Software Modeling and Verification

Staff

- Professors:

  Prof. Dr. Ir. Joost–Pieter Katoen PD
  Prof. em. Dr. Klaus Indermark
  `http://moves.rwth-aachen.de/`

- Secretary:

  Elke Ohlenforst

- Lecturer:

  Akademischer Oberrat Priv.–Doz. Dr. Thomas Noll

- Researchers:

  Dr. Erika Ábrahám (since April, from Univ. Freiburg)
  Dr. Henrik Bohnenkamp
  Tingting Han, M.Sc. (funded by the NWO)
  Dipl.-Inform. Carsten Kern
  Dipl.-Inform. Martin Neuhäußer (funded by the NWO)
  Dipl.-Inform. Stefan Rieger
  Dipl.-Inform. Volker Stolz (until February)
  Dipl.-Inform. Daniel Willems (since June, funded by the DFG)
  Ivan Zapreev, M.Sc. (funded by the NWO)

- Technical Staff:

  Arnd Gehrmann

- Diploma/Master Students:

  Berteun Damman
  Lars Helge Haß
  Tim Kemna
  Alexandru Mereacre
  Marcel Oldenkamp

- Student Researchers:

  Frank Birbacher
  Jonathan Heinen
  Thomas Kesselheim
  Christian Lücking
  Daniel Neider
  Denise Nimmerrichter
  Roman Rabinovich
  Michael Rohrbach
  Andreas Röll
  Franziska Roloff
  Ulrich Schremp
  Stefan Schulz
  Henning Stein
  Evamarie Storch
  Benedikt Westermann

- Visiting Scientists:

  Dr. Benedikt Bollig (ENS Cachan, F)
  Prof. Dr. Luboš Brim (Masaryk University, Brno/CZ)
  Dr. Dino Distefano (Queen Mary Univ. of London, GB)
  Prof. Dr. Bart Jacobs (Radboud University, Nijmegen/NL)
  Dr. Martin Leucker (TU Munich, D)
  Dr. Ir. Arend Rensink (University of Twente, NL)
  Dr. Mariëlle Stoelinga (University of Twente, NL)
  Nikola Trcka (TU Eindhoven, NL)
  Prof. Dr. Heike Wehrheim (Universität Paderborn, D)

Overview

The research programme of the *Software Modeling and Verification group (MOVES)* is concerned with the study, development and application of *formal methods* to software design in a broad sense. Our group aims at modeling and verifying *thrustworthiness aspects* (such as safety, reliability, performance and survivability) of software systems by applying mathematical theories and methods.

Major research topics of interest are:

- modeling formalisms for concurrent systems (such as process algebras, statecharts, message sequence diagrams and mobile process calculi);
- model checking and quantitative extensions thereof (in particular probabilistic model checking, cost bounds, abstractions, scheduling generation and analysis);
- semantics and analysis of modern programming languages (a.o., semantics of Erlang, heap abstractions and pointer analysis, multi-threading);
- probabilistic models for concurrency (i.e., the theory of models, abstraction, refinement, continuous-time stochastic models that exhibit nondeterminism etc.);
- testing and run-time verification with a focus on real-time issues.

Our research is conducted in the context of several projects that are funded by the NWO (Dutch Research Council), NWO and DFG, and the European Union. We participate in the Research Training Group on the Algorithmic Synthesis of Reactive Systems (ALGOSYN) and the UMIC Excellence Cluster.

In 2006, Volker Stolz obtained his PhD with a dissertation on run-time verification and received a position at the International Institute for Software Technology at the United Nations University in Macau. Erika Abráhám (University of Freiburg) joined our group in May 2006. She is working on the interregional SFB AVACS. Daniel Willems started his PhD work within the Research Training Group ALGOSYN. Welcome to the MOVES group!

