INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team MExICo

Modelling and Exploitation of Interaction and Concurrency

Saclay - Île-de-France

Theme : Programs, Verification and Proofs

Activity Report

2009

Table of contents

MExICo is a joint project with CNRS and Ecole Normale Supérieure of Cachan. The team has been created on March 1st, 2009 and is currently being evaluated in view of becoming an INRIA project.

1. Team

Research Scientist

Benedikt Bollig [Research Associate (CR) CNRS]
Stefan Haar [Team leader, Research Associate (CR) Inria, HdR]
Marc Zeitoun [From Sept 1st 2009, Professor on leave from Bordeaux, HdR]

Faculty Member

Thomas Chatain [Maître de Conférence ENS Cachan]
Paul Gastin [Professor ENS Cachan, HdR]
Serge Haddad [Professor ENS Cachan, HdR]
Stefan Schwoon [Maître de Conférence ENS Cachan, Chaire INRIA]

PhD Student

Sandie Balaguer [since November 2009]
Hilal Djafri [since October 2008]
Dorsaf El Hog [since October 2008]
Akshay Sundararaman [co-supervised with CMI, India; defence spring 2010]
Nathalie Sznajder [thesis defended on November 12, 2009]

Administrative Assistant

Isabelle Biercewicz [Secretary (SAR) Inria]

2. Overall Objectives

2.1. Scientific Objectives

2.1.1. Introduction

In the increasingly networked world, reliability of applications becomes ever more critical as the number of users of, e.g., communication systems, web services, transportation etc grows steadily. Management of networked systems, in a very general sense of the term, is therefore an even more crucial task, but also an even more difficult one.

MExICo strives to take advantage of distribution by orchestrating cooperation between different agents that observe local subsystems, and interact in a localized fashion.

The need for applying formal methods in the analysis and management of complex systems has long been recognized. It is with much less unanimity that the scientific community embraces methods based on asynchronous and distributed models. Centralized and sequential modeling still prevails.

However, we observe that crucial applications have increasing numbers of users, that networks providing services grow fast both in the number of participants and the physical size and degree of spatial distribution. Moreover, traditional *isolated* and *proprietary* software products for local systems are no longer typical for emerging applications.

In contrast to traditional centralized and sequential machinery for which purely functional specifications are efficient,Instead, applications are provided from diverse and non-coordinated sources, and their distribution (e.g. over the Web) must change the way we verify and manage them; in particular, one cannot ignore the impact of quantitative features such as delays or failure likelihoods on the functionalities of composite services in distributed systems.

We thus identify three main characteristics of complex distributed systems that constitute research challenges:

- *Concurrency* of behavior;
- *Interaction* of diverse and semi-transparent components; and
- management of *quantitative* aspects of behavior.

2.1.2. Concurrency

The increasing size and the networked nature of communication systems, controls, distributed services, etc confront us with an ever higher degree of parallelism between local processes. This field of application for our work includes telecommunication systems and composite web services. The challenge is to provide sound theoretical foundations and efficient algorithms for management of such systems, ranging from controller synthesis to fault diagnosis to integration and adaptation. While these tasks have received considerable attention in the *sequential* setting, managing *non-sequential* behavior requires profound modifications for existing approaches, and often the development of new approaches altogether. We see concurrency in distributed systems as an opportunity rather than a nuisance. Our goal is to *exploit* asynchronicity and distribution as an advantage. Clever use of adequate models, in particular *partial order semantics* (ranging from Mazurkiewicz traces to event structures to MSCs) actually helps in practice. In fact, the partial order vision allows us to make causal precedence relations explicit, and to perform diagnosis and test for the dependency between events. This is a conceptual advantage that interleaving-based approaches cannot match. The two key features of our work will be *(i)* the exploitation of concurrency by using asynchronous models with partial order semantics, and *(ii)* distribution of the agents performing management tasks.

2.1.3. Interaction

Systems and services exhibit non-trivial *interaction* between specialized and heterogeneous components. This interplay is challenging for several reasons. On one hand, a coordinated interplay of several components is required, though each has only a limited, partial view of the system's configuration. We refer to this problem as *distributed synthesis* or *distributed control*. An aggravating factor is that the structure of a component might be semi-transparent, which requires a form of *grey box management*.

2.1.4. Quantitative Features

Besides the logical functionalities of programs, the *quantitative* aspects of component behavior and interaction play an increasingly important role.

- *Real-time* properties cannot be neglected even if time is not an explicit functional issue, since transmission delays, parallelism, etc, can lead to time-outs striking, and thus change even the logical course of processes. Again, this phenomenon arises in telecommunications and web services, but also in transport systems.
- In the same contexts, *probabilities* need to be taken into account, for many diverse reasons such as unpredictable functionalities, or because the outcome of a computation may be governed by race conditions.
- Last but not least, constraints on *cost* cannot be ignored, be it in terms of money or any other limited resource, such as memory space or available CPU time.

