POWVER

Technical Report 2017-08

Title: **From Lotosphere to Thermosphere**

Author: Holger Hermanns

Report Number: 2017-08

ERC Project: Power to the People. Verified.

ERC Project ID: 695614

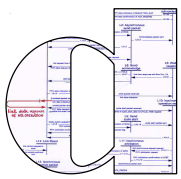
Funded Under: H2020-EU.1.1. – EXCELLENT SCIENCE

Host Institution: Universität des Saarlandes, Dependable Systems and Software
Saarland Informatics Campus

Published In: ModelEd, TestEd, TrustEd 2017

Please cite this publication as follows:

Holger Hermanns.
From Lotosphere to Thermosphere.
ModelEd, TestEd, TrustEd - Essays Dedicated to Ed Brinksma on the Occasion of His 60th Birthday.
Lecture Notes in Computer Science 10500, Springer 2017, ISBN 978-3-319-68269-3. 357–367.



POWER TO THE PEOPLE.
VERIFIED.
erc

From Lotosphere to Thermosphere

Holger Hermanns

Saarland University – Computer Science, Saarland Informatics Campus, Saarbrücken, Germany

Abstract. This paper reflects on the influential nature of some of the many scientific achievements linked to Ed Brinksma on the occasion of his 60th birthday. We in particular discuss pioneering contributions in the contexts of constraint-oriented specification, model-based testing, and cost-optimal timed reachability, as well as with respect to tools and algorithms for the construction and analysis of systems. We shed light on these achievements by linking a historical perspective with recent and very applied research directly rooted in these contributions.

1 Introduction

The scientific œuvre of Ed Brinksma has many facets. We here focus on four of them, since we consider those to be characteristic conerstones of his work and because we do feel they have notable impact on the world we live in. We discuss how the pioneering work of Ed Brinksma on (i) model-based testing, on (ii) constraint-oriented specification, and on (iii) cost-optimal reachability analysis is having impact on today's scientific forefront. We conclude by putting them into the greater context of his dedication to (iv) tools and algorithms for the construction and analysis of systems. The selection of facets considered naturally has a personal bias.

2 Model-based Testing

This section reviews how model-based testing has made its way from the university labs in Twente to customer appliances that assist in everyday life.

Testing Theory. Formal theories for testing were pioneered by Rocco De Nicola and Matthew Hennessy [12], originally motivated by the desire to characterise interesting formalisations of the notion of observable behaviour for transition systems, using an idealised but intuitive formalisation of testing. The first attempts to use this theory for automatic test derivation from formal specifications were made by Ed Brinksma in [7], and further developed in Twente jointly with Kars, Tretmans and coworkers [25]. This work was the nucleus for what is nowadays known as model-based testing, a technique with manifold and very practical applications.

Input-Output Conformance. Using a formal model as the specification of desired behaviour, *model-based testing* provides means to generate and carry out a suitable set of experiments on the implementation under test (IUT). This is done in an automated manner, with the goal to assert some notion of conformance of the IUT with respect to the model. The most prominent conformance relation in use is *input-output conformance* [24], developed by Jan Tretmans under the guidance of Ed Brinksma. It is defined for systems interacting *synchronously* with their environment, and especially with the model-based testing tool. The models are represented as input-output transitions systems (IOTS). In IOTS the transitions between states have a certain structure: each of them carries a name of an action ocurring and an identifier whether it is to be interpreted as an input (stimulus) to the implementation or an output (response) of the implementation (or an internal step).

Test Case Execution. A model-based testing tool performs automated inspection of the possible inputs and outputs while stepping through the states of a model. In each state of such a test case it either provides an input to or records an output from the IUT and accordingly updates its knowledge of what the current state of the model is. The test cases are executed on the actual software, system or device to be tested by translating the abstract transitions to concrete interactions with the IUT. Such a concrete execution of a test case (or of several test cases) ends in a test verdict of the form "pass", respectively "fail". Specifically, whenever an unexpected output of the IUT occurs, i.e. an output which is not foreseen by the current knowledge of the model state, the IUT is refuted with the verdict "fail".

