MLQA 2010: Models and Logics for Quantitative Analysis

An ERCIM Working Group

`http://wiki.ercim.eu/wg/MLQA`

Friday 9th July 2010, Edinburgh, Scotland

The ERCIM Working Group on Models and Logics for Quantitative Analysis (MLQA) is concerned with analysing **quantitative** properties, expressed in mathematical **logic**, of systems that are described using **process models**. MLQA 2010 is the second annual meeting of the working group, with the theme of:

Static Analysis versus Model Checking: similarities, differences, and synergies

The aim is to create a vibrant event consisting mainly of a number of invited talks that will cover some of the historical developments, survey the links established, establish state-of-the-art, identify the problems still worth pursuing and give a perspective on the implications and (novel) applications that can be foreseen. There will also be a poster session, in which participants will present current research projects, overviews of recent work, and future research challenges. Finally there will be a business meeting discussing the future of MLQA.

Flemming Nielson

MLQA Interim Chair

09:00–10:00: Session 1

9:00 *Bernhard Steffen*
From the How to the What: Static Analysis via Model Checking

Conceptually comparing the notions of Static Analysis and Model Checking is delicate. Where are the technical differences, in the use of logics, abstract interpretation, refinement techniques, fixpoint computation? Or rather in their pragmatics: the one is fast but often returns 'don't know', whereas the other may not terminate at all (cf. Patrick Cousot)? It is hard to agree on a borderline, and, in fact, earlier recongnized differences vanish as we go. This is a good sign, as it shows that syngery already happens here, as is also witnessed by the existence of conferences like VMCAI. The talk will therefore present a personal view, and a success story based on this understanding.

9:30 *Flemming Nielson*
Model Checking is Static Analysis of Modal Logic

Flow Logic is an approach to the static analysis of programs that has been developed for functional, imperative and object-oriented programming languages and for concurrent, distributed, mobile and cryptographic process calculi. It is often implemented using an efficient differential worklist based solver (the Succinct Solver) working on contraints presented in a stratified version of Alternation-free Least Fixed Point Logic (ALFP).

In this talk we show how to deal with modal logics; to be specific we show how to deal with modal logics in the families CTL, ACTL and alternation-free modal mu calculus. We prove that we obtain an exact characterisation of the semantics of formulae in the modal logics and that we remain within stratified ALFP. The computational complexity of model checking by means of static analysis is as for classical approaches to model checking.

Together with the work of Steffen et al this allows us to conclude that the problems of model checking and static analysis are reducible to each other in many cases. This provides further motivation for transferring methods and techniques between these two approaches to verifying and validating programs and systems.

This is joint work with Hanne Riis Nielson, Fyuyan Zhang and Piotr Filipiuk.

10:00–10:30: Coffee Break and Poster Session

10:30–12:30: Session 2

10:30 *Marta Kwiatkowska*
Quantitative Abstraction Refinement

Probabilistic model checking has established itself as a valuable technique for formal modelling and analysis of systems that exhibit stochastic behaviour. It has been used to study quantitative properties of a wide range of systems, from randomised communication protocols to biological signalling pathways. In practice, however, scalability quickly becomes a major issue and, for large or even infinite-state systems, abstraction is an essential tool. What is needed are automated and efficient methods for constructing such abstractions.

In non-probabilistic model checking, this is often done using counterexample-guided abstraction-refinement (CEGAR), which takes a simple, coarse abstraction and then repeatedly refines it until it is amenable to model checking. This talk describes recent and ongoing work on quantitative abstraction-refinement techniques, which can be used to automate the process of building abstractions for probabilistic models. This has already been applied to probabilistic verification of software and of real-time systems, where abstraction is essential.

11:00 *Joost-Pieter Katoen*
Invariant Generation for Probabilistic Programs

Model checking probabilistic programs, typically represented as Markov decision processes, is en vogue. Abstraction-refinement techniques (a la CEGAR) have been developed and parametric model checking approaches have recently been suggested. Prototypical tools support these techniques. Their usage for programs whose invariants are quantitative, i.e. arithmetic expressions in the program variables, is however limited. An alternative is to resort to static analysis techniques.