Joost-Pieter Katoen

Research Projects

Bounded Model Checking of Hybrid Systems

E. Ábrahám et al. (Albert-Ludwigs-Univ. Freiburg, Carl-von-Ossietzky Univ. Oldenburg, Univ. des Saarlandes)

In the context of the DfG Transregio AVACS project WP H1/2 we are investigating bounded model checking of hybrid systems. Given some system together with a specification expressing requirements on the system, the approach formulates the existence of a counterexample of a fixed length by a formula, and uses satisfiability checkers over different domains to check the formula for satisfiability.

We are interested in developing new techniques to accelarate the existing algorithms and to extend the domain that can be handled.

Observability and Fully Abstract Semantics for Class-Based Languages

E. Ábrahám, M. Steffen (Univ. of Oslo, NOR), A. Grüner (Christian-Albrechts-Univ. Kiel, D)

We are interested to define the externally observable behaviour of class-based object-oriented languages, including different language features like exceptions, sequentiality, concurrency, monitors, etc.

The semantics gives insight into the inherent language features, and allows to decide whether two components are observably equivalent. This is important in many practical fields, for example for compiler optimizations. Furthermore, fully abstract semantics are a good basis for the development of compositional proof methods.

The MoDeST Tool Environment

H. Bohnenkamp, J.-P. Katoen, H. Hermanns (Univ. d. Saarlandes, D), P. R. D'Argenio (Univ. Nacional de Córdoba, AR)

The specification language MODEST covers a wide spectrum of modelling concepts, ranging from plain labelled transition systems to stochastic systems like Generalised Semi-Markov Decision Processes. MODEST possesses a rigid, process-algebra style

semantics, and yet provides modern and flexible specification constructs. MODEST specifications constitute a coherent starting-point to analyse distinct system characteristics with various techniques, e.g., model checking to assess functional correctness and discrete-event simulation to establish the system's reliability. Analysis results thus refer to the *same* system specification, rather than to different (and potentially incompatible) specifications of system perspectives like in the UML.

The tool MOTOR (MODEST Tool enviRonment) is aimed to provide the means to analyse and evaluate MODEST specifications. The tool is written in the C++ programming language. The tool provides (i) interfacing capabilities for connection to existing tools for specific projected models, and (ii) also means for enhancement by *native* algorithms for analysis of (classes) of MODEST specifications. In earlier work, MOTOR has been connected to MÖBIUS, a performance evaluation tool suite that has been developed at the University of Illinois at Urbana-Champaign, US. Currently we are working on a state-space generator for MODEST and aim at connecting MODEST via MOTOR to the PRISM tool, a model-checker for probabilistic timed systems, developed at the University of Birmingham, UK.

Model-Based Testing

H. Bohnenkamp, A. Belinfante (Univ. Twente, NL), M. Stoelinga (Univ. Twente, NL)

Testing is one of the most natural, intuitive and effective methods to increase the reliability of software. Formal methods have been employed to analyse and systematise the testing idea in general, and to define notions of correctness of implementations with respect to specifications in particular. The IOCO testing theory reasons about black-box conformance testing of software components. The test-case generation and execution algorithms of IOCO have been implemented in TORX, a testing tool developed at the University of Twente.

We work on two topics in this area.

1. An extension to TORX to allow testing of real-time properties: *real-time testing*. Real-time testing means that the decisions whether an implementation under test has passed or failed a test is not only based on which outputs are observed, given a certain sequence of inputs, but also on *when* the outputs occur, given a certain sequence of inputs *applied at predefined times*. We use as input models non-deterministic *safety timed automata*.

2. In timed testing, reaching a verdict depends on time measurements. The imprecision of measurements can lead to false positives (a test fails although the implementation behaves correctly). We work on an extension of the ioco theory,

where the verdicts are not binary (pass/fail) but of a quantitiative nature: it is measured how close to the specified behaviour the implementation behaves.

