The Modest Toolset: An Integrated Environment for Quantitative Modelling and Verification*

Arnd Hartmanns and Holger Hermanns

Saarland University – Computer Science, Saarbrücken, Germany

Abstract Probabilities, real-time behaviour and continuous dynamics are the key ingredients of quantitative models enabling formal studies of non-functional properties such as dependability and performance. The Modest Toolset is based on networks of stochastic hybrid automata (SHA) as an overarching semantic foundation. Many existing automata-based formalisms are special cases of SHA. The toolset aims to facilitate reuse of modelling expertise via Modest, a high-level compositional modelling language; to allow reuse of existing models by providing import and export facilities for existing languages; and to permit reuse of existing tools by integrating them in a unified modelling and analysis environment.

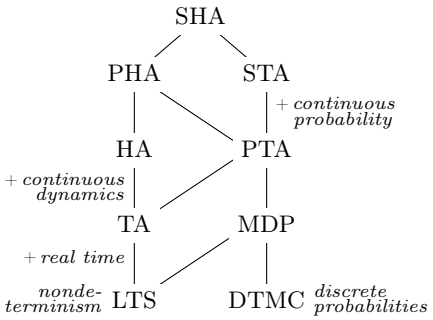
1 Introduction

Our reliance on complex safety-critical or economically vital systems such as fly-by-wire controllers, networked industrial automation systems or "smart" power grids increases at an ever-accelerating pace. The necessity to study the reliability and performance of these systems is evident. Over the last two decades, significant progress has been made in the area of formal methods to allow the construction of mathematically precise models of such systems and automatically evaluate properties of interest on the models. Classically, model checking has been used to study functional correctness properties such as safety or liveness. However, since a correct system implementation may still be prohibitively slow or energy-consuming, performance requirements need to be considered as well. The desire to evaluate both *qualitative* as well as *quantitative* properties fostered the development of integrative approaches that combine probabilities, real-time aspects or costs with formal verification techniques [1].

The Modest Toolset is an integrated collection of tools for the creation and analysis of formally specified behavioural models with quantitative aspects. It constitutes the second generation [8] of tools revolving around the Modest modelling language [7]. By now, it has become a versatile and extensible toolset based on the rich semantic foundation of networks of stochastic hybrid automata (SHA), supporting multiple input languages and multiple analysis backends.

* This work is supported by the Transregional Collaborative Research Centre SFB/TR 14 AVACS, the NWO-DFG bilateral project ROCKS, and the 7th EU Framework Programme under grant agreements 295261 (MEALS) and 318490 (SENSATION).

E. Ábrahám and K. Havelund (Eds.): TACAS 2014, LNCS 8413, pp. 593–598, 2014.



Key:

SHA	stochastic hybrid automata
PHA	probabilistic hybrid automata
STA	stochastic timed automata
HA	hybrid automata
PTA	probabilistic timed automata
TA	timed automata
MDP	Markov decision processes
LTS	labelled transition systems
DTMC	discrete-time Markov chains

Fig. 1. Submodels of stochastic hybrid automata

The Modest Toolset's aim is to incorporate the state of the art in research on the analysis of stochastic hybrid systems and special cases thereof, such as probabilistic real-time systems. In particular, it goes beyond the usual "research prototype" by providing a single, stable, easy-to-install and easy-to-use package.

In this paper, we illustrate how SHA provide a unified formalism for quantitative modelling that subsumes a wide variety of well-known automata-based models (Section 2); we highlight the Modest Toolset's approach to modelling and model reuse through its support of three very different input languages (Section 3); we give an overview of the available analysis backends for different specialisations of SHA (Section 4); and we provide some background on technical aspects of the toolset and its cross-platform user interface (Section 5).

Related work. Two tools have substantially inspired the design of the Modest Toolset: **Möbius** [9] is a prominent multiple-formalism, multiple-solution tool. Focussing on performance and dependability evaluation, its input formalisms include Petri nets, Markov chains and stochastic process algebras. **Cadp** [11], in contrast, is a tool suite for explicit-state system verification, comprising about fifty interoperable components, supporting various input languages and analysis approaches. The Modest Toolset has so far focused on reusing existing tools on the analysis side whereas Möbius and Cadp rely on their own implementations.

