Activity Report 2014

Exploratory Action ESTASYS

Efficient STAtistical methods in SYstems of systems

RESEARCH CENTER
Rennes - Bretagne-Atlantique

Table of contents

Exploratory Action ESTASYS

Keywords: Embedded Systems, Model-checking, Monte Carlo Methods, Model-driven Engineering

Estasys is an exploratory action create on February 2014. It is also a team from the Inria Rennes center.

Creation of the Exploratory Action: 2014 January 01.

1. Members

Research Scientist
Axel Legay [Team Leader, Researcher, Inria]

Engineers
Sean Sedwards [inria]
Rudolf Fahrenberg [Inria]
Louis-Marie Traonouez [inria]
Ngo Van-Chan [Inria,Started August 2014]
Kevin Corre [Inria,Left October 2014]

PhD Students
Cyrille Jegourel [Inria]
Mounir Chadli [PhD grant Algeria,Started December 2014]

Post-Doctoral Fellows
Jean Quilbeuf [Inria,Started August 2014]
Benoît Boyer [Inria,Left October 2014]
Fabrizio Biondi [Inria,Started April 2014]
Thomas Given-Wilson [Inria,Started October 2014]
Jin Hyun Kim [Inria,Started December 2014]

Administrative Assistant
Loïc Lesage

2. Overall Objectives

2.1. Overall Objectives

Computer systems play a central role in modern societies and their errors can have dramatic consequences. Industry and academics thus invest a considerable amount of effort developing techniques to prove the correctness of these systems. Among such techniques, one finds (1) *testing*, the traditional approach to detect bugs with test cases, and (2) *formal methods*, e.g., model checking (Turing award), that can *guarantee* the absence of bugs. Both approaches have been largely deployed on static systems, whose behaviour is entirely known. **ESTASYS focuses on developping brand new formal methods for Systems of Systems.**

3. Research Program

3.1. Systems of Systems, Heterogeneous Systems, Dynamicity, Statistical Model Checking

Formal methods rely on the notion of *transition system* (TS): an abstract machine that characterises a system's *complete* behaviour. This machine consists of a complete set of states (each representing full knowledge of the system at a given moment) and transitions between states, which may be labelled with labels chosen from some set of actions. This definition makes it necessary to have advanced knowledge of all the possible states of the system – to have a statically configured system. The algorithms used by formal methods perform an exhaustive exploration of the state space of the TS, so such methods suffer from the so-called *state-space explosion problem*. As a consequence, there are many real systems that are beyond the scope of such techniques. Despite this, over the last thirty years it has been shown that, when combined with heuristics such as partial order reductions or abstraction, **formal approaches are powerful enough to verify industrial-scale systems**.

The first wave of techniques was deployed to verify whether a certain set of (problem) states can be reached ('reachability'). Later, extensions of TS, such as *hybrid systems* and *stochastic automata*, were proposed to cope with new problems (e.g., energy consumption) or to reason on distributed real-time embedded components (possibly heterogeneous). It was quickly observed that the complexity of assessing correctness of such extended models arises not exclusively from the fact that they are large, but also because they introduce *undecidability*. As a concrete example, the reachability problem is already undecidable for any real-time system whose time evolution is described by a non-constant derivative equation.

This motivated the development of more efficient techniques that approximate the answer to the original problem or approximate the problem. Of these, perhaps the most successful quantitative technique is *Statistical Model Checking* , that can be seen as a trade-off between testing and formal verification. The core idea of SMC is to generate a number of *simulations* of the system and verify whether they satisfy a given property expressed in temporal logics, which can be done by using *runtime verification approaches*. The results are then used together with algorithms from the statistical area in order to decide whether the system satisfies the property with some probability. SMC resembles classical simulation-based techniques used in industry, but uses a formal model of systems and requirements. This not only gives a rigorous meaning to industrial practices, but also makes available more than twenty years of research in the area of *runtime verification*. Last but not least, **the use of statistical algorithms allows us to approximate undecidable problems**. Recent successful applications of SMC can be found in systems biology, security protocols and avionics. In particular, SMC was used to discover inconsistent requirements of an EADS airplane communication system.

3.1.1. Systems of Systems (SoS)

The advent of service-oriented and cloud architectures is leading to generations of computer systems that exhibit a new type of complexity: such systems are no longer statically configured, but comprise components that are systems in their own right, able to discover, select and bind on-the-fly to other components that can deliver services that they require. These complex systems, referred to as *Systems of Systems* (SoS), can change over time as each component creates and modifies the network over which it needs to operate: as they execute, the components create a network of their own and use it to fulfil their goals.

The Internet, made up of an unsupervised and rapidly growing, dynamically configured set of computers and physical connections, is an obvious illustration of the potential complexity of dynamic networks of interactions. Another example is the so-called "Flash Crash" in the U.S. equity market: on May 6, 2010, a block sale of 4.1 billion dollars of futures contracts executed on behalf of a fund-management company triggered a complex pattern of interactions between the high-frequency algorithmic trading systems (algos) that buy and sell blocks of financial instruments and made the Dow Jones Industrial Average drop more than 600 points, representing the disappearance of 800 billion dollars of market value. This example is an illustration of the faulty divergence of SoS behaviour, where the system starts to misbehave and dynamically creates new components that follow the same pattern and make the problem worse. Examples of this include

when a SoS detects high energy use and invokes a new component to reduce the energy, thus consuming *more* energy. **Until now, such divergence has been mostly handled by humans that eventually observe the faulty behaviour and manually intervene to stop it. This human-based solution is not always successful and clearly unsatisfactory, since it acts retrospectively, when the system has already failed.**

3.1.2. Grand Challenge and Breakthroughs of ESTASYS

SoS are an efficient means of achieving high performance and are thus becoming ubiquitous. Society's increasing reliance on SoS demands that they are reliable, but tools to guarantee this at the design stage do not exist. Most conventional formal analysis techniques, even those dedicated to adaptive systems, fail when applied to SoS because they are designed to reason on systems whose state space can be predicted in advance. **The grand challenge addressed by ESTASYS is the fundamental overhaul of formal methods techniques in the design of SoS life cycle**.

It is clear that SMC can be applied to the verification of complex systems. Unfortunately, SMC cannot yet be applied to SoS, because existing techniques are designed to capture the behaviour of statically configured systems, or systems whose dynamical configuration arises from permutations of known components. ESTASYS defines new abstract computational models and extend the state of the art of SMC to include SoS.

ESTASYS proposes a new formal methodology to support an evolutionary adaptive and iterative SoS life-cycle. *We foresee the following breakthroughs:*

1. Our ground-breaking computational model addresses the complex dynamic nature of SoS. The model is based on new interface theories that take into account behaviours of possibly unknown components and thus abstract what is unknown.
2. Cutting edge algorithms coming from the area of statistics and learning are exploited to make predictions about autonomous systems making local decisions. For example, **statistical abstraction** abstracts the behaviour of unknown environments; interleaving analysis and runtime monitoring of deployed systems to continuously update distributions embedded in the interfaces.
3. New statistical algorithms for SMC that scale efficiently and handle undecidability impacts the formal analysis of complex systems.
4. Our results are implemented in a professional toolset, ESTASYS-PLASMA, that is constructed in close collaboration with our industrial partners. This ensures relevance to industry and potentially high impact in the marketplace.

3.1.3. Methodology and Organization

ESTASYS's main challenge is to lay the foundation of a novel rigorous software construction methodology for SoS, based on simulation, statistics and industrial practices. ESTASYS establishes theories and empirical evidence for the introduction of formal verification-based approaches in the rigorous design of SoS.

ESTASYS addresses essential research questions for the introduction of formal techniques to support the SoS life-cycle. SoS occur in multiple disciplines and therefore there is a need for a common language. In particular, notions such as **autonomous decisions and dynamicity** must be standardized and well understood by those that will apply our methodology. Additionally, **characterizing the topological structure** of a SoS is essential for the study of component interactions and data exchanges. The complexity of SoS requires the development of a **sound formal semantic foundation** to support deployment of formal methods. We thus identify a minimal computational model that characterize SoS, on which classes of properties of interest can be defined. The project investigates new simulation-based approaches, combined with other domains (statistics, learning, ...), to verify such properties on the new computational model. Finally, ESTASYS identifies under which conditions the new techniques can be used, to take decisions during design and evolution time, leading to a fully integrated development cycle.

ESTASYS focuses on both the static and dynamic properties of SoS. ESTASYS establishes models for each component and investigates the connection and dynamical interactions between them. ESTASYS's activities are organize in six main tasks: tasks 1, 2 and 3 are responsible for breakthrough 1; task 4 is responsible for breakthrough 2; task 5 is responsible for breakthrough 3; task 6 is responsible for breakthrough 4.

Task 1. Characterizing SoS. Examples of SoS found in various areas, such as health care, smart buildings and energy grids, are analysed and used to standardize notions of autonomous decisions and dynamicity. We also study and classify SoS-related problems, such as faulty behaviour divergence. Our objective is to derive in Task 2 formal models that abstract the above classification.