Embedded Energy Managment Software. Embedded control software has become a major driver of industrial innovation, encompassing many critical, and sometimes safety-critical, application domains. A particularly delicate domain is the management of electric power: Embedded power management software has been traced to be the root of unintended and partly dangerous malfunctionings of laptops [27], smart phones [28], smart watches [29], pacemakers [30], and light electric vehicles [31]. The proper handling of electric power by software is obviously intricate. At the same time, electric power is the base commodity needed to innovate formerly all-mechanical systems.

EnergyBus. The EnergyBus is an emerging industrial standard for electric power transmission and management tailored to light electric mobility. At its core is an open specification for interoperability of the electrical components of e-bikes and other light electric vehicles, encompassing batteries, chargers, motors, sensors, and the human interface. The specification is based on the CANopen field bus. Its development is driven by EnergyBus e.V., an association formed by major industrial stakeholders in the e-bike domain. The EnergyBus specification itself is the nucleus for the joint IEC/ISO standardisation IEC/IS/TC69/JPT61851-3 and European Norm (EN) 50604, aiming at eventually enabling a single charger to be used across all light electric vehicles. By

mid 2018, this safety standard is scheduled to become a binding standard in Europe, thereby enabling effective public charging infrastructures for light electric mobility.

Applying Model-based Testing. State-of-the-art formal methods and tools have been and are being applied in the EnergyBus context to assure the general correctness and safety of EnergyBus protocol specifications [16], as well as to support implementers of EnergyBus in designing correct and safe implementations. For the latter, we have lately developed a tool platform for automated conformance testing of EnergyBus implementations against their formal specification. The tool platform is based on the Modest modelling language [17] and its accompanying Modest Toolset [18], which we extended with support for effective model-based testing against the EnergyBus protocol specification.

Asynchrony in Conformance Testing. The EnergyBus testing process itself motivated us to extend the supported conformance relation to asynchronous testing, especially in order to eliminate spurious errors. This is because the EnergyBus protocol uses CAN-based communication primitives. This setting however violates the synchrony hypothesis, just as many other settings do. In order to nevertheless provide testing facilities we were required to come up with a new and effective approach to model-based testing under asynchrony. By waiving the need to guess the possible output state of the IUT, we indeed manage to reduce the computational effort of the test generation algorithm while preserving soundness and conceptual completeness of the testing procedures. In addition, no restrictions on the specification model need to be imposed [14].

Industrial Uptake and Integration. In order to foster both, the application of formal methods in industry, as well as the quality and interoperability of EnergyBus devices reaching the consumer market, we have made our testing platform available to all industrial members of EnergyBus e.V. free-of-charge. This means that EnergyBus members can freely operate with the tool, so as to test conformance of their implementations directly against the formal specification of the EnergyBus protocol [15]. At the same time, we are ourselves performing tests of prototype devices, as soon as they are made available to us by members of the association. Our contributions in the context of the EnergyBus standardization efforts support the entire process from specification, modelling, verification and certification including both traditional test case programming and model-based testing. Specification inaccuracies as well as programming bugs have been found in tested prototype and retail devices. Based on our insights, documentation and implementations have been improved. We are not aware of any other standardization procedure with a similarly tight integration of formal methods.

3 Constraint-oriented Specification

This section discusses the constraint-oriented specification style, originally coined by Ed Brinksma, in the light of constraints on real-time behaviour. What may look like a surprising angle at first sight, is actually a natural and useful extension.

Behavioural Constraints by Composition. In the late 80ies of the last century, Ed Brinksma introduced *constraint-oriented specification* [8]. This specification style harvests features of multiway parallel composition operators as they are found in process languages such as LOTOS or CSP. Indeed, these operators can "implement" the power of logical conjunction with respect to sets of traces. The constraint-oriented specification style has shown its merits as an extremely useful tool in realistic applications, where it is used to carry out successive steps of logical refinement in specifications [13].