2 A Common Semantic Foundation

The Modest Toolset is built around a single overarching semantic model: networks of stochastic hybrid automata (SHA), i.e. sets of automata that run asynchronously and can communicate via shared actions and global variables. While action labels are used for synchronisation, a state-based approach is used for verification, i.e. the valuations of the global variables act as atomic propositions observable in properties. SHA combine three key modelling concepts:

Continuous dynamics To represent continuous processes, such as physical laws or chemical reactions, the evolution of general continuous variables over

MODEST:

```
process Channel()
{
  snd palt {
  :99: delay(2)
       rcv
  : 1: // msg lost
       {==}
  };
  Channel()
}
```

Guarded Commands:

```
module Channel
  l: [0..1]; // control loc
  c: clock;  // for delay
  invariant
    l=1 => c<=2
  endinvariant
  [snd] l = 0  -> 0.01:(l'=0)
       + 0.99:(l'=1)&(c'=0)
  [rcv] l=1 & c>=2 -> (l'=0)
endmodule
```

UPPAAL TA:

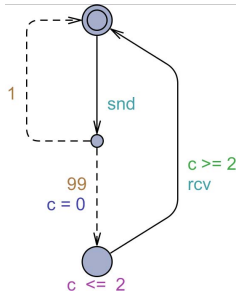


Fig. 2. Modelling a channel with loss probability 0.01 and transmission delay 2

time can be described using differential (in)equations. Continuous variables with constant derivative 1 are used as clocks to model real-time systems.

Nondeterminism To model concurrency (via an interleaving semantics) or the absence of knowledge over some choice, to abstract from details, and to represent the influence of an unknown environment, nondeterministic choices can be used. The number of choices may be finite or (countably or uncountably) infinite. The latter can be used to model nondeterministic delays.

Probability Probabilistic choices represent the case where an outcome is uncertain, but the probabilities of the outcomes are known. Such choices may be inherent to the system under study, e.g. in a randomised algorithm, or they may represent external influences such as failure rates where statistical data is available. Again, these choices may be discrete ("probabilistic") or continuous ("stochastic"), and they can be used to represent random delays.

On the syntactic representation of a SHA, each of these aspects is easy to identify. By restricting the occurrence of certain aspects, various well-known automata models appear as special cases of SHA as shown in Fig. 1. Additionally, sampling from the exponential distribution can be combined with clocks to obtain exponentially-distributed delays, allowing models based on continuous-time Markov chains to be represented as SHA, too.

3 Input Languages for Every Taste

As of the current version 2.0, the MODEST TOOLSET can process models specified in three very different input languages:

MODEST is a high-level textual modelling language. It is inspired by process algebras, but has an expressive programming language-like syntax that leads to concise models. MODEST was originally introduced with a STA semantics [7] and has recently been extended to allow the modelling of SHA [12].

Guarded Commands Probabilistic guarded commands are a low-level textual modelling language. Easy to learn with few key language constructs, it can be seen as the "assembly language" of quantitative modelling. It is the language

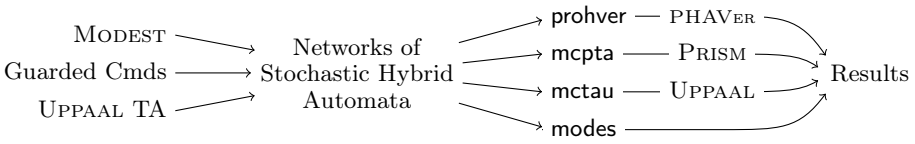


Fig. 3. Schematic overview of the Modest Toolset's components

of the Prism [17] model checker, so its support within the Modest Toolset allows the reuse of many existing Prism models.

Uppaal TA Uppaal is built upon a graphical interface to model (probabilistic) timed automata [3]. A textual language is used for expressions and to specify the composition of components. The Modest Toolset can import and export Uppaal TA models. It supports a useful subset of the language's advanced features such as parameterised templates and C-style functions.