Dependable Global Computing

J.-P. Katoen, R. De Nicola (U. Florence, I), D. Latella (CNR-ISTI, I), M. Loreti (U. Florence, I) and M. Massink (CNR-ISTI, I)

(funded by the DAAD and CNR-ISTI)

Due to their enormous size—networks typically consist of thousands or even millions of nodes—and their strong reliance on mobility and interaction, performance and dependability issues are of utmost importance for "network-aware computing". Spontaneous computer crashes may easily lead to failure of remote execution or process movement, while spurious network hick ups may cause loss of code fragments or unpredictable delays. The enormous magnitude of computing devices involved in global computing yields failure rates that no longer can be ignored. The presence of such random phenomena implies that correctness of global computing software and their privacy guarantees are no longer rigid notions like: "either it is safe or it is not" but have a less absolute nature, e.g.: "in 99.7% of the cases, privacy can be ensured". The intrinsic complexity of global computers, though, complicates the assessment of these issues severely. Systematic methods, techniques and tools—all based on solid mathematical foundations i.e., *formal methods*, are therefore needed to establish performance and dependability requirements and guarantees.

This project attempts to make a considerable step into this direction by extending a successful programming and specification formalism for global computing, KLAIM, with random delays, and by developing a novel stochastic spatial temporal logic as property specification language for performance and dependability guarantees.

Synthesis and Stochastic Assessment of Cost-Optimal Schedules

H. Bohnenkamp, A. Mader (Univ. Twente, NL), Y. Usenko (Univ. Eindhoven, NL), H. Hermanns (Univ. Saarbrücken, D), David Jansen (Univ. Twente, NL), Johann Hurink (Univ. Twente, NL).

It is well-known that model checkers can be used to solve scheduling problems. Schedules can be obtained by forcing the model checker to produce a counter-example for the claim "*There is no solution to the scheduling problem*" in the model at hand.

We apply schedule generation with model-checkers to a case study from the area of lacquer production, where lacquers are produced according to certain recipes, and production resources (vessels, mixers, dispersers and what not) are prone to failures. Failures cause the completion of a job to be late, which incurs costs (the later, the most expensive).

Although there are model checkers which can deal with costs (notably UPPAAL Cora), and optimise schedules to minimise cost, the stochastic nature of failures is something that can not be captured. We therefore combine schedule-generation with simulation in order to find good schedules:

1. We simulate each recipe to obtain the optimal time needed for it to incur minimal cost.
2. We incorporate this information into the model used for schedule generation.
3. We assess the quality of the generated schedules with simulation again.

The results show that this approach does indeed produce good schedules. Research is however still needed to refine this method.

QUPES: Verification of Quantitative Properties of Embedded Software

T. Han, J.-P. Katoen, M. Neuhäußer, D. Willems

(funded by the NWO)

The research challenge faced by the QUPES project is to adapt and enrich model checking, a successful technique for checking the logical correctness of system designs, to meet the requirements of state-of-the-art embedded software engineering. Embedded software typically executes on devices that, first and foremost, are not computers. This imposes high requirements on performance and economical resource usage. Due to its embedded nature, its robustness is of prime importance, and timely reactions to stimuli from its—mostly physical—environment are essential. This project proposes to assess these "non-functional" aspects (e.g., timeliness and robustness) as an integral part of the embedded software validation phase. The aim is to obtain a single framework that supports both the validation of qualitative (i.e., functional) as well as quantitative aspects of embedded software.

To accomplish this, model-checking techniques will be extended with ample means to reason about costs (power consumption, memory usage, and the like), efficiency, and robustness. In particular, we aim to develop verification algorithms for real-time

systems that exhibit both *(continuous-time) randomness as non-determinism* and extend this approach with *cost* aspects. Furthermore, advances to model checking of stochastic systems will be made by developing aggressive *abstraction* techniques and methods that effectively exploit the (compositional) structure of embedded software specifications for verification purposes. The latter activities are aimed at making stochastic model checking applicable to state spaces that are several orders of magnitude larger than currently can be handled. Our techniques will be tailored to hierarchical design notations (statechart diagrams) for embedded software.