Timing by Composition. One particular manifestation of its usefulness is its adoption to the time domain, in the form of *time constraints*. Together with multiway synchronisation, time constraints can gradually turn a untimed specification into one where certain occurences of actions are to be delayed. These constraints are added by composition. In the context of timed automata, this idea is implicitly present for instance in some of the modelling work related to the Bang & Olufsen audio/video power control protocol [19]. A full proposal has been developed in [21] in the context of interactive Markov chains [20]. There, it has been used to turn an untimed specification of a plain-old telephone system protocol into a timed specification, solely by the use of composition with time constraints. We here sketch the essence of time constraints recast into the setting of *timed automata* [1].

Timed Automata. Timed automata are a standard modelling formalism for real time systems. A timed automaton is an extension of finite state machines with non-negative real valued variables called *clocks* in order to capture timing constraints. Thus, a timed automaton is an annotated directed graph over a set of clocks C where vertices (called *locations*) are connected by edges, and both are decorated with conjunctions of clock constraints of the form $c \leq k$ or $c \geq k$ with c being a clock and $k \in \mathbb{N}$. For edges such constraints are called *guards*, for locations they are called *invariants*. Edges are additionally decorated with *reset sets* of clocks. Intuitively, taking an edge causes an instantaneous change of location and a reset to 0 for each clock in the reset set. However an edge may only be taken if its guard and the target location's invariant evaluate to true. It does not have to be taken however. As long as permitted by the invariant of the current location, time can advance there, meaning that all clocks increase continuously with their assigned rates, thus modelling the passing of time. Figure 1 depicts a small example of a timed automaton.

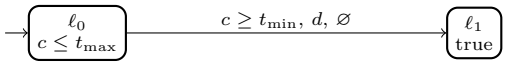


Fig. 1. A simple timed automaton *TA*. The invariant $c \leq t_{\max}$ in location ℓ_0 and the guard $c \geq t_{\min}$ on the edge together impose a nondeterministic delay of at least $t_{\min}$ and at most $t_{\max}$ before action d may occur. No clocks are reset, due to the reset set being $\varnothing$.

Time Constraints. Now assume we are given a (possibly entirely untimed) system, which encompasses (not necessarily disjoint) sets of actions S, D, and I. Furthermore assume that we want to ensure a delay of some duration for occurences of actions in D (to be delayed) after occurrence of any action in S (starting the delay), unless an action of I (interrupting the delay) occurs in the meanwhile. The delay we want to ensure has a duration of at least $t_{\min}$ and at most $t_{\max}$. So, we concretely assume

- an interval $[t_{\min}, t_{\max}] \subset \mathbb{R}^{\geq 0}$ of real time that determines the possible duration of the time constraint,
- a set of actions S (start) that determines when a delay starts,
- a set of actions D (delay) that are to be delayed, and
- a set of actions I (interrupt) each of which may interrupt the delay.

Based on this information, a simple two-location timed automaton needs to be constructed which operates with a single fresh clock c. The locations are ℓ_1 and ℓ_0. The invariant of ℓ_1 is true, the one of ℓ_0 is $c \leq t_{\max}$ (as already seen in Figure 1). Location ℓ_1 serves as initial location. Furthermore,

- for each $s \in S$ we have $\ell_1 \xrightarrow{\text{true},s,\{c\}} \ell_0$ and $\ell_0 \xrightarrow{\text{true},s,\{c\}} \ell_0$;
- for each $d \in D$ we have $\ell_1 \xrightarrow{\text{true},d,\emptyset} \ell_1$ and $\ell_0 \xrightarrow{c \geq t_{\min},d,\emptyset} \ell_1$;
- for each $i \in I$ we have $\ell_1 \xrightarrow{\text{true},i,\emptyset} \ell_1$ and $\ell_0 \xrightarrow{\text{true},i,\emptyset} \ell_1$.