Fig. 2 shows a comparison of the three languages for a small example PTA model. Through the use of the intermediate networks-of-SHA representation, models can be freely converted between the input languages.

4 Multiple Analysis Backends

A prime goal of the Modest Toolset is to facilitate the reuse of existing analysis tools for specific subsets of SHA where possible in order to concentrate development effort on key areas where current tool support is still lacking or non-existent. The following analysis backends are part of version 2.0 of the toolset:

prohver Computes upper bounds on max. probabilities of probabilistic safety properties in **SHA** [12]. Relies on a modified PHAVer [10] for a HA analysis.

mcpta Performs model checking of **PTA** using Prism for the probabilistic analysis; supports probabilistic and expected-time/expected-reward reachability properties in unbounded, time- and cost-bounded variants [14].

mctau Connects to Uppaal for model checking of **TA** [4], for which it is more efficient than mcpta. Automatically overapproximates probabilistic choices with nondeterminism for PTA, providing a quick first check of such models.

modes Performs statistical model checking and simulation of **STA** with an emphasis on the sound handling of nondeterministic models [5,6,16]. Its trace generation facilities are useful for model debugging and visualisation.

Fig. 3 gives a schematic overview of the input languages and analysis backends that form the Modest Toolset.

5 An Integrated Toolset

As presented in the previous sections, the Modest Toolset consists of several components and concepts. Several of its analysis backends have been developed independently and presented separately before. However, it is their combination

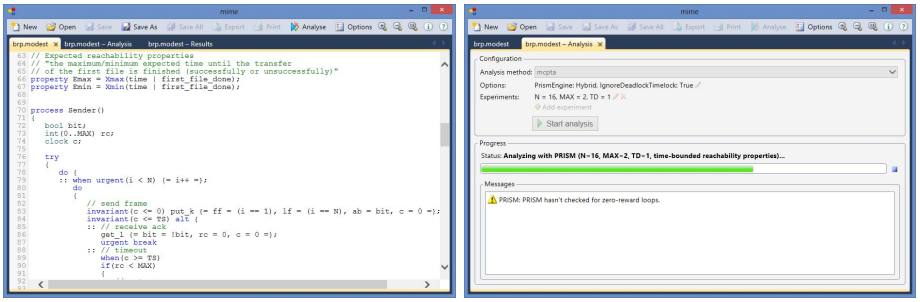


Fig. 4. The mime graphical user interface for modelling (left) and analysis (right)

and integration that give rise to the advance in utility that the toolset presents. This integration is visible in the main interfaces of the toolset:

mime is the toolset's graphical **user interface**. It provides a modern editor for the supported textual input languages and gives full access to the analysis backends and their configuration. mime is cross-platform, based on web technologies such as HTML5, Javascript and the WebSocket protocol. Fig. 4 shows two screenshots of the mime interface. For scripting and automation scenarios, all backends are also available as standalone command-line tools.

The toolset itself is built around a small set of object-oriented **programming interfaces** for input components, SHA-to-SHA model conversions, model restrictions (to enforce certain subsets of SHA) and analysis backends. Adding a new input language, for example, can be accomplished by implementing the `IInputFormalism` interface and providing a semantics in terms of networks of SHA; for mime support, syntax highlighting information can be included.

The Modest Toolset is implemented in C#. This allows the same binary distribution to run on 32- and 64-bit Windows, Mac OS and Linux machines. Libraries with a C interface are easy to use from C#. modes uses the runtime bytecode generation facilities in the standard `Reflection.Emit` namespace to generate fast simulation code for the specific model at hand.

6 Conclusion

We have presented the Modest Toolset, version 2.0, highlighting how it facilitates reuse of modelling expertise via Modest, a high-level compositional modelling language, while allowing reuse of existing models by providing import and export facilities for existing languages; and how it permits reuse of existing tools by integrating them in a unified modelling and analysis environment.