We present a constraint-based method for automatically generating quantitative invariants for probabilistic programs. We show how it can be used, in combination with proof-based methods, to verify properties of probabilistic programs that cannot be analysed by any of currently existing probabilistic model checkers. To our knowledge, this is the first automated method for quantitative-invariant generation.

(Joint work with Larissa Meinicke, Annabelle McIver and Carroll Morgan.)

11:30 *Marsha Chechik and Arie Gurfinkel*
Partial Models and Software Model-Checking

Partial models combine necessary and possible behaviours in a single model. Since they encode both over- and under-approximation of the behavior of the underlying system into a single model, they allow both verification and falsification of a broad class of properties. Thus, they are a natural choice for abstraction in model-checking.

Yasm is the first symbolic software model-checker to integrate partial models with the Counterexample-Guided Abstraction Refinement (CEGAR) framework. Yasm can prove and disprove temporal logic properties with equal effectiveness, and its performance is comparable to standard over-approximating model-checkers.

In this talk, we describe the types of partial models used by Yasm and highlight some of our recent developments in software model-checking, including proving non-termination (i.e. finding counterexamples to liveness properties) and reasoning about recursive programs.

12:00 *Orna Grumberg*
The 2-Valued and the 3-Valued Abstraction-Refinement Frameworks in Model Checking

Model checking is a widely used technique for automatic verification of hardware and software systems. Significant amount of research in model checking is devoted to extending its scope to larger and even infinite-state systems. Abstraction is one of the most successful approaches for doing so. It hides some of the irrelevant details of the system, thus resulting in a small (abstract) model whose correct behavior can be checked. Sometimes the abstraction is too coarse and a desired property cannot be checked on the abstract model. In this case, the abstract model is refined by adding some of the hidden details back.

In this talk we present two different frameworks for abstraction-refinement in model checking: The 2-valued (CEGAR) and the 3-valued (TVAR) frameworks. We will describe the abstract models and the type of properties that can be verified with those abstractions. We will also show when refinement is needed. Finally, we will mention some applications of TVAR and try to convince that it is most useful.

12:30–14:00: Lunch and Poster Session

14:00–15:00: Session 3

14:00 *David Monniaux*
Policy Iteration for Static Analysis

Static analysis by abstract interpretation has traditionally computed over-approximations of least fixed points (loops or recursive functions) using Kleene iteration accelerated by widening operators. In recent years, techniques inspired from game theory have been applied to fixpoint problems in certain abstract domains: min-policy iterations (E. Goubault's group) and max-policy iterations (H. Seidl's group). We shall outline these techniques.

Finally, we shall discuss recent results, obtained with T. Gawlitza, about the computation of least fixed points from succinct representations of programs, giving the same precision as explicitly distinguishing all program paths but with the same exponential complexity as the coarse analysis that merges program paths (a naive approach would incur double exponential complexity). The succinct representation is exploited through SMT-solving.

We would appreciate insights for possible applications of such techniques to succinct representations of e.g. probabilistic systems.

14:30 *Michael Huth*
From Validating Quantitative Models to Generating Valid Ones

Verification techniques for quantitative systems typically require a system model as object of their investigations. Probabilistic model checking is a representative example of this successful approach.

However, emerging needs and constraints of quantitative systems no longer allow for, or benefit from, a complete decoupling of the development of a system from its a posteriori verification.

We paint a somewhat personal picture of what this may mean in terms of challenges and opportunities for those who research the construction of valid quantitative systems.

15:00–15:30: Coffee Break and Poster Session

15:30–17:00: MLQA Business Meeting

About the future of MLQA and future research collaborations. Open to all attendees.

MLQA 2010: Poster Session

Friday 9th July 2010, Edinburgh, Scotland

As part of the second annual meeting of the ERCIM working group on Models and Logics for Quantitative Analysis, we sent out a call for posters that would report on recent research results in the field, and inspire future research directions. We were delighted to receive such a wide range of submissions, illustrating the breadth of research taking place across the MLQA member institutions. Furthermore, the many themes that are shared amongst the posters illustrate precisely the synergies that MLQA was set up to foster.