For singleton sets S, D, and I the result of the above construction is sketched in Figure 2. The construction needs slight adjustments if the three sets are not disjoint [20, 5.5].[1] The main functionality of the above construction is that it does insert a delay for actions of D, but otherwise does not interfere with actions of $S \cup D \cup I$.

Incorporating Time Constraints. It is precisely the constraint-oriented specification style originally proposed by Ed Brinksma [8] that enables us to incorporate such a constraint TC into a system SY by composition. All that is needed is a multiway parallel composition operator $||_A$ which synchronizes precisely the

[1] Notably, choosing the reset set to include c in $\ell_0 \xrightarrow{\text{true},s,\{c\}} \ell_0$ makes the delay restart should another action s occur while the delay is running. Another option would be to drop c from this set. This might be preferable dependent on the context.

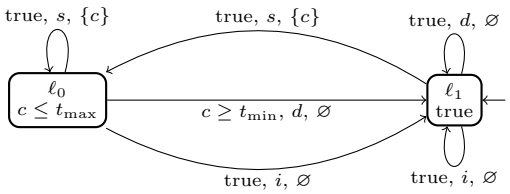


Fig. 2. A time constraint TC for $S = \{s\}$, $I = \{i\}$, and $D = \{d\}$ extending the timed automaton TA from Figure 1. The delay on action d is started upon occurrence of action s and can be interrupted by action i.

actions in A and otherwise lets actions proceed independently [9,6]. With this operator the time-constrained system is expressed as

$$SY \;||_{S \cup D \cup I}\; TC.$$

This system behaves just as SY behaves, except that whenever an action from S occurs in SY, all actions from D in SY are assured to be delayed at least by an amount of time that lies in the interval $[t_{\min}, t_{\max}]$ unless an action from I occurs in SY in the meanwhile. Further time constraints can be added to the system in the very same manner, as in

$$(\; \cdots (\; (\; SY \;||_{S_1 \cup D_1 \cup I_1}\; TC_1) \;||_{S_2 \cup D_2 \cup I_2}\; TC_2) \;\cdots ||_{S_n \cup D_n \cup I_n}\; TC_n).$$

Analysis. Overall, this approach can turn an untimed specification into a timed specification in a compositional manner. This makes the final system amenable to quantitative analysis, including real-time model checking and the like. A complete case study in this regard has been carried out in a setting with soft real-time [21]. It can also be combined with induction and data independence [11].

4 Cost-optimal Timed Reachability

This section elaborates on the concepts of cost-optimal scheduling, originally co-developed by Ed Brinksma, and how they are finding their way into tiny objects orbiting the earth.

Priced Timed Automata. In order to reason about resource consumption, Ed Brinksma and his collaborators have enriched timed automata with non-negative integer *costs* and non-negative *cost rates* in the form of annotations for edges and locations respectively [22]. The result are *priced timed automata* (PTA). The intuition is that cost accumulates continuously in a proportional manner to the sojourn time of locations and increases in a step upon taking an edge as specified by the respective annotations.

Cost-Optimal Reachability. The original problem considered in the context of PTA is that of computing the minimum cost to reach a certain target location in a given PTA. This so-called *cost-optimal reachability analysis* (CORA) has received dedicated attention and is implemented in a number of tools, most prominently Uppaal Cora [26]. As input Uppaal Cora accepts networks of PTAs extended by discrete variables, and thus allows for modular formalisation of individual components. The set of goal states is characterised by formulae over the variables in the network of PTAs.

Schedule Synthesis. One of the most prominent applications of this technique, explored in particular within the EU-funded AMETIST project, is schedule synthesis. The main strength of this approach is that the expressiveness of timed automata allows - unlike many classical approaches - the modelling of scheduling problems of very different kinds. Furthermore, the models are robust against changes in the parameter setting and against changes in the problem specification. A milestone in practical applicabilitly of this technique is a case study originally provided by AXXOM: an intricate scheduling problem for lacquer production [2]. A number of problems needed to be addressed for the modelling task, including information transfer from the industrial partner, the derivation of a timed automaton model for the case study, and the heuristics that have to be added in order to reduce the search space.

