Möbius Trace Analysis with Traviando

Ruth Lamprecht, Peter Kemper
Department of Computer Science
College of William and Mary
Williamsburg, VA, USA
{rlampy,kemper}@cs.wm.edu

Abstract

Model verification and validation is crucial to the credibility of a simulation study. The ability to investigate the detailed behavior of a complex stochastic model that is simulated in a powerful modeling environment like Möbius is important to identify root causes of errors in models or to gain confidence that a model performs as expected. In this paper, we briefly describe how the Traviando trace analyzer sheds light on what happens in a simulation run of a Möbius model.

1. Möbius Modeling

Möbius [1, 4] is a multi-paradigm multi-solution framework for the performance and dependability assessment of systems. It supports multiple ways to create large and complex models in a compositional manner. The ease with which complex models can be analyzed motivates a complementary visualization for the detailed behavior of a simulation. Being able to track details of a simulation trace helps an analyst in the verification and validation of a model. Möbius users can make use of Traviando [2, 5], a trace visualizer and analyzer, that serves as a backend tool for this purpose.

Modeling with Möbius follows a clear conceptional line of thought; a model specifies the dynamic behavior of a stochastic discrete event system and is built from basic, atomic models that are composed into larger and more complex models like the Rep/Join composed model in the left of Fig. 1. Möbius supports a number of formalisms to specify atomic models including stochastic activity networks (SANs) and a stochastic process algebra (PEPA). Atomic models can be composed either by action synchronization (like the parallel composition considered in process algebra setting or the synchronization of transitions in stochastic Petri nets) or by state sharing (state variables can be shared and accessed by actions from different atomic models). The definition of measures (rewards) of interest is performed in a separate step and Möbius provides a rich set of reward types including instant of time rewards, interval of time rewards, and accumulating rewards. Möbius provides both discrete event simulation for transient and steady state analysis, as well as numerical analysis of Markov chains.

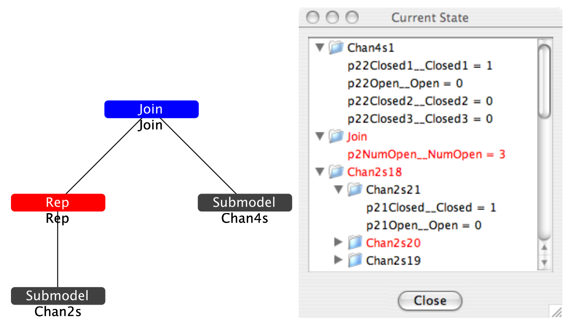

Figure 1. Rep/Join composed model with 18 replicated two state models and 1 four state model (left) and corresponding visualization of a state in a corresponding trace.

2. Möbius Trace Analysis

In Möbius, it is also possible to export a trace file of a simulation run into an XML formatted file that can be evaluated with a trace analyzer like Traviando to investigate the details of what happened in a simulation of a model. This is useful to verify that certain events happen in a trace, that particular states are frequently reached or that certain conditions hold throughout a simulation. Such a simulation trace usually contains a large amount of data so some automated support is needed to shed light on its content and to recognize anomalies or to track causes for an observed effect.

Visualization. The Traviando trace analyzer visualizes trace data as a variant of message sequence charts or UML

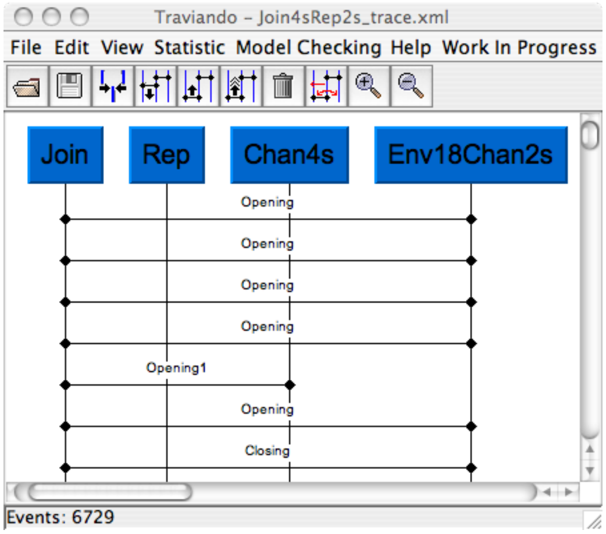

Figure 2. MSC visualization for a trace of a stochastic model of calcium signaling