This project takes place in the context of the DFG-NWO bilateral project VOSS2 (Validation of Stochastic Systems). In this project, we cooperate with the Universities of Bonn and Dresden (Prof. Christel Baier), Saarland (Prof. Holger Hermanns), Nijmegen (Prof. Frits Vaandrager), Twente (Prof. Boudewijn Haverkort), and Federal Armed Forces Munich (Prof. Markus Siegle).

Verifying Concurrent Pointer Programs with Unbounded Heap

J.-P. Katoen, Th. Noll, S. Rieger, D. Distefano (Queen Mary College London, UK), A. Rensink (U. Twente, NL)

The incorrect use of pointers is one of the most common source of software errors. Concurrency has a similar characteristic. Proving the correctness of concurrent pointer manipulating programs with unbounded heap, let alone algorithmically, is a highly non-trivial task. This project attempts to develop automated verification techniques and accompanying tool support for concurrent programs that manipulate dynamic, linked data structures. Initially, we focus on linked lists. As verification technique, we investigate the use of automata-based model-checking algorithms. First and second-order (monadic) temporal logics are employed for the specification of properties of such concurrent programs. These logics can easily express the dynamic creation and deletion of data. In our approach, we consider abstractions of linked data structures that are tailored to both the property and the program to be analysed. The main challenge is to achieve a fully automated technique to prove the correctness of concurrent pointer programs such as deadlock avoidance protocols, concurrent garbage collection algorithms and so forth.

MC=MC: Model Checking Infinite-State Markov Chains

J.-P. Katoen, I. Zapreev, B. Haverkort (Univ. Twente, NL), A. Remke (Univ. Twente, NL)

(funded by the NWO)

The MC=MC project focuses on model checking techniques for infinite-state systems, thereby combining previous work on model checking finite-state Markov chains and the evaluation of infinite-state stochastic models (e.g. as generated from infinite-state stochastic Petri nets).

As a part of this project we implement an experimental tool named MRMC (Markov Reward Model Checker). MRMC is a model checker for discrete-time and continuous-time Markov reward models. It supports reward extensions of PCTL and CSL (PRCTL and CSRL), and allows for the automated verification of properties concerning long-run and instantaneous rewards as well as cumulative rewards. In particular, it supports to check the reachability of a set of goal states (by only visiting legal states before) under a time and an accumulated reward constraint. Additionally MRMC provides (i) safe on-the-fly steady-state detection for time-bounded reachability, and (ii) bisimulation minimization for PCTL, CSL, PRCTL and CSRL logics, for the latter two in the case without impulse rewards.

MRMC is a command-line tool written in C-language. Which allows MRMC to be small and fast. We support LINUX platform only, but on WINDOWS it can be built and run under CYGWIN. The tool is distributed under the GNU general public license (GPL).

Our current research is concentrated on several topics: (i) the effect of bisimulation minimisation on probabilistic model checking, (ii) an empirical comparison of model checkers for probabilistic systems, and (iii) model checking CSL logic using simulation techniques. The idea behind the latter is to apply discrete event simulation instead of conventional techniques for model checking CTMCS. We employ regenerative and terminating simulations as underlying techniques.

Our future plans are (i) to investigate combinations of symmetry reduction with bisimulation minimization, and to extend our experimental work towards MDPS and simulation preorders, (ii) to use the results of the empirical tool comparisons in order to improve efficiency of our model checker, and (iii) to implement simulation techniques in it.

This project also takes place as part of the VOSS2 project.