Task 2. Formal Modeling of SoS. Classical theories do not provide for SoS, hence we require new formal models for SoS that take into account (i) dynamicity and emergent behaviours, (ii) autonomous decisions of components, and (iii) architectural constraints, including information regarding the viability of the hardware. In particular, we devise new logics tailored to the specific needs of SoS. Such logics, dynamic by nature, includes extended notions of quantification, such as energy, and considers hardware constraints and distributions of system configurations. Task 2 includes modelling the various components running within the SoS and their (dynamical) interactions. This requires the definition of a new type of interface able to work with heterogeneous components and to abstract the behaviour of unknown resources. Interfaces act as an abstraction for the internal behaviour of each component and encodes the dynamical constraints of the SoS. They are used to (i) model and define the authorised interactions between the components, (ii) reason on dynamical aspects and (iii) abstract unknown behaviour.

Task 3. Statistical abstraction interleaving design and deployment. Abstraction techniques are necessary to reduce the complexity of SoS and to model uncertainty. Specifically, **statistical abstractions** of the observed runtime behaviour of components is used to quantify, e.g., the probability that a number of new components satisfying some constraints is started at a given execution point. Runtime verification monitors the executions of the deployed system to create distributions embedded in the interfaces developed in Task 1. When a deployed system is available, ESTASYS interleaves simulation, analysis and runtime monitoring, using real behaviour to update the statistical abstractions, and eventually replace some of those abstractions by concrete ESTASYS-Interface models. The ESTASYS methodology adopts a Bayesian approach: (i) an initial, plausible distribution is 'guessed', based on whatever is known; (ii) the system is simulated using the current approximated distribution; (iii) the behaviour of the simulated system becomes the new approximation; (iv) the process is iterated as necessary. While learning-based simulation approaches, such as model fitting, can be used to learn the abstraction by conducting simulations from a finite set of initial components, we have to provide clear evidence that a global property holds on the system if it holds on its corresponding statistical abstraction. The task requires strong competences in statistics.

Task 4. Developing Efficient Simulation and Monitoring Algorithms for SoS. The ground-breaking models developed in Task 2 requires efficient simulation and monitoring techniques. This necessitates the study of new algorithms for dynamically configured systems and monitoring approaches to reason on heterogeneous components and the new quantitative logics and interface paradigms developed in Task 2.
A major difficulty in developing monitoring techniques for SoS is that the components have their own goals and behave differently in different environments. Unnecessary high-level hypotheses on properties may drastically increase simulation time and should be avoided.

Task 5. Developing Efficient Statistical Techniques for SoS. SoS pose new challenges for statistical techniques, requiring the study of new SMC algorithms dedicated to SoS goals. In contrast to existing SMC algorithms that can only be applied to pure stochastic systems, SMC algorithms for SoS have to take into account the non-deterministic aspects of autonomous decisions made by neighbour components. We postulate that this can be done by extending very recent advances in reinforcement learning algorithms. Rare events play an important role in system reliability, so we include rare-event simulation algorithms, such as importance sampling and importance splitting, which can reduce variance and significantly increase simulation efficiency.

Task 6. Evaluating the impact of statistical and simulation-based techniques. Evidence of the success of ESTASYS is provided by the publishing of a complete experimental environment, ESTASYS-PLASMA, that supports the empirical validation of ESTASYS's theories. ESTASYS-PLASMA contains efficient implementations of the results discovered in Tasks 2-5, and will provide intuitive feedback mechanisms so that the engineer can use the results of the verification process to improve SoS design.

4. Application Domains

4.1. Application Domains

In two years, ESTASYS should lead to the creation of a top class research team at Inria as well as to an interdisciplinary community of researchers and practitioners at the world level.

ESTASYS sets the foundations for an engineering domain dedicated to SoS that will benefit the European software industry. This is achieved by creating mathematical models that capture the computational power, autonomous decisions and complex stochastic and real-time dynamics of SoS. ESTASYS produces new decidability and complexity results, simulation-based techniques, and algorithms with correctness arguments. All aim at efficient reasoning about SoS and are traced back to case studies. **Our strategy to work in close collaboration with contact in industry will guarantee their wider adoption by the european software industry.**

In the near future, The ESTASYS-PLASMA toolset will be distributed as open source whenever possible, but will create a new market of tools for SoS.

5. New Software and Platforms

5.1. The Plasma Statistical Model Checker

Participants: Axel Legay [Coordinator], Sean Sedwards, Benoît Boyer, Louis-Marie Traonouez, Kevin Corre.

5.1.1. PLASMA

Statistical model checking (SMC) is a fast emerging technology for industrial scale verification and optimisation problems. In recognition of this, our group is developing a *P*latform for *L*earning and *A*dvanced *S*tatistical *M*odel checking *A*lgorithms: PLASMA.

PLASMA (see https://project.inria.fr/plasma-lab/) was conceived to have high performance and be extensible, using a proprietary virtual machine [48]. Since SMC requires only an executable semantics and is not constrained by decidability, we can easily implement different modelling languages and logics. Our involvement in the DANSE [1] and DALi [2] European projects has also made us aware of the need to provide efficient verification for externally implemented simulators. We thus devised PLASMA-lab, a modular SMC library that allows external users to tightly integrate their own code with our efficient SMC algorithms and integrated development environment [47]. PLASMA-lab has now been successfully integrated with DESYRE [3], Scilab [4] and MATLAB [5].

The PLASMA-lab architecture is now the basis of our free-standing tool, [6] which includes all the modelling languages, logics and algorithms developed by our group. In particular, we have recently developed cutting edge algorithms for rare events [50], [49], [26], nondeterminism [28], [34], [37] and learning [14], [41].

5.2. Quail

Participants: Axel Legay [Coordinator], Fabrizio Biondi [Coordinator], Jean Quilbeuf.

[1] http://www.danse-ip.eu
[2] http://www.ict-dali.eu
[3] http://www.ales.eu.com
[4] http://www.scilab.org
[5] http://www.mathworks.com
[6] https://project.inria.fr/plasma-lab

Privacy is a central for Systems of Systems and interconnected objects. We propose QUAIL, a tool that can be used to quantify privacy of components. QUAIL is the only tool able to perform an arbitrary-precision quantitative analysis of the security of a system depending on private information. Thanks to its Markovian semantics model, QUAIL computes the correlation between the system's observable output and the private information, obtaining the amount of bits of the secret that the attacker will infer by observing the output. QUAIL is open source and can be downloaded at https://project.inria.fr/quail/.

QUAIL is able to evaluate the safety of randomized protocols depending on secret data, allowing to verify a security protocol's effectiveness. QUAIL can also be used to find previously unknown security vulnerabilities in software systems and security protocols. The tool can verify whether a protocol is protecting its secret in a perfect way, and quantify how much the secret is exposed to being revealed otherwise.

QUAIL has been used to quantify whether voting protocols respect the anonymity of the voters, proving that preference ranking voting schemes are more secure than single preference ones. It has also been applied to the security of smart grids and a number of classic examples like dining cryptographers, authentication protocols and grades protocol.

Since its initial release in 2013, QUAIL's algorithm has been improved employing abstract trace exploration and statistical estimation techniques, making it thousands of times faster than the initial version and outperforming other comparable analysis tools on most use cases.

5.3. PyEcdar

Participants: Axel Legay [Coordinator], Louis-Marie Traonouez [Coordinator].

One of the main difficulties with Systems of Systems is to describe the connection and interactions between the components. We propose PyEcdar as a solution to this problem. PyEcdar (https://project.inria.fr/pyecdar/) is a free software that analyses timed games and timed specifications. The goal of the tool is to allow a fast prototyping of new analysis techniques. It currently allows to solve timed games based on timed automata models. These can be extended with adaptive features to represent dynamicity and to model software product lines.

The tool has been originally developed to analyze the robustness of timed specifications, in extension of the tool Ecdar (http://people.cs.aau.dk/~adavid/ecdar/). As Ecdar, it allows to compose components specifications based on Timed I/O Automata (TIOA), and it implements timed game algorithms for checking consistency and compatibility. Additionally, it features original methods for checking the robustness of these specifications.

The tool has been later extended to analyse adaptive systems. It therefore implements original algorithms for checking featured timed games against requirements expressed in the timed AdaCTL logic.

The tool is written in Python with around 3'000 lines of code. It uses a Python console as user interface, from which it can load TIOA components from XML files written in the UPPAAL format (http://www.uppaal.org/), and design complex system by combining the components using a simple algebra. Then, it can analyze these systems, transform them and save them in a new XML file.

6. New Results

6.1. Highlights of the Year

The Plasma statistical model checker has been made available to other scientists. ESTASYS has open a new branch on verifying the security of complex systems.

6.2. Verification of Heterogeneous Systems

Participants: Axel Legay, Benoît Boyer, Ngo Van-Chan, Jean Quilbeuf.