Robustness of Schedules. This analysis had to ignore two dimensions of the original problem specification as provided by AXXOM. These relate to quantitative stochastic influences due to failures, repairs, cleaning periods and other unforeseeable (and thus unplannable) events. To attack thesem the timed automata model of the production units has been refined into a stochastic timed automata model [4] in order to faithfully represent the stochastic perturbations and to assess the robustness of the system in light of these perturbations. The robustness of the schedules is assessed on the basis of estimates obtained by a discrete-event simulation-based analysis [5,23]. This two-step analysis approach, which combines timed automata-based verification with stochastic robustness analysis is a very striking and effective way to exploit the benefits of formal verification.

Scheduling in Thermosphere. Lately, we have applied this very same approach to a very challenging domain, the domain of *low-earth orbiting satellites*. This work was coined as part of the EU-funded SENSATION project, and continues as part of the ERC Advanced grant POWVER. For a satellite in low orbit all resources are sparse and the most critical resource of all is power. It is therefore crucial to have detailed knowledge on how much power is available for an energy harvesting satellite in orbit at every time – especially when in eclipse, where it draws its power from onboard batteries.

GomX–3 *Mission.* The GomX–3 CubeSat was a 3 liter ($30 \times 10 \times 10$cm, 3kg) nanosatellite designed, delivered, and operated by Danish sector leader

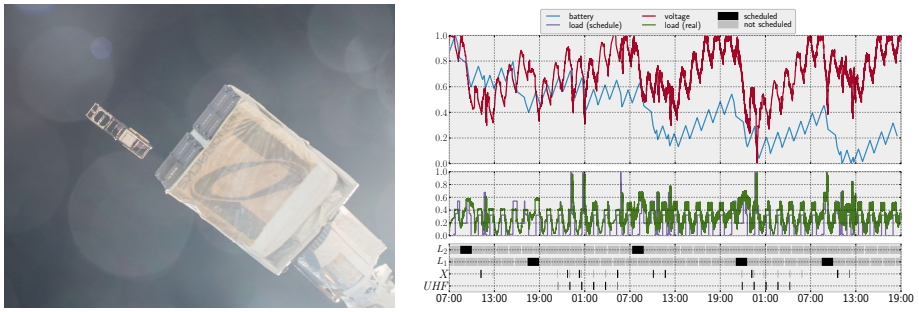


Fig. 3. The GomX–3 nanosatellite deployment from the ISS (left, picture taken by Astronaut Scott Kelly), and schedule effectuated March 20, 2016 7 AM to March 22, 2016 7 PM (right).

GomSpace. GomX–3 was the first ever In-Orbit Demonstration (IOD) CubeSat commissioned by ESA. The GomX–3 system used Commercial-off-the-shelf (COTS) base subsystems to reduce cost, enabling fast delivery so as to focus on payload development and testing. GomX–3 was launched from Japan aboard the HTV–5 on August 19, 2015. It successfully berthed to the ISS a few days later. GomX–3 was deployed from the ISS into thermosphere on October 5, 2015, it deorbited in October 2016. Figure 3 (left) shows the satellite at the time of deployment from ISS.

In-Orbit Scheduling. The heterogeneous timing aspects and the experimental nature of this application domain pose great challenges, making it impossible to use traditional scheduling approaches for periodic tasks. Our approach harvests work on schedulability analysis with (priced) timed automata, and is distinguished by the following features: (i) The timed automata modelling is very flexible, adaptive to changing requirements, and particularly well-suited for discussion with space engineers, since easy-to-grasp; (ii) A dynamic approach to the use of cost decorations and constraints allows for a split scheduling approach optimising over intervals, at the (acceptable) price of potential sub-optimality of the resulting overall schedules; (iii) A linear battery model is employed while computing scheduling, but prior to shipping any computed schedule is subjected to a quantitative validation on the vastly more accurate stochastic kinetic battery model, and possibly rejected. This last aspect is very close in spirit to the robust scheduling approach [23] discussed above. The stochastic validation step however is not based on simulation, but instead is exact (or conservative) up to discretisation. The procedure has been in use for the automatic and resource-optimal day-ahead scheduling of GomX–3. One of the schedules computed by the approach, and effectuated in by GomX–3 is displayed in Figure 3 (right).