2.2. Highlights

- In the fascinating and rather old domain of distributed synthesis where significative steps forwards are difficult to obtain, Nathalie Sznajder has defined a natural framework based on signal communication between processes, and shown decidability of the synthesis problem for the class of architectures with strongly connected communication graph. The results are presented in the publications [17], [6] as well as Nathalie's thesis, successfully defended on November 12, 2009 in Cachan.

3. Scientific Foundations

3.1. Concurrency

Participants: Benedikt Bollig, Thomas Chatain, Paul Gastin, Stefan Haar, Serge Haddad, Stefan Schwoon, Marc Zeitoun.

Concurrency: Property of systems allowing some interacting processes to be executed in parallel.

Diagnosis: The process of deducing from a partial observation of a system aspects of the internal states or events of that system; in particular, *fault diagnosis* aims at determining whether or not some non-observable fault event has occurred.

Conformance Testing: Feeding dedicated input into an implemented system IS and deducing, from the resulting output of I, whether I respects a formal specification S.

3.1.1. Introduction

It is well known that, whatever the intended form of analysis or control, a *global* view of the system state leads to overwhelming numbers of states and transitions, thus slowing down algorithms that need to explore the state space. Worse yet, it often blurs the mechanics that are at work rather than exhibiting them. Conversely, respecting concurrency relations avoids exhaustive enumeration of interleavings. It allows us to focus on 'essential' properties of non-sequential processes, which are expressible with causal precedence relations. These precedence relations are usually called causal (partial) orders. Concurrency is the explicit absence of such a precedence between actions that do not have to wait for one another. Both causal orders and concurrency are in fact essential elements of a specification. This is especially true when the specification is constructed in a distributed and modular way. Making these ordering relations explicit requires to leave the framework of state/interleaving based semantics. Therefore, we need to develop new dedicated algorithms for tasks such as conformance testing, fault diagnosis, or control for distributed discrete systems. Existing solutions for these problems often rely on centralized sequential models which do not scale up well.

3.1.2. Diagnosis

Participants: Benedikt Bollig, Paul Gastin, Stefan Haar, Stefan Schwoon, Marc Zeitoun.

Fault Diagnosis for discrete event systems is a crucial task in automatic control. Our focus is on *event oriented* (as opposed to *state oriented*) model-based diagnosis, asking e.g. the following questions:
given a - potentially large - *alarm pattern* formed of observations,

- what are the possible *fault scenarios* in the system that *explain* the pattern ?
- Based on the observation, can we deduce whether or not a certain - invisible - fault has actually occurred ?

Model-based diagnosis starts from a discrete event model of the observed system - or rather, its relevant aspects, such as possible fault propagations, abstracting away other dimensions. From this model, an extraction or unfolding process, guided by the observation, produces recursively the explanation candidates.

In asynchronous partial-order based diagnosis with Petri nets [54], [55], [59],[19], one unfolds the *labelled product* of a Petri net model $\mathcal{N}$ and an observed alarm pattern $\mathcal{A}$, also in Petri net form. We obtain an acyclic net giving partial order representation of the behaviors compatible with the alarm pattern. A recursive online procedure filters out those runs *(configurations)* that explain *exactly* $\mathcal{A}$. The Petri-net based approach generalizes to dynamically evolving topologies, in dynamical systems modeled by graph grammars, see [36].

3.1.3. Observability and Diagnosability

Diagnosis algorithms have to operate in contexts with low observability, i.e., in systems where many events are invisible to the supervisor. Checking *observability* and *diagnosability* for the supervised systems is therefore a crucial and non-trivial task in its own right. Analysis of the relational structure of occurrence nets allows us to check whether the system exhibits sufficient visibility to allow diagnosis. Developing efficient methods for both verification of *diagnosability checking* under concurrency, and the *diagnosis* itself for distributed, composite and asynchronous systems, is an important field for *MExICo*.

3.1.4. Distribution

Distributed computation of unfoldings allows one to factor the unfolding of the global system into smaller *local* unfoldings, by local supervisors associated with sub-networks and communicating among each other. In [55], [37], elements of a methodology for distributed computation of unfoldings between several supervisors, underwritten by algebraic properties of the category of Petri nets have been developed. Generalizations, in particular to Graph Grammars, are still do be done.

Computing diagnosis in a distributed way is only one aspect of a much vaster topic, that of *distributed diagnosis* (see [53], [67]). In fact, it involves a more abstract and often indirect reasoning to conclude whether or not some given invisible fault has occurred. Combination of local scenarios is in general not sufficient: the global system may have behaviors that do not reveal themselves as faulty (or, dually, non-faulty) on any local supervisor's domain (compare [35], [39]). Rather, the local diagnosers have to join all *information* that is available to them locally, and then deduce collectively further information from the combination of their views. In particular, even the *absence* of fault evidence on all peers may allow to deduce fault occurrence jointly, see [72], [73]. Automatizing such procedures for the supervision and management of distributed and locally monitored asynchronous systems is a mid-term goal of *MExICo*. We have started a cooperation on these topics with Philippe Dague from University Paris-Sud; the above participants within the team have established a working group on the subject of distributed diagnosability.