sequence diagrams. It assumes that a trace contains events that relate to a finite set of processes that can be involved in an event. If an event relates only to one process it is considered a local event; otherwise it is an interaction event (with the two subclasses of directed and undirected events). To preserve the compositional structure of a Möbius model, every node in a Replicate/Join or Graph composed model is mapped to an individual process in the trace format with corresponding implications on the distribution of local actions and state variables. The description of state is assumed to result from value settings to a finite set of state variables. The state variables are partitioned with respect to the processes they belong to.

In order to make the graphical visualization as MSCs scale with a large number of processes, Traviando supports grouping processes into so-called *environments* such that all events are preserved but some events that were connecting several processes before may show as a local event of an environment process as the effect of grouping. Fig. 2 shows a trace for the model of Fig.1 where the processes for the 18 instances of submodel Chan2s are grouped into a new environment process Env18Chan2s. Fig. 1 also shows the visualization of state information, which is a tree-type structure that reflects the grouping of state variables into processes and the grouping of processes into environments. In order to make it easy to see what values change as an effect of an event, those changes are highlighted in red when an event is selected in Fig 2.

Navigation. Given a large trace file, it is important to assist a user in finding parts that are of particular interest. Traviando allows a user to browse a trace based on sequences of events. Starting with the selection of a particular event, e.g., the failure of particular component, one can either choose to visit a corresponding location in the MSC visualization (by default the first location where the event occurs) or to select among observed successor events to partition a large set of matching locations into a number of subsets and navigate by the specification of longer and longer subsequences that start with the event of interest to more specific locations of interest. This feature is particularly valuable to locate rare events or to consider groups of subsequences that frequently occur simultaneously.

Reduction. Simulation traces contain lots of data, yet sometimes in steady state analysis of stochastic performance and dependability models, a lot of the simulated behavior will repeat over and over again (of course with some variation due to the impact of pseudo random numbers). Traviando provides functionality to analyze the cyclic behavior with graphics that show if states are repeatedly visited or how the length of the trace evolves if cycles are removed (the so-called progress of a trace [3]). Traviando can also visualize the reduced trace, which is helpful to see a shortened, direct way to reach a particular state in a simulation and to track down root causes for such a behavior.

More Features for Trace Analysis. Traviando also supports various statistics on traces as well as the model checking of a trace with respect to LTL formulas. Model checking results are visualized by color highlighting in the MSC.

Other Applications for Traviando. The trace analysis functionality of Traviando has been successfully used with other tools as well, including the network simulator NS2, the APNN toolbox, and the ProC/B toolset. Some recent efforts are dedicated to visualizing witness traces generated by the SAL modelchecker. More information on Traviando can be found at [5].

Acknowledgments We acknowledge the support of the Möbius group at UIUC, in particular the work of Mike McQuinn to export trace data for Traviando.

References

[1] D. D. Deavours et al. The Möbius framework and its implementation. *IEEE TSE*, 28(10):956–969, 2002.

[2] Peter Kemper. A trace-based visual inspection technique to detect errors in simulation models. In *Winter Simulation Conference*, ACM, 2007.

[3] Peter Kemper, Carsten Tepper. Automated analysis of simulation traces - separating progress from repetitive behavior. In *QEST*, IEEE, 2007. An extended and revised version will appear in IEEE TSE.

[4] The Möbius Tool. www.mobius.uiuc.edu.

[5] Traviando. www.cs.wm.edu/~kemper/traviando.html.