We would like to thank all the authors, and we hope that you will enjoy reading and discussing their posters.

Nataliya Skrypnyuk Michael Smith

MLQA 2010 poster session organisers

p-Automata Approach to Probabilistic Verification

Michael Huth, Nir Piterman and Daniel Wagner

The poster introduces an automata based approach to the verification of probabilistic systems. Our so-called p-automata combine the combinatorial structure of alternating tree automata with the ability to quantify probabilities of regular sets of paths. These foundations enable abstraction-based probabilistic model checking for probabilistic specifications that subsume Markov chains, PCTL, LTL and CTL* like logics.

The poster is based on results from "New Foundations for Discrete-Time Probabilistic Verification" to appear in Proceedings of QEST 2010.

Abstractions of Stochastic Process Algebras

Nataliya Skrypnyuk and Michael Smith

Stochastic process algebras are compositional modelling languages that can be used to describe systems whose behaviour evolves probabilistically over time — from distributed computer systems, to signalling pathways in cells. Two such languages are Interactive Markov Chains (IMC), and the Performance Evaluation Process Algebra (PEPA), which handle synchronisation between model components in different ways. The semantics of PEPA can be described as a continuous time Markov chain (CTMC), whereas for IMC it is a continuous time Markov decision process (CTMDP).

In both PEPA and IMC, it is very easy to write a model whose state space is exponentially larger than its description. It is therefore important to look for ways to abstract such models, so that they can become small enough to analyse. Importantly, we want to perform the abstraction at the language level, so that we maintain the compact description of the model. In this poster, we illustrate two different approaches to this — abstraction of IMC using pathway analysis (a type of static analysis), and abstraction of PEPA using compositional aggregation of states.

Quantitative Assessment of Web Services: Applying SCOWS to a Real-Life Scenario

Igor Cappello and Paola Quaglia

We present an example of application of SCOWS to a real-life scenario (loan request) developed within the SENSORIA European Project. SCOWS is a stochastic process algebra tailored for the purpose of quantitative assessment of Web Services: applying the labelled semantics to a SCOWS service S, it is possible to derive the Labelled Transition System (LTS) representing all the possible behaviours of S. The LTS can then be used to generate a CTMC on which quantitative model checking can take place.

In order to automatically derive the LTS and minimize the state space needed to represent it, we implemented SCOWS_lts, a tool written in Java whose output can be imported in a model checker (e.g. PRISM) and used to perform quantitative analysis. We report some results obtained assessing the impact of one parameter (the rate at which a clerk processes the submitted loan request) on the overall performance of the service.

High Security at a Low Cost

Ender Yüksel, Hanne Riis Nielson and Flemming Nielson

In the future tiny devices with microcontrollers and sensors will be in charge of numerous activities in our lives. Tracking our energy consumption and CO_2 emission, controlling our living conditions, enforcing security, and monitoring our health will be some examples of their functions. These devices will form wireless networks to communicate with one another, moreover their power consumption will be very low. It is not hard to predict that our modern society will depend on the correct operation of these devices, and the security of the network they are operating.

Such sensor-based systems, also known as "cyber-physical systems", achieve security by means of cryptographic protocols. In a simplistic setting where the power consumption should be minimum and the processing power is limited, it is more likely that all devices in the network will share the same cryptographic key. In this study, we are working on the trade-off between two challenges: "the cryptographic key should be changed frequently to preserve security" and "the cryptographic key should be changed rarely to save power." We work on the ZigBee wireless sensor network standard, that offers the advantages of simple and low resource communication. We model the system as a continuous-time Markov chain, and analyze it by posing a number of questions shedding light on its behaviour. The properties we are interested in are expressed in continuous stochastic logic, and probabilistic model checker Prism is used in the analysis.