This part concerns Tasks 1, 2 and 4 of the action. We characterize and formalize heterogeneous aspects of SoS and then we define efficient monitoring algorithms and representations for their requirements. We then combine the results with Statistical Model Checking (Task 5).

Systems of Systems (SoS) are very large scale systems with particular characteristics. SoS are not directly built from scratch by a single designer or a single team but are obtained as the composition of simpler systems. SoS have strong reliability and dependability requirements, as they aim to provide a service over a long running period. SoS may dynamically modify themselves by connecting to new systems, updating or disconnecting faulty ones, making it impossible to statically know the set of subsystems that are part of the SoS before runtime.

One of the main difficulty arising when developing SoS is the fact that subsystems may have been designed with a different goal in mind. In particular, some subsystems may have their own goal which differs from the global goal of the SoS. Furthermore, each subsystem may be developed in a particular computation model, making it difficult to find a common unifying semantics for the whole SoS. Finally, SoS may exhibit some emergent behaviors that are hardly predictable at design time.

One of the solutions to allow simulation of a SoS is to rely on a common interface for interconnecting the subsystems. The Functional Mockup Interface (FMI) standard is a natural candidate for such an interface. The different components of a SoS developed in different models of computation can be translated to Functional Mockup Units (FMU). Then a so-called master algorithm coordinates the FMUs composing the system. The execution of each FMU is either directly handled by the master algorithm or relies on an external tool for its execution.

Because the subsystems composing a SoS are of heterogeneous nature, it is difficult to find a common semantics model for the whole system. Furthermore, building such a transition system is not tractable due to the complexity of the system. Thus verification through traditional model checking is not possible for SoS. However, since the FMI/FMU framework enables simulation of such systems, the statistical model checking approach can be used.

The DANSE EU project aims to provide a complete tool chain from the modeling to the verification of SoS. At the higher level, the modeling is done in UPDM using the RHAPSODY tool. At the same level, the designer can express requirements over the model using some patterns written in GCSL. The UPDM model can then be translated into a FMI/FMU format that can be simulated by a dedicated tool, named DESYRE. Similarly, the GCSL requirements are transformed into BLTL formulas. Finally, the PLASMA statistical model checker has been integrated with the DESYRE tool chain in order to check the BLTL formulas based on the simulations provided by DESYRE.

6.2.1. Papers:

[45] (W) This report presents some of the results of the first year of Danse, one of the first EU IP projects dedicated to System of Systems. Concretely, we offer a tool chain that allows to specify SoS and SoS requirements at high level, and analyse them using powerful toolsets coming from the formal verification area. At the high level, we use UPDM, the system model provided by the british army as well as a new type of contract based on behavioral patterns. At low level, we rely on a powerful simulation toolset combined with recent advances from the area of statistical model checking. The approach has been applied to a case study developed at EADS Innovation Works.

[51] (W) Exhaustive formal verification for systems of systems (SoS) is impractical and cannot be applied on a large scale. In this paper we propose to use statistical model checking for efficient verification of SoS. We address three relevant aspects for systems of systems: 1) the model of the SoS, which includes stochastic aspects; 2) the formalization of the SoS requirements in the form of contracts; 3) the tool-chain to support statistical model checking for SoS. We adapt the SMC technique for application to heterogeneous SoS. We extend the UPDM/SysML specification language to express the SoS requirements that the implemented strategies over the SoS must satisfy. The requirements are specified with a new contract language specifically designed for SoS, targeting a high-level English-pattern language, but relying on an accurate semantics given by the standard temporal logics. The

contracts are verified against the UPDM/SysML specification using the Statistical Model Checker (SMC) PLASMA combined with the simulation engine DESYRE, which integrates heterogeneous behavioral models through the functional mock-up interface (FMI) standard. The tool-chain allows computing an estimation of the satisfiability of the contracts by the SoS. The results help the system architect to trade-off different solutions to guide the evolution of the SoS.

6.3. Formal Models for Variability

Participants: Axel Legay, Rudolf Fahrenberg, Jin Hyun Kim.

This part of the report is more concerned with task 2. It studies variability aspects in the broad scope. To simplify the study for the first year, we use the concept of software product lines. Later we shall use the results in federation of embedded systems, which is a particular class of Systems of systems.

Variability is ubiquitous in today's systems, be it in the form of configuration options or extensible architectures. By mastering variability, developers can adapt their system to changing requirements without having to develop entirely new applications. Variability is central in the context of SoS whose behaviors depend on interconnected objects. To gain information on managing variability, we have focused on Software Product Lines. Software Product Lines (SPLs) are a popular form of variability-intensive systems. They are families of similar software systems developed together to make economies of scale. SoS can be viewed as examples of product lines with interconnected obbjects. SPL engineering aims to facilitate the development of the members of a family (called *products* or *variants*) by identifying upfront their commonalities and differences. Variability in SPLs is commonly represented in terms of *features*, *i.e.*, units of difference between products that appear natural to stakeholders. Each product of an SPL is therefore defined by its set of features. Hierarchies of features and dependencies between features (*e.g.*, requires, excludes) are typically captured in a *Feature Model* (FM), *i.e.* a tree-like structure that specifies which combinations of features are valid.

6.3.1. Papers:

[15] (C) The model-checking problem for Software Products Lines (SPLs) is harder than for single systems: variability constitutes a new source of complexity that exacerbates the state-explosion problem. Abstraction techniques have successfully alleviated state explosion in single-system models. However, they need to be adapted to SPLs, to take into account the set of variants that produce a counterexample. In this paper, we apply CEGAR (Counterexample-Guided Abstraction Refinement) and we design new forms of abstraction specifically for SPLs. We carry out experiments to evaluate the efficiency of our new abstractions. The results show that our abstractions, combined with an appropriate refinement strategy, hold the potential to achieve large reductions in verification time, although they sometimes perform worse. We discuss in which cases a given abstraction should be used.

-

[18] (C) In this work, We explore how ideas of statistical testing, based on a usage model (a Markov chain), can be used to extract configurations of interest according to the likelihood of their executions. These executions are gathered in featured transition systems, compact representation of SPL behaviour. We discuss possible scenarios and give a prioritization procedure validated on a web-based learning management software.

6.4. Statistical Model Checking

Participants: Axel Legay, Sean Sedwards, Benoît Boyer, Louis-Marie Traonouez, Kevin Corre.

This section covers Tasks 4 and 5 of the action. It consists in developping Simulation based techniques and efficient statistical algorithms for SoS.

The use of test cases remains the default means of ensuring the correct behaviour of systems in industry, but this technique is limited by the need to hypothesise scenarios that cause interesting behaviour and the fact that a reasonable set of test cases is unlikely to cover all possible eventualities. Static analysis is more thorough and has been successful in debugging very large systems, but its ability to analyse complex dynamical properties is limited. In contrast, model checking is an exhaustive technique that verifies whether a system satisfies a dynamical temporal logic property under all possible scenarios. For nondeterministic and probabilistic systems, numerical model checking quantifies the probability that a system satisfies a property. It can also be used to quantify the expected cost or reward of sets of executions.

Numerical model checking gives precise, accurate and certain results by exhaustively exploring the state space of the model, however the exponential growth of the state space with system size (the 'state explosion problem) typically limits its applicability to "toy" systems. Symbolic model checking using efficient data structures can make certain very large models tractable. It may also be possible to construct simpler but behaviourally equivalent models using various symmetry reduction techniques, such as partial order reduction, bisimulation and lumping. If a new system is being constructed, it may be possible to guarantee the overall behaviour by verifying the behaviour of its subcomponents and limiting the way they interact. Despite these techniques, however, the size, unpredictability and heterogeneity of real systems usually make numerical techniques infeasible. Moreover, even if a system has been specified not to misbehave, it is nevertheless necessary to check that it meets its specification.

Simulation-based approaches are becoming increasingly tractable due to the availability of high performance parallel hardware and algorithms. In particular, statistical model checking (SMC) combines the simplicity of testing with the formality of numerical model checking. The core idea of SMC is to create multiple independent execution traces of a system and count how many satisfy a property specified in temporal logic. The proportion of satisfying traces is an estimate of the probability that the system satisfies the property. By thus modelling the executions of a system as a Bernoulli random variable, the absolute error of the estimate can be bounded using, for example, a confidence interval or a Chernoff bound. It is also possible to use efficient sequential hypothesis testing, to decide with specified statistical confidence whether the probability of a property is above or below a given threshold. Since SMC requires multiple independent simulations, it may be efficiently divided on parallel computer architectures, such as grids, clusters, clouds and general purpose computing on graphics processors (GPGPU).