The toolset and the Modest language have been used on several case studies, most notably to analyse safety properties of a wireless bicycle brake [2] and to evaluate stability, availability and fairness characteristics of power micro-generation control algorithms [15]. For a more extensive list of case studies, we refer the interested reader to [13].

The Modest Toolset, including example models, is available for download on its website, which also provides documentation, a list of relevant publications and the description of several case studies, at `www.modestchecker.net`.

Planned improvements and extensions include distributed simulation and graphical automata modelling. We are very open for collaborations on case studies, new input languages and connecting to more analysis backends.

References

1. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.P.: Performance evaluation and model checking join forces. Commun. ACM 53(9), 76–85 (2010)
2. Baró Graf, H., Hermanns, H., Kulshrestha, J., Peter, J., Vahldiek, A., Vasudevan, A.: A verified wireless safety critical hard real-time design. In: WoWMoM. IEEE (2011)
3. Behrmann, G., David, A., Larsen, K.G.: A tutorial on UPPAAL. In: Bernardo, M., Corradini, F. (eds.) SFM-RT 2004. LNCS, vol. 3185, pp. 200–236. Springer, Heidelberg (2004)
4. Bogdoll, J., David, A., Hartmanns, A., Hermanns, H.: mctau: Bridging the gap between modest and UPPAAL. In: Donaldson, A., Parker, D. (eds.) SPIN 2012. LNCS, vol. 7385, pp. 227–233. Springer, Heidelberg (2012)
5. Bogdoll, J., Ferrer Fioriti, L.M., Hartmanns, A., Hermanns, H.: Partial order methods for statistical model checking and simulation. In: Bruni, R., Dingel, J. (eds.) FMOODS/FORTE 2011. LNCS, vol. 6722, pp. 59–74. Springer, Heidelberg (2011)
6. Bogdoll, J., Hartmanns, A., Hermanns, H.: Simulation and statistical model checking for Modestly nondeterministic models. In: Schmitt, J.B. (ed.) MMB/DFT 2012. LNCS, vol. 7201, pp. 249–252. Springer, Heidelberg (2012)
7. Bohnenkamp, H.C., D'Argenio, P.R., Hermanns, H., Katoen, J.P.: MoDeST: A compositional modeling formalism for hard and softly timed systems. IEEE Trans. Software Eng. 32(10), 812–830 (2006)
8. Bohnenkamp, H.C., Hermanns, H., Katoen, J.-P.: MOTOR: The MODEST Tool Environment. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 500–504. Springer, Heidelberg (2007)
9. Courtney, T., Gaonkar, S., Keefe, K., Rozier, E., Sanders, W.H.: Möbius 2.3: An extensible tool for dependability, security, and performance evaluation of large and complex system models. In: DSN, pp. 353–358. IEEE (2009)
10. Frehse, G.: PHAVer: Algorithmic verification of hybrid systems past HyTech. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 258–273. Springer, Heidelberg (2005)
11. Garavel, H., Lang, F., Mateescu, R., Serwe, W.: Cadp 2011: a toolbox for the construction and analysis of distributed processes. STTT 15(2), 89–107 (2013)
12. Hahn, E.M., Hartmanns, A., Hermanns, H., Katoen, J.P.: A compositional modelling and analysis framework for stochastic hybrid systems. Formal Methods in System Design 43(2), 191–232 (2013)
13. Hartmanns, A.: Modest - a unified language for quantitative models. In: FDL, pp. 44–51. IEEE (2012)
14. Hartmanns, A., Hermanns, H.: A Modest approach to checking probabilistic timed automata. In: QEST, pp. 187–196. IEEE Computer Society (2009)
15. Hartmanns, A., Hermanns, H., Berrang, P.: A comparative analysis of decentralized power grid stabilization strategies. In: Winter Simulation Conference (2012)
16. Hartmanns, A., Timmer, M.: On-the-fly confluence detection for statistical model checking. In: Brat, G., Rungta, N., Venet, A. (eds.) NFM 2013. LNCS, vol. 7871, pp. 337–351. Springer, Heidelberg (2013)
17. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011)