Towards Dynamic Adaptation of Probabilistic Systems

Erik de Vink, Suzana Andova and Luuk Groenewegen

Dynamic system adaptation is modeled in Paradigm as coordination of collaborating components. A special component McPal allows for addition of new behavior, of new constraints and of new control in view of a new collaboration. McPal gradually adapts the system dynamics. It is shown that the approach also applies to the probabilistic setting. For a client-server example, where McPal gradually adds probabilistic behavior to deterministic components, precise modeling of changing system dynamics is achieved. This modeling of the transient behavior, spanning the complete migration range from as-is collaboration to to-be collaboration, serves as a stepping stone for quantitative analysis of the system during adaptation.

Model Checking is Static Analysis

Fuyuan Zhang and Piotr Filipiuk

The link between model checking and static analysis has been of great interest to academia for many years. Early work of Bernhard Steffen and David Schmidt has shown that classic data-flow analysis is an instance of model checking. The poster gives an overview of our research aiming to show that also model checking is static analysis of modal logics and that model checking can be carried by the static analysis engine; namely ALFP suite. We use Flow Logic, which is a state-of-the-art approach to static analysis that bridges the gap between a number of approaches to static analysis including Data Flow Analysis, Constraint Based Analysis, Abstract Interpretation and Type and Effect Systems. In the developments it has been demonstrated that Flow Logic is a robust approach that is able to deal with a variety of calculi, programming languages and recently modal logics such as CTL, ACTL and alternation-free mu-calculus. In order to calculate the analysis solution, we use ALFP Suite that computes the least model guaranteed by the Moore family theorem for ALFP. In its current version, the suite supports two solvers, one being a differential worklist solver that is based on a representation of relations as prefix trees and the other being a solver in continuation passing style that is based on a BDD representation of relations.

A Stochastic Hybrid Process Algebra

Vashti Galpin, Jane Hillston and Luca Bortolussi

The process algebra HYPE was developed for the modelling of systems with discrete and continuous behaviour. We now extend HYPE with stochastic behaviour to broaden its applicability. In HYPE, events are categorised as urgent which means they must occur as soon as the condition governing them becomes true; or as non-urgent meaning that they happen at some unspecified point in the future. In stochastic HYPE, we remove the second option and introduce stochastic events which have associated exponential distributions. HYPE models can be interpreted as hybrid automata, whereas stochastic HYPE models require a more expressive semantics, namely Transition Driven Stochastic Hybrid Automata, a subset of Piecewise Deterministic Markov Processes.

We illustrate the use of stochastic HYPE by modelling delay-tolerant networks. These networks do not always have end-to-end connectivity and hence nodes in the network need additional storage for packets (bundles in delay-tolerant network terminology) that cannot be forwarded due to lack of connectivity. We describe individual buffer components made up of input and output subcomponents, and compose these to form the network model. Simulations illustrate the effect of different buffer sizes on the number of dropped packets.

A Linear Operator Framework for Analysing Resource Usage

David Cachera, Thomas Jensen, Arnaud Jobin and Pascal Sotin

We present a semantics-based framework for analysing the quantitative behaviour of programs with regard to resource usage. We start from an operational semantics in which costs are modelled using a dioid structure, which generalizes the classical (max, plus) semiring structure over reals. The dioid structure of costs allows for defining the quantitative semantics as a linear operator. We then develop an approximation framework of such a semantics in order to effectively compute global cost information from the program. We show how this framework is related to, and thus can benefit from, the theory of abstract interpretation for analyzing qualitative properties, We focus on the notion of long-run cost which models the asymptotic average cost of a program, and show that our abstraction technique provides a correct approximation of the concrete long-run cost. We illustrate our approach on an application example devoted to the computation of energy consumption for a language with complex array operations and explicit energy modes management.

Verification of Continuous Dynamical Systems by Timed Automata

Christoffer Sloth and Rafael Wisniewski

The poster outlines a method for abstracting continuous dynamical systems by finite timed automata, which allows automatic verification of continuous systems. The novelty of the method is that the abstraction is generated from a partition of state space which is generated by intersecting set-differences of positive invariant sets.