Knowing a result with less than 100% confidence is often sufficient in real applications, since the confidence bounds may be made arbitrarily tight. Moreover, a swiftly achieved approximation may prevent a lot of wasted time during model design. For many complex systems, SMC offers the only feasible means of quantifying performance. Historically relevant SMC tools include APMC, YMER and VESTA. Well-established numerical model checkers, such as PRISM and UPPAAL, are now also including SMC engines. Dedicated SMC tools under active development include COSMOS and our own tool PLASMA. Recognising that SMC may be applied to any discrete event trace obtained by stochastic simulation, we have devised PLASMA-lab, a modular library of SMC algorithms that may be used to construct domain-specific SMC tools. PLASMA-lab has become the main vehicle of our ongoing development of SMC algorithms.

Our group is devising cutting edge techniques for SMC. In particular, we are developing new learning algorithms (Sect. 6.4.3), algorithms for nondeterministic systems (Sect. 6.4.1), and algorithms for rare events (Sect. 6.4.2).

6.4.1. Algorithms for Nondeterminism

Nondeterministic models are of fundamental importance in defining complexity and are useful models of concurrency optimisation problems. This latter application is of particular importance in the context of systems constructed from subsystems ("Systems of Systems") that interact in an unpredictable way. Verifying or optimising such systems is problematic for numerical techniques because the state space is typically intractable. Nondeterminism is challenging for simulation-based techniques because, by definition, an executable semantics is not determined.

We have thus begun a line of research to develop SMC algorithms for nondeterministic systems. Our initial focus is Markov decision processes (MDP), however we are in the process of extending our work to various nondeterministc timed automata. Recent attempts to provide approximative algorithms for MDPs either do not address the standard verification problems, consider only a "spurious" subset of the standard problems or contain significant misconceptions and limitations.

In [28], we presented the first complete set of scalable SMC algorithms for MDPs. Our techniques are based on the idea of encoding a history-dependent scheduler as the seed of a pseudo-randomised hash function. Schedulers are thus chosen at random by selecting random seeds. The possibly infinite behaviour of the scheduler is completely encoded in $\mathcal{O}(1)$ memory. We presented simple sampling algorithms to find optimal schedulers and constructed the statistical confidence bounds necessary to find the optima of multiple estimates.

In [34] we devised the notion of "smart sampling" to dramatically improve the performance of the simple algorithms presented in [28]. The basic idea is to use part of the simulation budget to generate a crude estimate of the optimal scheduler and to use this information to better allocate the remaining budget. We successfully applied our algorithms to a number of standard case studies from the literature. We also highlighted the limitations of our approach.

The algorithms in [28], [34] find schedulers that minimise or maximise the probability of a property. In [37] we have adapted our algorithms to minimise or maximise the expected reward of a system. This adaptation is not entirely straightforward because the standard definition of reward properties assumes an exhaustive exploration of the state space of the MDP. We have included an implicit hypothesis test to include this assumption. In other respects optimising rewards is less challenging than optimising probabilities because rewards are effectively based on properties having probability 1. We demonstrate the accuracy of our rewards-based algorithms on standard case studies from the literature.

6.4.2. Rare Events in SMC

Rare properties are often highly relevant to system performance (e.g., bugs and system failure are required to be rare) but pose a problem for statistical model checking because they are difficult to observe. Fortunately, rare event techniques such as *importance sampling* and *importance splitting* may be successfully applied to statistical model checking.

In a previous work [50], we explicitly considered the use of importance sampling in the context of statistical model checking. We presented a simple algorithm that uses the notion of cross-entropy to find the optimal parameters for an importance sampling distribution. In contrast to previous work, our algorithm uses a low dimensional vector of parameters to define this distribution and thus avoids the often intractable explicit representation of a transition matrix. We showed that our parametrisation leads to a unique optimum and can produce many orders of magnitude improvement in simulation efficiency. We demonstrated the efficacy of our methodology by applying it to models from reliability engineering and biochemistry.

Our contribution [49] was the first attempt to use importance splitting with SMC to overcome the Rare Event problem. The basic idea is to decompose a logical property into nested properties whose probabilities are easier to estimate. Importance splitting achieves this by estimating a sequence of conditional probabilities, whose product is the required result. To apply this idea to model checking it is necessary to define a score function based on logical properties, and a set of levels that delimit the conditional probabilities. We described the necessary and desirable properties of score functions and levels. We illustrated how a score function may be derived from a property and gave two importance splitting algorithms: one that uses fixed levels and one that discovers optimal levels adaptively.

6.4.3. SMC with Changes and Simulink

We have proposed a new SMC algorithm for detecting probability changes in dynamic systems. We have adapted CUSUM, an algorithm that can be used to detect changes in signal monitoring. We show that CUSUM can be used to detect when the probability to satisfy a given property drops below some value. This algorithm offers new possibilities to detect, e.g., emergent behaviors in dynamic systems. Our main contributions has been to extend temporal logic with a change-based operator.

All these SMC algorithms are implemented in PLASMA-Lab, and have been recently exported to MATLAB/Simulink – a widely used environment for modeling, simulating and analyzing multidomain dynamic systems – through an integration of MATLAB/Simulink and PLASMA-lab. This integration exploit MATLAB Control, a library allowing to interact with MATLAB from Java. We have developed two different methods to link the two environments. The first method includes a new plugin for PLASMA-lab that allows to load and execute Simulink models within PLASMA-lab, and therefore apply SMC algorithms to these models. The second method proposes an application that can be launched directly within MATLAB and provide the PLASMA-Lab SMC algorithms.

We have submitted a paper [41] that presents the new CUSUM algorithm and the integration between PLASMA-Lab and Simulink. In this paper, we apply these results to a case-study developed with Simulink that models a temperature controller of a pig shed. We show how to use PLASMA-Lab to check SMC requirements, perform parameters optimisation and detect failures in the model using the new CUSUM algorithm.

6.4.4. Papers

[48] (C) Statistical model checking (SMC) offers the potential to decide and quantify dynamical properties of models with intractably large state space, opening up the possibility to verify the performance of complex real-world systems. Rare properties and long simulations pose a challenge to this approach, so here we present a fast and compact statistical model checking platform, PLASMA, that incorporates an efficient simulation engine and uses importance sampling to reduce the number and length of simulations when properties are rare. For increased flexibility and ef-ficiency PLASMA compiles both model and property into bytecode that is executed on an in-built memory-efficient virtual machine.

[47] (C) We present PLASMA-lab, a statistical model checking (SMC) library that provides the functionality to create custom statistical model checkers based on arbitrary discrete event modelling languages. PLASMA-lab is written in Java for maximum cross-platform compatibility and has already been incorporated in various performance-critical software and embedded hardware platforms. Users need only implement a few simple methods in a simulator class to take advantage of our efficient SMC algorithms. PLASMA-lab may be instantiated from the command line or from within other software. We have constructed a graphical user interface (GUI) that exposes the functionality of PLASMA-lab and facilitates its use as a standalone application with multiple 'drop-in' modelling languages. The GUI adds the notion of projects and experiments, and implements a simple, practical means of distributing simulations using remote clients.

[41] (C; submitted) Statistical Model Checking (SMC) is a powerful and widely used approach that consists in extracting global information on the system by monitoring some of its executions. In this paper, we add two new stones to the cathedral of results on SMC, that are 1. a new algorithm to detect emergent behaviors at runtime, and 2. an integration of Plasma Lab, a powerful SMC checker, as a library of Simulink. Our results are illustrated on a realistic case study.

[26] (C) In this paper, we make use of the notion of a *score function* to improve the granularity of a logical property. We show that such a score function may take advantage of heuristics, so long as it also rigorously respects certain properties. To demonstrate our importance splitting approach we present an optimal adaptive importance splitting algorithm and an heuristic score function. We give experimental results that demonstrate a significant improvement in performance over alternative approaches.

[43] (C; submitted) We introduce feedback-control statistical system checking (FC-SSC), a new approach to statistical model checking that exploits principles of feedback-control for the analysis of cyber-physical systems (CPS). FC-SSC uses stochastic system identification to learn a CPS model, importance sampling to estimate the CPS state, and importance splitting to control the CPS so that the probability that the CPS satisfies a given property can be efficiently inferred. We illustrate the utility of FC-SSC on two example applications, each of which is simple enough to be easily understood, yet complex enough to exhibit all of FC-SCC's features. To the best of our knowledge, FC-SSC is the first statistical system checker to efficiently estimate the probability of rare events in realistic

CPS applications or in any complex probabilistic program whose model is either not available, or is infeasible to derive through static-analysis techniques.

6.5. Quantitative Reasoning

Participants: Axel Legay, Rudolf Fahrenberg, Louis-Marie Traonouez.

This part is concerned with Tasks 1 and 2. Mostly, we focus on quantifying properties of interconnected objects such as CPS (SoS and CPS share a lot of commonalities).

Model checking of systems deals with the question whether a given model of a computer system satisfies the properties one might want to require of it. This is a well-established and successful approach to formal verification of safety-critical computer systems.