Three-valued Abstraction for Continuous-Time Markov Chains

J.-P. Katoen, D. Willems, M. Leucker (TU Munich), V. Wolf (Univ. Mannheim)

In traditional model checking, abstract models contain may and must transitions as over- and under-approximation, respectively of the concrete transition relation. This concept can be lifted to Markov chains in a rather natural way by replacing transition probabilities by *intervals* where lower and upper bounds act as under- and over-approximation, respectively. We investigate such techniques for continuous-time Markov chains (CTMCs). The main technical complication is that besides transition probabilities, one has to determine the residence time of an abstract state that results from concrete states with distinct residence times. We show that intervals of transition probabilities and intervals on residence times (or combinations thereof) are not satisfactory in terms of precision. Instead, we suggest to overcome this imprecision by using *uniform* CTMCs, i.e., CTMCs in which all states have equal residence times and use transition probability intervals. The abstraction is shown to preserve simulation: concrete states are simulated by their abstract counterparts. We can show that extreme schedulers suffice, i.e., schedulers that only consider lower and upper bounds. This allows to compute reachability probabilities up to a given tolerance ε rather efficiently. Using a three-valued semantics of the *Continuous Stochastic Logic* it can be shown that the abstraction is indeed conservative for affirmative and negative verification results.

Synthesis of Design Models from Scenarios by Learning

J.-P. Katoen, C. Kern, B. Bollig (ENS Cachan, F), Martin Leucker (TU Munich)

The elicitation of requirements is the main initial phase in the typical software engineering development cycle. Popular requirement engineering methods, such as the Inquiry Cycle, exploit use cases and scenarios to specify the system's requirements. A scenario is a partial fragment of the system's behavior, describing the system components, their message exchange and concurrency. Their intuitive yet formal nature has resulted in a broad acceptance. Such scenarios can be either positive or negative, indicating a desired or unwanted system behavior, respectively. Different scenarios together form a more complete description of the system behavior.

The following design phase in software engineering is a major challenge as it is concerned with a paradigm shift between the *requirement* specification—a partial, overlapping and possibly inconsistent description of the system's behavior—and a conforming *design model*, a complete behavioral description of the system (at a high level of abstraction). During the synthesis of design models, usually automata-based models that

are focused on intra-agent communication, conflicting requirements will be detected and need to be resolved. Typical resulting changes to requirements specifications include adding or deleting scenarios, and fixing errors that are found by a thorough analysis (e.g., model checking) of the design model. Obtaining a complete and consistent set of requirements together with a related design model is thus a highly iterative process.

Our main goal is to bridge the gap between requirements, provided as a set of scenarios, and conforming design models. The novel aspect of our approach is to exploit *learning* for the synthesis of design models. In particular, we derived a procedure that infers a message-passing automaton (MPA) from a given set of positive and negative scenarios of the system's behavior provided as message sequence charts (MSCs). Furthermore we investigated which classes of regular MSC languages and corresponding MPA can or cannot be learned. Moreover we developed a dedicated tool (called *Smyle*) that supports our approach.

Verification of Erlang Programs

Th. Noll, M. Neuhäußer, L.H. Haß, C.K. Roy (Queen's Univ., Kingston, CA)

Software written for telecommunication applications usually has to meet high quality demands. Due to its complexity and its nondeterministic behaviour validation methods which are purely based on testing are generally not sufficient to ensure that the requirements are met. Therefore formal verification methods are highly desirable.

In this project we are developing and studying verification approaches which are tailored to Erlang, a programming language for implementing open, distributed telecommunication software. The complex dynamic and concurrent behaviour of such systems makes standard, finite–state model–checking techniques inapplicable in this setting.

We are tackling this problem from two different sides:

- To make Erlang systems amenable to automatic model checking techniques, one thread of our research focuses on abstraction techniques which can be employed to reduce the state space of the system under consideration. More concretely we have developed a formal definition of the syntax and semantics of a core version of Erlang in *Rewriting Logic*, a unified semantic framework for concurrency which is founded on conditional term rewriting modulo equational theories. In particular, the term rewriting machinery can be employed to model the operational behaviour of programs in terms of transition systems, and equations allow us to define abstraction mappings on the state space.

  A prototype version of an Erlang evaluator has been implemented in Maude, which is a specification language supporting the Rewriting Logic framework.

The results obtained so far are very promising, inviting to further investigate the benefits of equational abstractions for Erlang programs.