It is chosen to abstract continuous systems by timed automata, since tools for efficient verification of such models exist. Additionally, Lyapunov functions are chosen for generating the partition, because their sub-level sets are positive invariant; hence, it is possible to determine a priori if the abstraction is sound or complete. Furthermore, for certain systems it is possible to approximate the reachable set of the continuous system arbitrarily close by the timed automaton. The structure added to the abstraction by partitioning the state space by positive invariant sets allows the timed automaton to be composed of multiple timed automata. This makes it possible to parallelize the verification process of the timed automaton.

Some methods require explicit solutions of the differential equations that describe the continuous dynamics, which are usually unknown. Therefore, the proposed method only relies on the Lyapunov functions and their derivatives, which can be calculated. The proposed abstraction is applied to an example, which illustrates how sound and complete abstractions are generated.

A Broadcast Based Stochastic Calculus

Lei Song, Flemming Nielson and Bo Friis Nielson

In our current work we give a stochastic calculus based on broadcast communication. Only broadcast actions are associated with rates while all the reception actions are passive and assigned with weights. By doing so there is no need to handle synchronisation between actions with different rates. We also give a label transition system of our calculus from which we can get a CTMC naturally. We show the usefulness of our calculus by giving several examples from performance analysis setting such as batch Markovian arrival process (BMAP), marked markovian arrival process (MMAP) and closed queueing networks.

Tailoring the Shape Calculus for Quantitative Analysis

Massimo Callisto De Donato, Flavio Corradini, Maria Rita Di Berardini, Emanuela Merelli and Luca Tesei

The Shape Calculus, a bio-inspired language, is a non-deterministic calculus in which processes are composed of a 3D shape moving in a space and of a behaviour. A process behaviour is formally expressed as a CCS-like process algebra with deterministic time where channels — the active sites — are particular surfaces of the shape. A site can bind, after a proper collision, to another compatible channel forming a more complex shape and a new process. Contextually to the introduction of the Shape Calculus we defined BioShape, an environment in which a network of Shape Calculus processes, together with more specific information about motion and behaviour, can be simulated. In particular, we showed that such a tool is suitable for representing a biological phenomenon at different spatial and temporal scales and can be used to perform in silico experiments.

Our aim is to complete the Shape Calculus by providing verification techniques at specification level. The objective is to find the right abstractions and boundaries that permit the application of existing quantitative model checking or quantitative equivalence checking techniques to the evolution of a given network of Shape Calculus processes. Also suitable logic languages for specifying the properties must be identified. In such a way we target to improve Shape Calculus to promote a coupling between the calculus and the BioShape tool to work in synergy towards the gaining of quantitative information about simulated environments.

MLQA: Mission Statement

Models and Logics for Quantitative Analysis

The ERCIM Working Group on Models and Logics for Quantitative Analysis (MLQA) is concerned with analysing **quantitative** properties, expressed in mathematical **logic**, of systems that are described using **process models**. The focus of MLQA is specifically in four major areas:

1. **Process models** — formally described by transition systems, automata, or process calculi.
2. **Logics** — for expressing discrete, stochastic, and continuous properties.
3. **Algorithms, theory and tools**.
4. **Applications** — with a particular emphasis on embedded systems and service oriented systems, but also considering IT-guided workflow systems and biological systems.

The goal of MLQA is to create a venue for knowledge sharing in this exciting area, and a network for young researchers. In doing so we aim to share tools developed within the field, to discuss research directions, and to eventually to formulate a European project or network on formal quantitative analysis.

1 Research Topics

1.1 Non-Technical Mission Statement

IT Systems

A large fraction of contemporary Information Technology systems are either *Embedded Systems* (offering autonomous and intelligent control of complex physical systems) or *Service Oriented Architectures* (providing web services designed to support Machine to Machine interaction over a network). This tendency will greatly increase in what is going to become the *Internet of the Future* — an integrated system comprising telecommunication, the Internet, and small systems embedded in domestic applicances. Cutting edge examples include intelligent vehicles that actively prevent accidents, intelligent homes that actively support your lifestyle, and services for handling electronic shopping and secure payments. On a larger scale, the future integration of medical equipment, emergency support systems, electronic hospital records, and next generation communication technologies are all examples that point towards the trend of *Service Oriented Systems* incorporating a number of *Embedded Components*. On an even larger scale, we begin to see *IT-Guided Workflow Systems* where the human activities are mainly those of domain experts (e.g. a doctor who is an expert in a given treatment) rather than being in charge of the overall workflow (e.g. monitoring the treatment history from the point of view of the patient). Going outside of the traditional domains of IT Systems there is also a growing use of computer science modelling and analysis techniques within the *Life Sciences* — in particular the modelling of components of Biological Systems.