Results. The GomX–3 in-orbit experiments have demonstrated an indeed great fit between the technology developed and the needs of the LEO satellite sector.

The schedules generated are of unmatched quality: It became apparent that relative to a comparative manual scheduling approach, better quality schedules with respect to (i) number of experiments performed, (ii) avoidance of planning mistakes, (iii) scheduling workload, and (iv) battery depletion risk are provided. At the same time, the availability of scheduling tool support flexibilises the satellite design process considerably, since it allows the GomSpace engineers to obtain answers to what-if questions, in combination with their in-house tools. This helps shortening development times and thus time-to-orbit. In fact, GomSpace will launch a constellation consisting of two spacecrafts (GomX–4 A and B) soon and is actively pursuing several projects with much larger constellations. Deploying constellations of a large number of satellites (2 to 1000) brings a new level of complexity to the game. The need to operate a large number of satellites asks for a larger level of automation to be used than has previously been the case in the space industry. For larger constellations tools for optimization, automation and validation are not only a benefit, but an absolutely necessity for proper operations.

5 Conclusion

This paper has reviewed high-impact pioneering contributions of Ed Brinksma in the contexts of constraint-oriented specification, of model-based testing, and of cost-optimal timed reachability. These are mainfestations of a general theme overarching his scientific work, namely *software tools supporting the application of formal methods*. Before being promoted to Rector Magnificus at Universiteit Twente he for many years held the Chair for Formal Methods and Tools ("Formele Methoden en Gereedschappen"). During this period, he heavily invested in tool support for formal methods, including tools for formal testing, verification of soft- and hard-real time systems, algebraic specifications, and many more. And very many of his projects of national, European and international scale have had a distinguished focus on the advancements on the software support side, notably including LOTOSPHERE, SVC, VHS, ARTIST, AMETIST, and QUASIMODO. Together with Kim Larsen (co-founder of UPPAAL [3]), Bernhard Steffen, and Rance Cleaveland (co-founders of the Concurrency Workbench [10]) he founded an international conference on tools and algorithms for the construction and analysis of systems (TACAS). This conference is nowadays simply *the* conference on tools and algorithms for the construction and analysis of systems.

Acknowledgments. We gratefully acknowledge insightful comments by Sadie Creese (University of Oxford) on an early draft of this paper. This work is supported by the ERC Advanced Investigators Grant 695614 (POWVER), and has profited from the EU-funded projects SENSATION, QUASIMODO, and AMETIST.