When the models of the systems contain quantitative information, which is needed to represent the material on which the SoS is running, the model checking problem becomes complicated by the fact that in most cases, quantitative properties of the systems do not need to be satisfied exactly. Indeed, the model or the properties might be subject to measurement error, or probabilistic information might only be an approximation. In this case, it is of little use to know whether or not a model satisfies a specification precisely; what is needed instead is a notion of *satisfaction distance*: a measure which can assess to which extent a quantitative model satisfies a quantitative specification.

In other words, what is needed is a notion of satisfaction which is robust in the sense that small deviations in the model or the specification only lead to small changes in the outcome of the model checking question.

For reasoning about distributed systems or **systems-of-systems**, an important role is played by specification theories. Such systems are often far too complex to reason about, or model-check, as a whole, and additionally they might be composed of a large number of components which are implemented by different vendors. Hence one needs methods for compositional reasoning, which allow to infer properties of a system from properties of its components, and for incremental design, which allow to synthesize and refine specifications in a step-wise manner.

Such specification theories are by now well-established e.g. in the incarnations of interface theories and (disjunctive) modal transition systems. Additionally to defining a formalism for describing and model-checking specifications, they provide notions of refinement of specifications, logical conjunction of specifications, and structural composition and quotient.

When the models and specifications contain quantitative information, all the above notions need to be made robust. One needs to introduce a quantitative version of refinement, and the operations on specifications need to be continuous with respect to refinement distance: compositions of specifications with small refinement distance need themselves to have small refinement distance.

6.5.1. Theory papers:

[33] (J; submitted) There are two fundamentally different approaches to specifying and verifying properties of systems. The logical approach makes use of specifications given as formulae of temporal or modal logics and relies on efficient model checking algorithms; the behavioural approach exploits various equivalence or refinement checking methods, provided the specifications are given in the same formalism as implementations. In this paper we provide translations between the logical formalism of nu-calculus and the behavioural formalism of disjunctive modal transition systems. The translations preserve structural properties of the specification and allow us to perform logical operations on the behavioural specifications as well as behavioural compositions on logical formulae. The unification of both approaches provides additional methods for component-based stepwise design.

[4] (C) This paper studies a difference operator for stochastic systems whose specifications are represented by Abstract Probabilistic Automata (APAs). In the case refinement fails between two specifications, the target of this operator is to produce a specification APA that represents all witness PAs of this failure. Our contribution is an algorithm that allows to approximate the difference of two APAs with arbitrary precision. Our technique relies on new quantitative notions of distances between

APAs used to assess convergence of the approximations, as well as on an in-depth inspection of the refinement relation for APAs. The procedure is effective and not more complex to implement than refinement checking.

[21] (C) We provide a framework for compositional and iterative design and verification of systems with quantitative information, such as rewards, time or energy. It is based on disjunctive modal transition systems where we allow actions to bear various types of quantitative information. Throughout the design process the actions can be further refined and the information made more precise. We show how to compute the results of standard operations on the systems, including the quotient (residual), which has not been previously considered for quantitative non-deterministic systems. Our quantitative framework has close connections to the modal nu-calculus and is compositional with respect to general notions of distances between systems and the standard operations.

[35] (J; submitted) We provide a framework for compositional and iterative design and verification of systems with quantitative information, such as rewards, time or energy. It is based on disjunctive modal transition systems where we allow actions to bear various types of quantitative information. Throughout the design process the actions can be further refined and the information made more precise. We show how to compute the results of standard operations on the systems, including the quotient (residual), which has not been previously considered for quantitative non-deterministic systems. Our quantitative framework has close connections to the modal nu-calculus and is compositional with respect to general notions of distances between systems and the standard operations.

[6] (J) This paper proposes a new theory of quantitative specifications. It generalizes the notions of stepwise refinement and compositional design operations from the Boolean to an arbitrary quantitative setting. Using a great number of examples, it is shown that this general approach permits to unify many interesting quantitative approaches to system design.

[7] (J) We present a distance-agnostic approach to quantitative verification. Taking as input an unspecified distance on system traces, or executions, we develop a game-based framework which allows us to define a spectrum of different interesting system distances corresponding to the given trace distance. Thus we extend the classic linear-time–branching-time spectrum to a quantitative setting, parametrized by trace distance. We also prove a general transfer principle which allows us to transfer counterexamples from the qualitative to the quantitative setting, showing that all system distances are mutually topologically inequivalent.

[25] (C) We introduce a new notion of structural refinement, a sound abstraction of logical implication, for the modal nu-calculus. Using new translations between the modal nu-calculus and disjunctive modal transition systems, we show that these two specification formalisms are structurally equivalent.Using our translations, we also transfer the structural operations of composition and quotient from disjunctive modal transition systems to the modal nu-calculus. This shows that the modal nu-calculus supports composition and decomposition of specifications.

6.5.2. Application papers:

[32] (C; submitted) We suggest a method for measuring the degree to which features interact in feature-oriented software development. We argue that our method is practically feasible, easily extendable and useful from a developer's point of view.

[19] (C) Class diagrams are among the most popular modeling languages in industrial use. In a model-driven development process, class diagrams evolve, so it is important to be able to assess differences between revisions, as well as to propagate differences using suitable merge operations. Existing differencing and merging methods are mainly syntactic, concentrating on edit operations applied to model elements, or they are based on sampling: enumerating some examples of instances which characterize the difference between two diagrams. This paper presents the first known (to the best of our knowledge) automatic model merging and differencing operators supported by a formal semantic theory guaranteeing that they are semantically sound. All instances of the merge of a model and its difference with another model are automatically instances of the second model. The differences

we synthesize are represented using class diagram notation (not edits, or instances), which allows creation of a simple yet flexible algebra for diffing and merging. It also allows presenting changes comprehensively, in a notation already known to users.

[20] (C) We propose a new similarity measure between texts which, contrary to the current state-of-the-art approaches, takes a global view of the texts to be compared. We have implemented a tool to compute our textual distance and conducted experiments on several corpuses of texts. The experiments show that our methods can reliably identify different global types of texts.

[23] (C) Reliable model transformations are essential for agile modeling. We propose to employ a configurable-semantics approach to develop automatic model transformations which are correct by design and can be integrated smoothly into existing tools and work flows.

[39] (C; submitted) Nowadays, large software systems are mostly built using existing services. These are not always designed to interact, i.e., their public interfaces often present some mismatches. Checking compatibility of service interfaces allows one to avoid erroneous executions when composing the services and ensures correct reuse and interaction. Service compatibility has been intensively studied, in particular for discovery purposes, but most of existing approaches return a Boolean result. In this paper, we present a quantitative approach for measuring the compatibility degree of service interfaces. Our method is generic and flooding-based, and fully automated by a prototype tool.

6.5.3. Surveys:

[22] Modal transition systems provide a behavioral and compositional specification formalism for reactive systems. We survey two extensions of modal transition systems: parametric modal transition systems for specifications with parameters, and weighted modal transition systems for quantitative specifications.

[24] We survey extensions of modal transition systems to specification theories for probabilistic and timed systems.

6.6. Privacy and Security

Participants: Axel Legay, Fabrizio Biondi, Jean Quilbeuf, Thomas Given-Wilson.

6.6.1. Information-Theoretical Quantification of Security Properties

This part of the work was not foreseen at the beginning of the action. It concerns security aspects, and more precisely quantifying privacy of data. This aspect is in fact central for SoS and all our algorithms developed for Tasks 4 and 5 should be adapted to solve a series of problems linked to privacy in interconnected object and dynamical environment. For now, we only studied the foundations.

Information theory provides a powerful quantitative approach to measuring security and privacy properties of systems. By measuring the *information leakage* of a system security properties can be quantified, validated, or falsified. When security concerns are non-binary, information theoretic measures can quantify exactly how much information is leaked. The knowledge of such informations is strategic in the developments of component-based systems.

The quantitative information-theoretical approach to security models the correlation between the secret information of the system and the output that the system produces. Such output can be observed by the attacker, and the attacker tries to infer the value of the secret by combining this information with its knowledge of the system.

Armed with the produced output and the source code of the system, the attacker tries to infer the value of the secret. The quantitative analysis we implement computes with arbitrary precision the number of bits of the secret that the attacker will expectedly infer. This expected number of bits is the information leakage of the system.

The quantitative approach generalizes the qualitative approach and thus provides superior analysis. In particular, a system respects non-interference if and only if its leakage is equal to zero. In practice very few systems respect non-interference, and for those who don't it is imperative to be able to distinguish between the ones leaking a very small amount of bits and the ones leaking a significant amount of bits, since only the latter are considered to pose a security vulnerability to the system.

Since black box security analyzes are immediately invalidated whenever an attacker gains information about the source code of the system, we assume that the attacker has a white box view of the system, meaning that it has access to the system's source code. This approach is also consistent with the fact that many security protocol implementations are in fact open source.

The scope of modern software projects is too large to be analyzed manually. For this reason we provide tools that can support the analyst and locate security vulnerabilities in large codebases and projects. We work with a variety of tools, including commercial software analysis tools being adapted with our techniques, and tools such as QUAIL developed here by our team.