Properties

The stability of the IT infrastructure of our future society demands that a number of fundamental properties can be validated for the IT Systems of interest. This spans properties related to *security* (e.g. 'no virus can allow outsiders to get access to my Internet bank'), to *performance / dependability* (e.g. 'my critical Internet service will be available 99.99% of the time'), and to *resource usage* (e.g. 'the control system rotates and adjusts the windmill such that at least 60% of the potential wind energy

is utilised'). Even the formulation of these properties becomes non-trivial when addressing IT-Guided Workflow Systems that have humans as 'subsystems', and when addressing descriptions of the three dimensional behaviour of Biological Systems.

The Challenge

Due to their interaction with the surrounding, physical environment, the modelling and validation of embedded and service oriented systems must include *discrete* (e.g. providing security guarantees), *stochastic* (e.g. dealing with performance) as well as *continuous* aspects (e.g. providing measurements of resource usage). A shift in paradigm from the study of *discrete* properties is required to develop IT Systems that also require *stochastic* and *continuous* properties — not least when addressing IT-Guided Workflow Systems. The use of *stochastic* and *continuous* properties is equally important in the domain of Life Sciences.

Objectives

To meet the above challenge we need powerful modelling methods and algorithms for the analysis of discrete, stochastic and continuous properties. The aim of this working group is to create a venue for knowledge sharing in this exciting area, and also to create a network for young researchers. Furthermore, we aim to share tools for performing analyses, and to create joint European research projects on quantitative analysis.

1.2 Technical Mission Statement

Models of IT Systems

The construction of IT Systems spans a large number of abstraction levels ranging from low-level, hardware-oriented programming languages (e.g. VHDL), to high-level programming languages (e.g. C++ and Java), and object-oriented development notations (e.g. UML).

To ensure applicability at all levels, and independently of concrete programming languages, we will focus on modelling system behaviour by means of *process models* expressed using process calculi, transition systems, or automata. These are able to model a wide variety of discrete and stochastic features, although further development is needed to fully account for the continuous ones. The study of open systems is well researched, but needs to be extended in the case of both IT-Guided Systems, where human components cannot be fully described, and Biological Systems, where components need to be understood in three dimensions.

Specifications of Properties

To ensure the quality of systems, the *Safety Instrumented Systems* standard is pervasive in the area of embedded systems, and the *Common Criteria* standard is widely used in many NATO countries; they are examples of international standards that emphasise the need for validating that systems are

- *functionally correct* (react as expected),
- *safe* (do not cause damage on environment or users),
- *highly efficient* while demanding few resources,
- *secure* against hackers and viruses,
- *stable* (do not crash),
- *fault tolerant* (offer vital functionality even when partially crashed).

To ensure a uniform approach to studying these important properties, we will base our work on *logical specification formalisms*. This allows us to accommodate seemingly dissimilar properties within the same formalism, and facilitates the construction of automatic validation engines. These formalisms are able to

express all discrete properties and a good selection of stochastic and continuous ones. They have been widely used for analysing Embedded Systems and Service Oriented Architectures but may need to be developed further to fully deal with IT-Guided Workflow Systems and Biological Systems.

The Challenge

Whilst more work is needed to ensure that models and specifications can *express* phenomena of interest, the main challenge is to provide effective *methods*, *techniques*, and *algorithms* for verifying models of IT Systems against specifications of properties. New challenges may arise from the study of IT-Guided Systems, where human components by their very nature cannot be fully described, and in describing the multi-dimensional nature of Biological Systems.

We will now take a closer look at the current state-of-the-art and future challenges within the discrete, stochastic, and continuous dimensions respectively.

