC) port of a Simulink block.

2.2 Tradeoffs between the alternatives

Each of the three approaches have different tradeoffs. For example, when modeling discrete switches using Simulink blocks, the discontinuity detection could be distributed throughout the model and could possibly be evaluated even when not strictly necessary. Writing continuous dynamics textually is error-prone less powerful than graphical modeling and can be error prone as long complicated textual dynamics are hard to read and debug. Lastly, managing mode-switching and state handoff must be orchestrated carefully with state and external initial condition (EIC) ports in Types I and III, which may lead to diagram clutter.

In this demo, the participants would be able to see examples of Type I, II, and III models, in order to compare and contrast these modeling considerations.

3 CONSTRUCTING GRAPHICAL HYBRID AUTOMATA

Graphical hybrid automata is the new modeling formalism presented in the companion paper [1]. Use of the formalism involves (i) creating a new type of Stateflow state, called a Simulink based state, whose internal dynamics are defined as if it were a Simulink subsystem, and (ii) creating remote textual and/or graphical state access for reading and modifying the continuous state in transition guards and reset actions.

3.1 Constructing Simulink® based states

There are two workflows for constructing Simulink based states in Stateflow.

(1) A straightforward workflow is to drag-and-drop a Simulink based state from the Stateflow palette, double click on it,

HSCC '18, April 11–13, 2018, Porto, Portugal

ACM ISBN 978-1-4503-5642-8/18/04.
https://doi.org/10.1145/3178126.3187007

and start modeling the continuous dynamics using Simulink blocks.
(2) Another workflow is to copy an existing Action subsystem in Simulink with dynamics already modeled inside it, and then paste it onto the Stateflow canvas.

Demo participants would be able to try out both workflows for creating Simulink based states.

3.2 Defining remote access to states of a block

Guards and reset actions on transitions in Stateflow can depend on and manipulate the states of a block by means of two mechanisms:

(1) graphical remote access using State Reader and State Writer blocks, and
(2) textual remote access using Stateflow syntax (dot notation).

Demo participants would be able to try out both of these mechanisms for creating remote access to block state.

3.3 Reuse via library links

When there are more than one Simulink based states in a model that have the same functionality, model reuse can be achieved by creating library links in Simulink, rather than creating multiple copies. Demo participants would be able to see this library link usage in action.

4 REVIEWING EXAMPLE MODELS

The companion tool paper [1] presents four examples highlighting different use cases, workflows, and features.

(1) **Modeling a clutch.** This model is used as a running example to describe various elements of the new formalism.
(2) **Modeling a pole vault jump.** This pole vault model is intended to showcase the use of **arbitrarily different continuous states** in different modes and the copy-paste workflow of constructing Simulink based states.
(3) **Modeling a mode-switching controller.** This mode-switching example is intended to showcase the use of **discrete-time dynamics**, **built-in timers**, and **reuse** via library links.
(4) **Modeling a cardiac cell.** This example is intended to showcase the **conciseness** of graphical hybrid automata with respect to the existing graphical modeling alternatives.

Demo participants would be able to view, edit, simulate, and compare these models with existing alternative approaches as applicable.

REFERENCES

[1] A. Rajhans, S. Avadhanula, A. Chutinan, P. J. Mosterman, and F. Zhang. Graphical modeling of hybrid dynamics with simulink and stateflow. In *HSCC 2018.* Accepted.