We applied the leakage analysis provided by QUAIL to several case studies. Our case studies (voting protocol and smart grid coordination) have in common that a publicly disclosed information is computed from the secret of every participant in the model. In the voting example, the vote of a given voter is secret, but the number of votes for each candidates is public. Similarly, in the smart grid example, the consumption of one of the houses is secret, but the consumption of a whole quarter can be deduced. Qualitative analyses are either too restrictive or too permissive on these types of systems. For instance, non-interference will reject them as the public information depends on the secret. Declassification approaches will accept them, even if the number of voters or consumers is 2, in which case the secret can be deduced.

The development of better tools for quantitative security builds upon both theoretical developments in information theory, and development of the tools themselves. These often progress in parallel with each supporting the findings of the other, and increasing the demands and understanding upon each other.

6.6.1.1. Papers:

[3] (J; submitted) The quantification of information leakage provides a quantitative evaluation of the security of a system. We propose the usage of Markovian processes to model deterministic and probabilistic systems. By using a methodology generalizing the lattice of information approach we model refined attackers capable to observe the internal behavior of the system, and quantify the information leakage of such systems. We also use our method to obtain an algorithm for the computation of channel capacity from our Markovian models. Finally, we show how to use the method to analyze timed and non-timed attacks on the Onion Routing protocol.

[46] (C) Quantitative security analysis evaluates and compares how effectively a system protects its secret data. We introduce QUAIL, the first tool able to perform an arbitrary-precision quantitative analysis of the security of a system depending on private information. QUAIL builds a Markov Chain model of the system's behavior as observed by an attacker, and computes the correlation between the system's observable output and the behavior depending on the private information, obtaining the expected amount of bits of the secret that the attacker will infer by observing the system. QUAIL is able to evaluate the safety of randomized protocols depending on secret data, allowing to verify a security protocol's effectiveness. We experiment with a few examples and show that QUAIL's security analysis is more accurate and revealing than results of other tools.

[40] (C; submitted) Quantitative security techniques have been proven effective to measure the security of systems against various types of attackers. However, such techniques are based on computing exponentially large channel matrices or Markov chains, making them impractical for large programs. We propose a different approach based on abstract trace analysis. By analyzing directly sets of execution traces of the program and computing security measures on the results, we are able to scale down the exponential cost of the problem. Also, we are able to appy statistical simulation techniques, allowing us to obtain significant results even without exploring the full space of traces. We have implemented the resulting algorithms in the QUAIL tool. We compare their effectiveness

against the state of the art LeakWatch tool on two case studies: privacy of user consumption in smart grid systems and anonymity of voters in different voting schemes.

[12] (C) In an election, it is imperative that the vote of the single voters remain anonymous and undisclosed. Alas, modern anonymity approaches acknowledge that there is an unavoidable leak of anonymity just by publishing data related to the secret, like the election's result. Information theory is applied to quantify this leak and ascertain that it remains below an acceptable threshold. We apply modern quantitative anonymity analysis techniques via the state-of-the-art QUAIL tool to the voting scenario. We consider different voting typologies and establish which are more effective in protecting the voter's privacy. We further demonstrate the effectiveness of the protocols in protecting the privacy of the single voters, deriving an important desirable property of protocols depending on composite secrets.

[13] (C) In recent years, quantitative security techniques have been providing effective measures of the security of a system against an attacker. Such techniques usually assume that the system produces a finite amount of observations based on a finite amount of secret bits and terminates, and the attack is based on these observations. By modeling systems with Markov chains, we are able to measure the effectiveness of attacks on non-terminating systems. Such systems do not necessarily produce a finite amount of output and are not necessarily based on a finite amount of secret bits. We provide characterizations and algorithms to define meaningful measures of security for non-terminating systems, and to compute them when possible. We also study the bounded versions of the problems, and show examples of non-terminating programs and how their effectiveness in protecting their secret can be measured.

7. Partnerships and Cooperations

7.1. Regional Initiatives

7.1.1. ESTASE

Participants: Axel Legay, Sean Sedwards.

ESTASE is a create project whose main objective was to initiate the creation of the plasma toolset as well as to propose new model checking algorithms for rare events.

7.1.2. Privacy

Participants: Axel Legay, Fabrizio Biondi, Jean Quilbeuf.

Privacy is a regional project whose objective is to quantify privacy of data. This includes, e.g., quantifying the anonymity of a voting protocol.

7.1.3. Variability

Participants: Axel Legay, Jin Hyun Kim, Louis-Marie Traonouez.

Variability is a regional project whose objective is to lift scheduling techniques to connected-objects. The main application of the project is Systems of Systems.

7.2. National Initiatives

7.2.1. ANR Malthy

Participants: Axel Legay, Rudolf Fahrenberg, Louis-Marie Traonouez.

The objective of this project is to study new models and techniques to reason on quantitative systems. We mainly focus on the composition of timed components in a dynamic setting.

7.2.2. BGLE SyS2Soft

Participants: Axel Legay, Thomas Given-Wilson, Cyrille Jegourel.

This national project studies various languages and techniques for quantitative systems.

7.3. European Initiatives

7.3.1. Danse

Program: FP7

Project acronym: DANSE

Project title: Designing for Adaptability and evolutioN in System of systems Engineering

Duration: mois année début - mois année fin

Coordinator: Offis

Abstract: Design and verification of Systems of Systems. We contributed by proposing the first verification engine for Heterogeneous SoS. For doing so, we have combined Plasma with Desyre that is a simulator for SoS described via the standardised FMI/FMU approach.

7.3.2. Meals

Program: Marie Curie

Project acronym: Meals

Project title: Mobility between Europe and Argentina applying Logics to Systems

Duration: Octobre 2012 – Octobre 2016

Coordinator: Germany (Saarbrucken) and Argentina ()

Abstract: Colaborative action on the topic of quantitative systems

7.3.3. Sensation

Program: Fet ProActif

Project acronym: Sensation

Project title: Self Energy-Supporting Autonomous Computation

Duration: Octobre 2012 – Octobre 2015

Coordinator: Aalborg University

Abstract: Development of new results for energy-centric systems. We contributed by proposing new algorithms for rare-event simulation.

7.3.4. DALI

Program: FP7

Project acronym: DALI

Project title: Devices for assisted living

Duration: Octobre 2011 - Octobre 2014

Coordinator: Trento University

Abstract: Development of a machine to guide a lady in a commercial center. We contributed by designing the cognitive algorithm. The machine is one example of a component of a large SoS that has its own objective but whose global behavior depends on those of other components. This is also a good illustration that our tool can be miniaturized to work in a small robot.

7.3.5. EMC2

Program: ARTEMIS

Project acronym: EMC2

Project title: Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments

Duration: mars 2014 – mars 2017

Coordinator: Infineon

Abstract: Large initiative on embadded systems and SoS. We will contribute with our expertise from DANSE and Sensation projects.

7.4. International Initiatives

Our team has strong collaboration with University of Namur, Carnegie Mellon University, University of Aalborg, Verimag Grenoble, and University of Waterloo. So far, those activities have not yet been funded.

7.5. International Research Visitors

7.5.1. Visits of International Scientists

7.5.1.1. Internships

- Jan Kretinsky, PostDoc at IST Austria
- Karin Quaas, PostDoc at Leipzig University
- Kim Larsen, Professor at Aalborg University
- Zoltan Esik, Professor at University of Szeged

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific events organisation

8.1.1.1. General chair, Scientific chair

- Axel Legay has been tutorial chair for the *European Joint Conferences on Theory and Practice of Software (ETAPS)*

8.1.1.2. Member of the organizing committee

- Axel Legay was a member of the organization committee for the *European Joint Conferences on Theory and Practice of Software (ETAPS)*

8.1.2. Scientific events selection

8.1.2.1. Responsable of the conference program committee

- Axel Legay has been the Program Chair for *Formal Modelling and Analysis of Timed Systems* conference
- Axel Legay has been the Program Chair for *From Programs to Systems – The Systems Perspective in Computing* workshop organized in honour of Joseph Sifakis
- Axel Legay has been the Program Chair for *Software Product Line Analysis Tools* workshop

8.1.2.2. Member of the conference program committee

- Axel Legay was PC member of MEMOCODE, FORMATS, FORMALIZE,SPLC,ASE,FACS,FORTE.
- Louis-Marie Traonouez was PC member of FORMATS
- Rudolf Fahrenberg was PC member of MFCS

8.1.3. Journal

8.1.3.1. Member of the editorial board

- Axel Legay is a member of the editorial board of the newly created journal for masterminding changes (FOMACS).

8.1.4. Teaching

Software Verification

Axel Legay: Software Verification, 40 hours, Royal Holloway, University of London

Axel Legay: Modélisation et Vérification Formelle par Automates, 12 hours, University of Rennes 1.

Rudolf Fahrenberg: Modélisation et Vérification Formelle par Automates, TP 12 hours, University of Rennes 1.