Discrete Models and Properties

The Computer Science approach to IT Systems has traditionally been based on discrete models that consciously abstract away from physical phenomena that are less relevant — for example, the discrete model of a finite automaton is often a useful abstraction of an electronic device. *Static analysis* and *model checking* are two of the most prominent techniques for analysing discrete systems. In many ways they are complementary, and they are largely developed by independent research communities. In 2007 the Turing Award was given to Clarke, Emerson and Sifakis for their work on model checking. The techniques are used by some of the largest international companies (e.g. IBM, Intel, Microsoft). A related technique is *theorem proving*, which may provide less automation but is often able to deal with stronger properties.

The scientists involved in this working group are reputed for their contributions within these areas, and want to exploit this unique position in creating new combined analysis techniques that are more powerful than seen before.

Stochastic Models and Properties

Quantitative properties of the environment of a given IT System are often accompanied by uncertainties that are best described using stochastic or probabilistic models — for example, *Markov Chains*, *Markov Decision Processes*, and *Continuous Time Markov Decision Processes*. There has been significant research in probabilistic and stochastic process models over the last 10–15 years, with exciting developments towards modelling *Biological Systems* over the last 5–10 years.

Within a number of European projects, the scientists involved in this working group have contributed significantly to model checking methods for different types of Markov model (Discrete / Continuous Time Markov Chains, and Markov Decision Processes). The working group will focus on extensions to models that combine stochastic and continuous aspects, and will develop — so far non-existing — static analysis techniques that are applicable to stochastic models.

From the point of view of *"traditional" mathematical modelling*, this working group offers a unique chance to integrate and further develop recent advances in stochastic models. This involves techniques for dealing with the joint distribution of distinct properties of interest. An important example would be the joint distribution of the time and energy required to complete a certain task.

Continuous Models and Properties

Within classical Control Theory, the predominant way of modelling an IT System is through a set of *differential equations*, which describe the evolution of physical phenomena in the environment, when regulated by a given control program. A serious restriction is that this only allows for systems with completely deterministic behaviour to be modelled, making it impossible to handle distributed computations. To overcome this restriction, the new area of *Hybrid Systems* has emerged in the intersection between Computer Science and Control Theory.

The scientists involved in this working group will concentrate on the challenge of defining property-preserving transformations from hybrid system models to discrete models, so that the powerful analysis

techniques for discrete systems may be applied. We will also devote effort to defining — so far mainly absent — logical specification formalisms and analysis methods.

From the point of view of *"traditional" mathematical modelling*, this approach offers a unique chance to develop tractable ways of dealing with important control system properties, such as reachability (the ability of a system to attain a specified goal) and stability (the ability to maintain the goal in spite of disturbances), which are currently beyond the state-of-the-art. A challenging and *crosscutting* combination of stochastic and continuous models is the study of *stochastic differential equations* — an area known as Stochastic Control Theory. Here, the challenge is to find an adequate stochastic process, which realistically models the uncertainty inherent in the given problem, and to develop the corresponding algorithms.

2 Membership

At present the following institutions, which are members of the national organisations taking part in ERCIM, support the MLQA working group (in alphabetical order and just one contact person per site):

- CNR (Italy) / *Diego Latella*
- CWI (Holland) / *Jan Rutten*
- INRIA (France) / *Catuscia Palamidessi*
- Oxford University (England) / *Marta Kwiatkowska*
- RWTH Aachen University (Germany) / *Joost-Pieter Katoen*
- Technical University of Berlin (Germany) / *Uwe Nestmann*
- Technical University of Denmark / *Flemming Nielson*
- Trinity College (Ireland) / *Matthew Hennessy*
- Università di Firenze (Italy) / *Rocco De Nicola*
- University of Trento (Italy) / *Paola Quaglia*
- University of Edinburgh (Scotland) / *Stephen Gilmore*
- Universität des Saarlandes (Germany) / *Holger Hermanns*
- University of Aalborg (Denmark) / *Kim Guldstrand Larsen*

Individual membership is open to all researchers; please follow the procedure outlined at:

`http://wiki.ercim.eu/wg/MLQA/index.php/Contact`