- In a second approach we try to benefit from existing work by translating the given Erlang program into a specification language for which analysis and verification methods have been already developed. Due to the dynamic and mobile process and communication structures which are present in many Erlang applications, classical languages such as LOTOS or Promela are not suitable for this purpose. Rather we are using the π*–calculus*, a name–passing process algebra which allows to describe concurrent systems with a dynamically developing communication topology.

Other Activities

J.-P. Katoen

- Member of the Steering Committee of ETAPS (*European Joint Conferences on Theory and Practice of Software*).
- Member of the Steering Committee of QEST (*Quantitative Evaluation of Systems*).
- Board Member of the Dutch Society on Theoretical Computer Science (NVTI).
- Member of the Program Committee of the following events:
  - 2nd Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2006)
  - Foundations in Software Technology and Theoretical Computer Science (FSTTCS 2006)
  - 2nd Int. Symp. on Leveraging Applications of Formal Methods (ISoLA 2006)
  - 3rd International Verification Workshop (VERIFY 2006)
  - Quantitative Aspects of Programming Languages (QAPL 2006)
  - 3rd Int. Conf. on Quantitative Evaluation of Systems (QEST 2006)
  - 150 Years Markov Anniversary Symposium (NSMC 2006)
  - 6th Int. Andrei Ershov Memorial Conf. on Perspectives of System Informatics (PSI 2006)
  - 13th GI/ITG Conf. on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB 2006)
- Member of the IFIP Working Group 1.8 on Concurrency Theory.
- Member of the EPSRC Review College (Engineering and Physical Sciences Research Council).
- Member of external PhD committees.
- Member of the editorial board of the Journal of Software.
- Guest editor (together with M. Woodside and G. Franceschinis) of a special issue of IEEE Transactions on Software Engineering on QEST 2004 (vol 32, no. 8)
- Guest editor (together with P. Buchholz and M. Verhoef) of a special issue of Journal on Software Tools for Technology Transfer on ISOLA 2004 (vol. 8, no. 6)

K. Indermark

- Member of the Editorial Board of
  - *Fundamenta Informaticae,* Annales Societatis Mathematicae Polonae
  - *Aachener Beiträge zur Informatik*
- Additional member of RWTH Faculty of Electrical Engineering and Information Technology
- Referee for Deutsche Forschungsgemeinschaft (DFG)

Th. Noll

- Program committee member of the *21st Annual ACM Symposium on Applied Computing* (SAC '06)
- Member of external PhD committees.
- Member of the examination boards for Computer Science and Computational Material Science
- Student advisor for the following subsidiary subjects within CS: Electrical Engineering, Civil Engineering, and Medicine
- Organization of teaching service of CS Department (`http://www-i2.informatik.rwth-aachen.de/Teaching/Service/`)

Talks and Publications

Talks

E. Ábrahám: *Parallel SAT Solving in Bounded Model Checking,* Presented at PDMC'06

E. Ábrahám: *Bounded Model Checking mit Parametric Data Structures,* Presented at the Fourth International Workshop on Bounded Model Checking (BMC06)

E. Ábrahám: *Dynamic Heap-Abstraction for Open, Object-Oriented Systems with Thread Classes,* Invited talk at Logical Approaches to Computational Barriers (CiE'06)

H. Bohnenkamp: *Synthesis and Stochastic Assessment of Cost-Optimal Schedules,* Post-MMB-Workshop, March 2006, Erlangen

H. Bohnenkamp: *Are you still there?,* IPA Herfstdagen 2006, Bergen, NL

H. Bohnenkamp: *Axxom, Scheduling, Costs and Optimums,* Voss-Meeting January 2006, Twente University

H. Bohnenkamp: *Timed Testing with TorX,* IPA Lentedagen 2006, Vught, NL

H. Bohnenkamp: *Timed Testing with TorX,* ITG Workshop Model-Based Testing 2006, Nürnberg

T. Han: *Counterexamples in Probabilistic Model Checking,* VOSS2 Workshop, Rolduc NL, Sept. 2006

T. Han: *Counterexamples in Probabilistic Model Checking,* IPA Herfstdagen on Stochastic Systems, Bergen NL, Nov. 2006

K. Indermark: *Farewell addresses to the Computer Science Graduates, Sommerfest, July 7, and Tag der Informatik, December 1, 2006,*