Louis-Marie Traonouez: Verification et Test des Systemes Embarques, 20 hours, ESIR Rennes.

8.1.5. Supervision

PhD : Cyrille Jegourel, Rare event techniques for Statistical Model checking, Novembre 2011, Axel Legay

8.1.6. Juries

Axel Legay has been a member of the PhD jury for Hoa Lee (Trento), Maxime Cordy (Namur), and Cyrille Jegourel (Rennes).

9. Bibliography

Publications of the year

Articles in International Peer-Reviewed Journals

[1] X. ALLAMIGEON, A. LEGAY, U. FAHRENBERG, R. KATZ, S. GAUBERT. *Tropical Fourier–Motzkin elimination, with an application to real-time verification*, in "International Journal of Algebra and Computation (IJAC)", 2014, vol. 24, n^{o} 5, pp. 569 - 607 [*DOI :* 10.1142/S0218196714500258], https://hal.inria.fr/hal-01087367

[2] S. BENSALEM, M. BOZGA, A. LEGAY, T.-H. NGUYEN, J. SIFAKIS, R. YAN. *Component-based Verification using Incremental design and Invariants*, in "Software and Systems Modeling", 2014, pp. 1-25 [*DOI :* 10.1007/s10270-014-0410-8], https://hal.inria.fr/hal-01087682

[3] F. BIONDI, A. LEGAY, P. MALACARIA, A. WĄSOWSKI. *Quantifying Information Leakage of Randomized Protocols*, in "Theoretical Computer Science", 2014, pp. 68 - 87 [*DOI :* 10.1007/978-3-642-35873-9_7], https://hal.inria.fr/hal-01088193

[4] B. DELAHAYE, U. FAHRENBERG, K. G. LARSEN, A. LEGAY. *Refinement and Difference for Probabilistic Automata*, in "Logical Methods in Computer Science", June 2014, pp. LMCS-2013-936, https://hal.archives-ouvertes.fr/hal-01010866

[5] B. DELAHAYE, K. G. LARSEN, A. LEGAY. *Stuttering for Abstract Probabilistic Automata*, in "Journal of Logic and Algebraic Programming", January 2014, pp. 1–19 [*DOI :* 10.1016/J.JLAP.2013.05.006], https://hal.archives-ouvertes.fr/hal-01084342

[6] U. FAHRENBERG, A. LEGAY. *General quantitative specification theories with modal transition systems*, in "Acta Informatica", 2014, pp. 261-295 [*DOI :* 10.1007/s00236-014-0196-8], https://hal.inria.fr/hal-01087314

[7] U. FAHRENBERG, A. LEGAY. *The quantitative linear-time–branching-time spectrum*, in "Journal of Theoretical Computer Science (TCS)", 2014, pp. 54-69 [*DOI :* 10.1016/J.TCS.2013.07.030], https://hal.inria.fr/hal-01087368

[8] J. B. FERREIRA FILHO, O. BARAIS, M. ACHER, J. LE NOIR, A. LEGAY, B. BAUDRY. *Generating Counterexamples of Model-based Software Product Lines*, in "Software Tools for Technology Transfer (STTT)", July 2014, https://hal.inria.fr/hal-01026581

[9] K. G. LARSEN, A. LEGAY, L.-M. TRAONOUEZ, A. WĄSOWSKI. *Robust Synthesis for Real Time Systems*, in "Journal of Theoretical Computer Science (TCS)", January 2014, vol. 515, pp. 96 - 122 [*DOI :* 10.1016/J.TCS.2013.08.015], https://hal.archives-ouvertes.fr/hal-01087778

[10] A. NOURI, S. BENSALEM, M. BOZGA, B. DELAHAYE, C. JEGOUREL, A. LEGAY. *Statistical model checking QoS properties of systems with SBIP*, in "International Journal on Software Tools for Technology Transfer", 2014, 14 p. [*DOI :* 10.1007/s10009-014-0313-6], https://hal.inria.fr/hal-01087822

Invited Conferences

[11] A. LEGAY, S. SEDWARDS. *On Statistical Model Checking with PLASMA*, in "The 8th International Symposium on Theoretical Aspects of Software Engineering", Changsha, China, IEEE, September 2014, https://hal.inria.fr/hal-01088859

International Conferences with Proceedings

[12] F. BIONDI, A. LEGAY. *Quantitative Anonymity Evaluation of Voting Protocols*, in "12th International Conference on Software Engineering and Formal Methods", Grenoble, France, September 2014, https://hal.inria.fr/hal-01088188

[13] F. BIONDI, A. LEGAY, B. F. NIELSEN, P. MALACARIA, A. WASOWSKI. *Information Leakage of Non-Terminating Processes* , in "IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science", Delhi, India, December 2014 [*DOI :* 10.4230/LIPICS.FSTTCS.2014.517], https://hal.inria.fr/hal-01086879

[14] B. BOYER, A. LEGAY, L.-M. TRAONOUEZ. *A Formalism for Stochastic Adaptive Systems*, in "Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications", Corfu, Greece, Lecture Notes in Computer Science, Springer, October 2014, vol. 8803, pp. 160 - 176 [*DOI :* 10.1007/978-3-662-45231-8_12], https://hal.archives-ouvertes.fr/hal-01087327

[15] M. CORDY, P. HEYMANS, A. LEGAY, P.-Y. SCHOBBENS, B. DAWAGNE, M. LEUCKER. *Counterexample Guided Abstraction Refinement of Product-Line Behavioural Models*, in "FSE 2014 : International Symposium on Foundations of Software Engineering", Hong Kong, Hong Kong SAR China, ACM, November 2014, pp. 190-201 [*DOI :* 10.1145/2635868.2635919], https://hal.inria.fr/hal-01087789

[16] B. DELAHAYE, J. L. FIADEIRO, A. LEGAY, A. LOPES. *Heterogeneous Timed Machines*, in "11th International Colloquium on Theoretical Aspects of Computing", Bucharest, France, September 2014, 18 p. , https://hal.archives-ouvertes.fr/hal-01010877

[17] X. DEVROEY, G. PERROUIN, M. CORDY, M. PAPADAKIS, A. LEGAY, P.-Y. SCHOBBENS. *A Variability Perspective of Mutation Analysis*, in "FSE 2014 : International Symposium on Foundations of Software Engineering", Hong Kong, Hong Kong SAR China, ACM, November 2014, pp. 841-844 [*DOI :* 10.1145/2635868.2666610], https://hal.inria.fr/hal-01087644

[18] X. DEVROEY, G. PERROUIN, M. CORDY, P.-Y. SCHOBBENS, A. LEGAY, P. HEYMANS. *Towards statistical prioritization for software product lines testing*, in "VAMOS", Nice, France, January 2014, pp. 1 - 7 [*DOI :* 10.1145/2556624.2556635], https://hal.inria.fr/hal-01092958

[19] U. FAHRENBERG, M. ACHER, A. LEGAY, A. WĄSOWSKI. *Sound Merging and Differencing for Class Diagrams*, in "FASE 2014 : 17th International Conference on Fundamental Approaches to Software Engineering", Grenoble, France, S. GNESI, A. RENSINK (editors), LNCS : Fundamental Approaches to Software Engineering, Springer, April 2014, vol. 8411, pp. 63 - 78 [*DOI :* 10.1007/978-3-642-54804-8_5], https://hal.inria.fr/hal-01087323

[20] U. FAHRENBERG, F. BIONDI, K. CORRE, C. JEGOUREL, S. KONGSHØJ, A. LEGAY. *Measuring Global Similarity between Texts*, in "SLSP 2014 : Second International Conference on Statistical Language and Speech Processing", Grenoble, France, Springer, October 2014, pp. 220-232 [*DOI :* 10.1007/978-3-319-11397-5_17], https://hal.inria.fr/hal-01087009

[21] U. FAHRENBERG, J. KŘETÍNSKÝ, A. LEGAY, L.-M. TRAONOUEZ. *Compositionality for Quantitative Specifications*, in "FACS", Bertinoro, Italy, September 2014, https://hal.inria.fr/hal-01087320

[22] U. FAHRENBERG, K. G. LARSEN, A. LEGAY, L.-M. TRAONOUEZ. *Parametric and Quantitative Extensions of Modal Transition Systems*, in "FPS@ETAPS", Grenoble, France, April 2014 [*DOI :* 10.1007/978-3-642-54848-2_6], https://hal.inria.fr/hal-01087363

[23] U. FAHRENBERG, A. LEGAY. *Configurable Formal Methods for Extreme Modeling*, in "XM@MoDELS", Valencia, Spain, September 2014, https://hal.inria.fr/hal-01087370

[24] U. FAHRENBERG, A. LEGAY, L.-M. TRAONOUEZ. *Specification Theories for Probabilistic and Real-Time Systems*, in "FPS@ETAPS", Grenoble, France, April 2014 [*DOI :* 10.1007/978-3-642-54848-2_7], https://hal.inria.fr/hal-01087364