References

1. Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
2. Gerd Behrmann, Ed Brinksma, Martijn Hendriks, and Angelika Mader. Production scheduling by reachability analysis - A case study. In *19th International Parallel and Distributed Processing Symposium (IPDPS 2005), CD-ROM / Abstracts Proceedings, 4-8 April 2005, Denver, CO, USA*. IEEE Computer Society, 2005.
3. Johan Bengtsson, Kim Guldstrand Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. UPPAAL - a tool suite for automatic verification of real-time systems. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems III: Verification and Control, Proceedings of the DIMACS/SYCON Workshop, October 22-25, 1995, Ruttgers University, New Brunswick, NJ, USA*, volume 1066 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 1995.
4. Henrik C. Bohnenkamp, Pedro R. D'Argenio, Holger Hermanns, and Joost-Pieter Katoen. MODEST: A compositional modeling formalism for hard and softly timed systems. *IEEE Trans. Software Eng.*, 32(10):812–830, 2006.
5. Henrik C. Bohnenkamp, Holger Hermanns, Ric Klaren, Angelika Mader, and Yaroslav S. Usenko. Synthesis and stochastic assessment of schedules for lacquer production. In *1st International Conference on Quantitative Evaluation of Systems (QEST 2004), 27-30 September 2004, Enschede, The Netherlands*, pages 28–37. IEEE Computer Society, 2004.
6. Tommaso Bolognesi and Ed Brinksma. Introduction to the ISO specification language LOTOS. *Computer Networks*, 14:25–59, 1987.
7. Ed Brinksma. A theory for the derivation of tests. In Sudhir Aggarwal and Krishna K. Sabnani, editors, *Protocol Specification, Testing and Verification V, Proceedings of the IFIP WG6.1 Eighth International Conference on Protocol Specification, Testing and Verification, 1988*, pages 171–194. North-Holland, 1988.
8. Ed Brinksma. Constraint-oriented specification in a constructive formal description technique. In J. W. de Bakker, Willem P. de Roever, and Grzegorz Rozenberg, editors, *Stepwise Refinement of Distributed Systems, Models, Formalisms, Correctness, REX Workshop, Mook, The Netherlands, May 29 - June 2, 1989, Proceedings*, volume 430 of *Lecture Notes in Computer Science*, pages 130–152. Springer, 1989.
9. Stephen D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *J. ACM*, 31(3):560–599, 1984.
10. Rance Cleaveland, Joachim Parrow, and Bernhard Steffen. The Concurrency Workbench: A semantics-based tool for the verification of concurrent systems. *ACM Trans. Program. Lang. Syst.*, 15(1):36–72, 1993.
11. S. J. Creese and A. W. Roscoe. Verifying an infinite family of inductions simultaneously using data independence and FDR. In Jianping Wu, Samuel T. Chanson, and Qiang Gao, editors, *Formal Methods for Protocol Engineering and Distributed Systems, FORTE XII / PSTV XIX'99, IFIP TC6 WG6.1 Joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE XII) and Protocol Specification, Testing and Verification (PSTV XIX), October 5-8, 1999, Beijing, China*, volume 156 of *IFIP Conference Proceedings*, pages 437–452. Kluwer, 1999.
12. Rocco De Nicola and Matthew Hennessy. Testing equivalences for processes. *Theor. Comput. Sci.*, 34:83–133, 1984.
13. Hubert Garavel and Wendelin Serwe. The unheralded value of the multiway rendezvous: Illustration with the production cell benchmark. In Holger Hermanns and

Peter Höfner, editors, *Proceedings 2nd Workshop on Models for Formal Analysis of Real Systems, MARS@ETAPS 2017, Uppsala, Sweden, 29th April 2017.*, volume 244 of *EPTCS*, pages 230–270, 2017.