J.-P. Katoen: *Advances in Probabilistic Model Checking,* Invited Presentation at the German Verification Day, Aug. 2006

J.-P. Katoen: *Advancing Probabilistic Model Checking,* Workshop on Advances in Model Checking (in honour of Gerard J. Holzmann), Nov. 2006

J.-P. Katoen: *Counterexamples in DTMC Model Checking,* Dagstuhl Seminar on Specification, Verification, and Testing of Open Systems, Oct. 2006

J.-P. Katoen: *Foundations of Probabilistic Model Checking,* IPA Autumn Days on Stochastic Systems, Nov. 2006

J.-P. Katoen: *Model Checking LTL and SPIN,* IPA Basic Course on Formal Methods, Jan. 2006

J.-P. Katoen: *Model Checking Probabilistic Systems,* Four lectures at the University of Florence, June 2006

C. Kern: *MSCan - A Tool for Analyzing MSC Specifications,* TACAS March 2006 (Wien)

C. Kern: *Synthesis of Design Models from Scenarios by Learning,* RWTH Aachen University,Doctoral Symposium, December 2006

Th. Noll: *Towards Automatic Verification of Erlang Programs by pi-Calculus Translation,* ACM SIGPLAN 2006 Erlang Workshop, September 2006

S. Rieger: *Optimierung linearen Codes,* 23. Workshop der Gi-FG Programmiersprachen und Rechenkonzepte, Bad Honnef, May 2006

I.S. Zapreev: *On-the-fly Steady-state detection,* University of Freiburg, Germany, July 2006

I.S. Zapreev: *Safe On-The-Fly Steady-State Detection for Time-Bounded Reachability,* QEST'06, Riverside, CA, USA, Sept. 2006

Publications

E. Ábrahám, T. Schubert, B. Becker, M. Fraenzle, C. Herde: *Parallel SAT Solving in Bounded Model Checking, ,* Proc. of PDMC'06

E. Ábrahám, M. Herbstritt, B. Becker, M. Steffen: *Memory-Aware Bounded Model Checking for Linear Hybrid Systems,* Proceedings of the 9th. Workshop for Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV'06)

E. Ábrahám, M. Herbstritt, B. Becker, M. Steffen: *Bounded Model Checking mit Parametric Data Structures,* Proc. of the Fourth International Workshop on Bounded Model Checking (BMC'06)

E. Ábrahám, A. Grüner, Martin Steffen: *Dynamic Heap-Abstraction for Open, Object-Oriented Systems with Thread Classes,* Proc. of Logical Approaches to Computational Barriers: Second Conference on Computability in Europe (CiE'06)

E. Ábrahám, A. Grüner, M. Steffen: *Dynamic Heap-Abstraction for Open, Object-Oriented System with Thread Classes,* Technical Report, Christian-Albrechts-University Kiel, Institute for Computer Science, TR 0601, 2006

E. Ábrahám, A. Grüner, M. Steffen: *Abstract Interface Behavior of Object-Oriented Languages with Monitors,* Proc. of the 8th IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS'06)

E. Ábrahám, A. Grüner, M. Steffen: *Abstract Interface Behavior of Object-Oriented Languages with Monitors,* Technical Report, Christian-Albrechts-University Kiel, Institute for Computer Science, TR 0612, 2006

C. Baier, H. Hermanns, J.-P. Katoen, V. Wolf: *Bisimulation and Simulation Relations for Markov Chains,* Electr. Notes Theor. Comput. Sci. 162: 73-78 (2006)

J. Berendsen, D.N. Jansen, J.-P. Katoen: *Probably on time and within budget – On reachability in priced probabilistic timed automata,* Quantitative Evaluation of Systems (QEST), IEEE CS Press, Riverside USA, Sept. 2006