[25] U. FAHRENBERG, A. LEGAY, L.-M. TRAONOUEZ. *Structural Refinement for the Modal nu-Calculus*, in "ICTAC", Bucarest, Romania, September 2014, pp. 169 - 187 [*DOI :* 10.1007/978-3-319-10882-7_11], https://hal.inria.fr/hal-01087295

[26] C. JEGOUREL, A. LEGAY, S. SEDWARDS. *An Effective Heuristic for Adaptive Importance Splitting in Statistical Model Checking*, in "International Symposium On Leveraging Applications of Formal Methods, Verification and Validation", Corfou, Greece, October 2014, pp. 143 - 159 [*DOI :* 10.1007/978-3-662-45231-8_11], https://hal.inria.fr/hal-01087828

[27] A. LEGAY, S. SEDWARDS. *Statistical Abstraction Boosts Design and Test Efficiency of Evolving Critical Systems*, in "Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change", Corfu, Greece, T. MARGARIA, B. STEFFEN (editors), Lecture Notes in Computer Science, EasyConference, October 2014, vol. 8802, pp. 4 - 25 [*DOI :* 10.1007/978-3-662-45234-9_2], https://hal.inria.fr/hal-01087858

[28] A. LEGAY, S. SEDWARDS, L.-M. TRAONOUEZ. *Scalable Verification of Markov Decision Processes*, in "4th Workshop on Formal Methods in the Development of Software (FMDS 2014)", Grenoble, France, Lecture Notes in Computer Science, September 2014, https://hal.inria.fr/hal-01088396

[29] S. NAUJOKAT, L.-M. TRAONOUEZ, M. ISBERNER, B. STEFFEN, A. LEGAY. *Domain-Specific Code Generator Modeling: A Case Study for Multi-faceted Concurrent Systems*, in "Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change", Corfu, Greece, Lecture Notes in Computer Science, Springer, October 2014, vol. 8802, pp. 481 - 498 [*DOI :* 10.1007/978-3-662-45234-9_33], https://hal.archives-ouvertes.fr/hal-01087322

[30] A. NOURI, M. BOZGA, A. MOLNOS, A. LEGAY, S. BENSALEM. *Building Faithful High-level Models and Performance Evaluation of Manycore Embedded Systems*, in "MEMOCODE", Lausanne, Switzerland, October 2014 [*DOI :* 10.1109/MEMCOD.2014.6961864], https://hal.inria.fr/hal-01087671

[31] A. NOURI, B. RAMAN, M. BOZGA, A. LEGAY, S. BENSALEM. *Faster Statistical Model Checking by Means of Abstraction and Learning*, in "RV", Toronto, Canada, September 2014 [*DOI :* 10.1007/978-3-319-11164-3_28], https://hal.inria.fr/hal-01087676

Research Reports

[32] J. M. ATLEE, U. FAHRENBERG, A. LEGAY. *Measuring Behaviour Interactions between Product-Line Features*, Inria Rennes, 2014, https://hal.inria.fr/hal-01088160

[33] N. BENEŠ, U. FAHRENBERG, J. KŘETÍNSKÝ, A. LEGAY, L.-M. TRAONOUEZ. *Logical vs. Behavioural Specifications*, Inria Rennes, 2014, https://hal.inria.fr/hal-01088150

[34] P. D'ARGENIO, A. LEGAY, S. SEDWARDS, L.-M. TRAONOUEZ. *Smart Sampling for Lightweight Verification of Markov Decision Processes*, Inria Rennes - Bretagne Atlantique, équipe ESTASYS, October 2014, Submitted to conference, https://hal.inria.fr/hal-01088633

[35] U. FAHRENBERG, J. KŘETÍNSKÝ, A. LEGAY, L.-M. TRAONOUEZ. *Compositionality for Quantitative Specifications*, Inria Rennes, 2014, https://hal.inria.fr/hal-01088154

[36] U. FAHRENBERG, A. LEGAY. *Homotopy Bisimilarity for Higher-Dimensional Automata*, Inria Rennes, September 2014, https://hal.inria.fr/hal-01087294

[37] A. LEGAY, S. SEDWARDS, L.-M. TRAONOUEZ. *Lightweight Verification of Markov Decision Processes with Rewards*, Inria Rennes - Bretagne Atlantique, October 2014, Submitted to conference, https://hal.inria.fr/hal-01088684

[38] V. C. NGO, A. LEGAY, J. QUILBEUF. *Dynamic Verification of SystemC with Statistical Model Checking*, Inria Rennes - Bretagne Atlantique, équipe ESTASYS, October 2014, n^{o} RR-8644, 25 p. , https://hal.inria.fr/hal-01089742

[39] M. OUEDERNI, U. FAHRENBERG, A. LEGAY, G. SALAÜN. *Flooding-Based Algorithm for Behavioural Compatibility Measuring*, Inria Rennes, 2014, https://hal.inria.fr/hal-01088157

Other Publications

[40] F. BIONDI, J. QUILBEUF, A. LEGAY. *Information Leakage by Trace Analysis in QUAIL*, November 2014, https://hal.inria.fr/hal-01088208

[41] B. BOYER, K. CORRE, A. LEGAY, L.-M. TRAONOUEZ. *Statistical Model Checking with Changes and Simulink*, 2014, https://hal.archives-ouvertes.fr/hal-01087821

[42] C. JEGOUREL. *Rare Event Simulation for Statistical Model Checking*, November 2014, https://hal.inria.fr/hal-01088479

[43] K. KALAJDZIC, C. JEGOUREL, E. BARTOCCI, A. LEGAY, S. SMOLKA, R. GROSU. *Model Checking as Control: Feedback Control for Statistical Model Checking of Cyber-Physical Systems*, November 2014, https://hal.inria.fr/hal-01087977

[44] Z. ÉSIK, U. FAHRENBERG, A. LEGAY. **-Continuous Kleene ω-Algebras*, December 2014, https://hal.inria.fr/hal-01100104

References in notes

[45] A. ARNOLD, B. BOYER, A. LEGAY. *Contracts and Behavioral Patterns for Systems of systems: The EU IP DANSE approach*, January 2013, 21 p. , https://hal.inria.fr/hal-00778039

[46] F. BIONDI, A. LEGAY, L.-M. TRAONOUEZ, A. WASOWSKI. *QUAIL: A Quantitative Security Analyzer for Imperative Code*, in "Computer Aided Verification - 25th International Conference", Saint Petersburg, Russia, Lecture Notes in Computer Science, Springer, July 2013, vol. 8044, pp. 702 - 707 [*DOI :* 10.1007/978-3-642-39799-8_49], https://hal.archives-ouvertes.fr/hal-01087804

[47] B. BOYER, K. CORRE, A. LEGAY, S. SEDWARDS. *PLASMA-lab: A Flexible, Distributable Statistical Model Checking Library*, in "Quantitative Evaluation of Systems", Buenos Aires, Argentina, K. JOSHI, M. SIEGLE, M. STOELINGA, P. R. D'ARGENIO (editors), Lecture Notes in Computer Science, August 2013, vol. 8054, pp. 160 - 164 [*DOI :* 10.1007/978-3-642-40196-1_12], https://hal.inria.fr/hal-01088411

[48] C. JEGOUREL, A. LEGAY, S. SEDWARDS. *A Platform for High Performance Statistical Model Checking – PLASMA*, in "TACAS 2012 - 18th International Conference Tools and Algorithms for the Construction and Analysis of Systems", Tallinn, Estonia, C. FLANAGAN, B. KÖNIG (editors), LNCS - Lecture Notes in Computer Science, Springer, March 2012, vol. 7214, pp. 498 - 503 [*DOI :* 10.1007/978-3-642-28756-5_37], https://hal.inria.fr/hal-01087824

[49] C. JEGOUREL, A. LEGAY, S. SEDWARDS. *Importance Splitting for Statistical Model Checking Rare Properties*, in "Computer Aided Verification", Saint-Pétersbourg, Russia, July 2013, pp. 576 - 591 [*DOI :* 10.1007/978-3-642-39799-8_38], https://hal.inria.fr/hal-01087826

[50] C. JÉGOUREL, A. LEGAY, S. SEDWARDS. *Cross-Entropy Optimisation of Importance Sampling Parameters for Statistical Model Checking*, in "Computer Aided Verification", Berkeley, United States, July 2012, pp. 327 - 342 [*DOI :* 10.1007/978-3-642-31424-7_26], https://hal.inria.fr/hal-01087341

[51] A. MIGNOGNA, L. MANGERUCA, B. BOYER, A. LEGAY, A. ARNOLD. *SoS contract verification using statistical model checking*, 2013, pp. 67 - 83, Workshop paper presenting the Toolchain to be produced by the DANSE project for modelling and verifying systems of systems [*DOI :* 10.4204/EPTCS.133.7], https://hal.inria.fr/hal-01090330