14. Alexander Graf-Brill, Arnd Hartmanns, Holger Hermanns, and Steffen Rose. Model-based testing for asynchronous systems. In *Critical Systems: Formal Methods and Automated Verification - Joint 22nd International Workshop on Formal Methods for Industrial Critical Systems and 17th International Workshop on Automated Verification of Critical Systems, FMICS-AVoCS 2017, Torino, Italy, September 18-20, 2017, Proceedings.*, Lecture Notes in Computer Science. Springer, 2017.
15. Alexander Graf-Brill, Arnd Hartmanns, Holger Hermanns, and Steffen Rose. Modelling and certification for electric mobility. In *15th IEEE International Conference on Industrial Informatics, INDIN 2017, Emden, Germany, July 24-26, 2017*. IEEE, 2017.
16. Alexander Graf-Brill, Holger Hermanns, and Hubert Garavel. A model-based certification framework for the energybus standard. In Erika Ábrahám and Catuscia Palamidessi, editors, *Formal Techniques for Distributed Objects, Components, and Systems - 34th IFIP WG 6.1 International Conference, FORTE 2014, Held as Part of the 9th International Federated Conference on Distributed Computing Techniques, DisCoTec 2014, Berlin, Germany, June 3-5, 2014. Proceedings*, volume 8461 of *Lecture Notes in Computer Science*, pages 84–99. Springer, 2014.
17. Ernst Moritz Hahn, Arnd Hartmanns, Holger Hermanns, and Joost-Pieter Katoen. A compositional modelling and analysis framework for stochastic hybrid systems. *Formal Methods in System Design*, 43(2):191–232, 2013.
18. Arnd Hartmanns and Holger Hermanns. The modest toolset: An integrated environment for quantitative modelling and verification. In Erika Ábrahám and Klaus Havelund, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings*, volume 8413 of *Lecture Notes in Computer Science*, pages 593–598. Springer, 2014.
19. Klaus Havelund, Kim Guldstrand Larsen, and Arne Skou. Formal verification of a power controller using the real-time model checker UPPAAL. In Joost-Pieter Katoen, editor, *Formal Methods for Real-Time and Probabilistic Systems, 5th International AMAST Workshop, ARTS'99, Bamberg, Germany, May 26-28, 1999. Proceedings*, volume 1601 of *Lecture Notes in Computer Science*, pages 277–298. Springer, 1999.
20. Holger Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *Lecture Notes in Computer Science*. Springer, 2002.
21. Holger Hermanns and Joost-Pieter Katoen. Automated compositional Markov chain generation for a plain-old telephone system. *Sci. Comput. Program.*, 36(1):97–127, 2000.
22. Kim Guldstrand Larsen, Gerd Behrmann, Ed Brinksma, Ansgar Fehnker, Thomas Hune, Paul Pettersson, and Judi Romijn. As cheap as possible: Efficient cost-optimal reachability for priced timed automata. In Gérard Berry, Hubert Comon, and Alain Finkel, editors, *Computer Aided Verification, 13th International Conference, CAV 2001, Paris, France, July 18-22, 2001, Proceedings*, volume 2102 of *Lecture Notes in Computer Science*, pages 493–505. Springer, 2001.

23. Angelika Mader, Henrik C. Bohnenkamp, Yaroslav S. Usenko, David N. Jansen, Johann Hurink, and Holger Hermanns. Synthesis and stochastic assessment of cost-optimal schedules. *STTT*, 12(5):305–318, 2010.
24. Jan Tretmans. Model based testing with labelled transition systems. In Robert M. Hierons, Jonathan P. Bowen, and Mark Harman, editors, *Formal Methods and Testing, An Outcome of the FORTEST Network, Revised Selected Papers*, volume 4949 of *Lecture Notes in Computer Science*, pages 1–38. Springer, 2008.
25. Jan Tretmans, Pim Kars, and Ed Brinksma. Protocol conformance testing: A formal perspective on ISO IS-9646. In Jan Kroon, Rudolf Jan Heijink, and Ed Brinksma, editors, *Protocol Test Systems, IV, Proceedings of the IFIP TC6/WG6.1 Fourth International Workshop on Protocol Test Systems, Leidschendam, The Netherlands, 15-17 October, 1991*, volume C-3 of *IFIP Transactions*, pages 131–142. North-Holland, 1991.
26. Uppaal Cora. `http://people.cs.aau.dk/~adavid/cora/introduction.html`, 2005. Online; accessed: 2017-07-31.
27. Catastrophic Surface Pro 3 battery life finally has its firmware fix. `http://arstechnica.com/?p=945575`, 2016. Online; last access 2017-07-31.
28. Samsung recalls Galaxy Note 7 worldwide due to exploding battery fears. `http://theverge.com/2016/9/2/12767670`, 2016. Online; last access 2017-07-31.
29. Basis Peak watches recalled. `http://techcrunch.com/2016/08/03/basis-peak-watches-recalled-due-to-overheating/`, 2016. Online; last access 2017-07-31.
30. Important: Medical device correction, EnRhythm pacemakers. `http://www.medtronic.com/enrhythm-advisory/downloads/enrhythm-battery-issues_physician-letter.pdf`, 2010. Online; last access 2017-07-31.
31. Qualitätsprobleme bei E-Bikes: Schlappe Akkus, anfällige Elektronik. `http://www.spiegel.de/auto/aktuell/a-790142.html`, 2011. Online; last access 2017-07-31.