H. Bohnenkamp, P.R. D'Argenio, H. Hermanns, J.-P. Katoen: *MoDeST: A compositional modeling formalism for real-time and stochastic systems,* IEEE Transactions on Software Engineering, 32(10), pp. 812–830

B. Bollig, C. Kern, M. Schlütter, V. Stolz: *MSCan - A Tool for Analyzing MSC Specifications,* Proceedings of the 12th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'06), Lecture Notes in Computer Science 3920, pp. 455–458, Springer, March 2006

B. Bollig, J.-P. Katoen, C. Kern, M. Leucker: *Replaying Play in and Play out: Synthesis of Design Models from Scenarios by Learning,* Research Report, AIB-2006-12, RWTH Aachen, Oct. 2006

M. Bravetti, H. Hermanns, J.-P. Katoen: *YMCA: Why Markov Chain Algebra?,* Electronic Notes in Theoretical Computer Science, 162, pp. 107–112, 2006

R. De Nicola, J.-P. Katoen, D. Latella, M. Massink: *Towards a logic for performance and mobility,* Electronic Notes in Theoretical Computer Science, 153, pp. 161–175, 2006

D. Distefano, J.-P. Katoen, A. Rensink: *Safety and liveness in concurrent pointer programs,* Formal Methods for Components and Objects, LNCS 4111, pp. 280–312, Springer, Nov. 2006

T. Han, J.-P. Katoen: *Counterexamples in probabilistic model checking,* Technical Report, AIB 006-09, RWTH Aachen, July 2006

K. Indermark, Th. Noll: *Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information,* Acta Informatica 43, pp. 1–43, 2006

Th. Noll, S. Rieger: *Optimization of Straight-Line Code Revisited,* Softwaretechnik-Trends 26(2), Gesellschaft für Informatik e.V., 2006

Th. Noll, C.K. Roy: *Towards Automatic Verification of Erlang Programs by pi-Calculus Translation,* Proceedings of the ACM SIGPLAN 2006 Erlang Workshop, pp. 38–50, ACM Press, 2006

J.-P. Katoen: *Constrained-Oriented Specification of Performance Aspects,* In: M.J. van Sinderen and L. Ferreira Pires (eds), Architectural Design of Open Distributed Systens: From Interface to Telematics, Liber Amicorum dedicated to Chris Vissers, pp. 47–54, Universal Press, 2006

J.-P. Katoen, I.S. Zapreev: *Safe on-the-fly steady-state detection for time-bounded reachability,* Quantitative Evaluation of Systems (QEST), pp. 301–310, IEEE CS Press, Riverside, USA, Sept. 2006

J.-P. Katoen: *Stochastic model checking,* Stochastic Hybrid Systems: Recent Developments and Research Trends, Taylor and Francis, Nov. 2006

A. Mader, H. Bohnenkamp, Y.S. Usenko, D.N. Jansen, J. Hurink, H. Hermanns: *Synthesis and Stochastic Assessment of Cost-Optimal Schedules,* Technical Report, 06-14, University of Twente, 2006

M. Neuhäußer, Th. Noll: *Abstraction and Model Checking of Core Erlang Programs in Maude,* Proceedings of the 6th International Workshop on Rewriting Logic and its Applications (WRLA'06), Vienna, April 2006, Elsevier Inc., to appear

Th. Noll, C.K. Roy: *Towards Automatic Verification of Erlang Programs by pi-Calculus Translation,* Proceedings of the ACM SIGPLAN 2006 Erlang Workshop, ACM Press, 2006, 38–50

M. Weber: *Parallel Algorithms for Verification of Large Systems,* PhD Thesis, Technical Report AIB-2006-02

D. Willems: *Abstraktion zeitstetiger Markov-Ketten,* Diploma Thesis, Faculty of Mathematics, Computer Sciences and Natural Sciences, RWTH Aachen University, April 2006