3.1.5. Testing

Participants: Benedikt Bollig, Paul Gastin, Stefan Haar.

3.1.5.1. Introduction

The gap between specification and implementation is at the heart of research on formal testing. The general *conformance testing problem* can be defined as follows: Does an implementation $\mathcal{M}'$ conform a given specification $\mathcal{M}$? Here, both $\mathcal{M}$ and $\mathcal{M}'$ are assumed to have input and output channels. The formal model $\mathcal{M}$ of the specification is entirely known and can be used for analysis. On the other hand, the implementation $\mathcal{M}'$ is unknown but interacts with the environment through observable input and output channels. So the behavior of $\mathcal{M}'$ is partially controlled by input streams, and partially observable via output streams. The Testing problem consists in computing, from the knowledge of $\mathcal{M}$, *input streams* for $\mathcal{M}'$ such that observation of the resulting output streams from $\mathcal{M}'$ allows to determine whether $\mathcal{M}'$ conforms to $\mathcal{M}$ as intended.

In this project, we focus on distributed or asynchronous versions of the conformance testing problem. There are two main difficulties. First, due to the distributed nature of the system, it may not be possible to have a unique global observer for the outcome of a test. Hence, we may need to use *local* observers which will record only *partial views* of the execution. Due to this, it is difficult or even impossible to reconstruct a coherent global execution. The second difficulty is the lack of global synchronization in distributed asynchronous systems. Up to now, models were described with I/O automata having a centralized control, hence inducing global synchronizations.

3.1.5.2. Asynchronous Testing

Since 2006 and in particular during his sabbatical stay at the University of Ottawa, Stefan Haar has been working with Guy-Vincent Jourdan and Gregor v. Bochmann of UOttawa and Claude Jard of IRISA on asynchronous testing. In the synchronous (sequential) approach, the model is described by an I/O automaton with a centralized control and transitions labeled with individual input or output actions. This approach has known limitations when inputs and outputs are distributed over remote sites, a feature that is characteristic of , e.g., web computing. To account for concurrency in the system, they have developed in [61], [43] asynchronous conformance testing for automata with transitions labeled with (finite) partial orders of I/O. Intuitively, this is a "big step" semantics where each step allows concurrency but the system is synchronized before the next big step. This is already an important improvement on the synchronous setting. The non-trivial challenge is now to cope with fully asynchronous specifications using models with decentralized control such as Petri nets.

3.1.5.3. Local Testing

Message-Sequence-Charts (MSCs) provide models of behaviors of distributed systems with communicating processes. An important problem is to test whether an implementation conforms to a specification given for instance by an HMSC. In *local testing*, one proceeds by injecting messages to the local processes and observing the responses: for each process p, a local observer records the sequence of events at p. If each local observation is consistent with some MSC defined by the specification, the implementation passes the test. If local testing on individual processes suffices to check conformance, the given specification (an HMSC language) is called locally testable. Local testability turns out to be undecidable even for regular HMSC languages [35]; the main difficulty lies in the existence of implied scenarios, i.e., global behaviors which are locally consistent with different specification scenarios. There are two approaches to attack the problem of local testing in light of this bottleneck. One is to allow joint observations of tuples of processes. This gives rise to the problem of k-testability where one allows joint observations of up to k processes [39]. We will look for structural conditions on the model or the specification ensuring k-testability. Another tactic would be to recognize that practical implementations always work with bounded buffers and impose an upper bound B on the buffer size. The set of B-bounded MSCs in the k-closure of a regular MSC language is again regular, so the B-bounded k-testability problem is decidable for all regular HMSC-definable specifications. The focus could now be on efficiently identifying the smallest k for which an HMSC specification is k-testable. Another interesting problem is to identify a minimal set of tests to validate a k-testable specification.

The first step that should be reached in the near future is the completion of asynchronous testing in the setting without any big-step synchronization. In parallel, work on the problems in local testing should progress sufficiently to allow, in a mid-term perspective, to understand the relations and possible interconnections between local (i.e. distributed) and asynchronous (centralized) testing. The mid-to long term goal (perhaps not yet to achieve in a four-year term) is the comprehensive formalization of testing and testability in asynchronous systems with distributed architecture and test protocols.

3.2. Interaction

Participants: Benedikt Bollig, Thomas Chatain, Paul Gastin, Stefan Haar, Serge Haddad, Marc Zeitoun.

3.2.1. Introduction

Systems and services exhibit non-trivial *interaction* between specialized and heterogeneous components. This interplay is challenging for several reasons. On one hand, a coordinated interplay of several components is required, though each has only a limited, partial view of the system's configuration. We refer to this problem as *distributed synthesis* or *distributed control*. An aggravating factor is that the structure of a component might be semi-transparent, which requires a form of *grey box management*.

Interaction, one of the main characteristics of systems under consideration, often involves an environment that is not under the control of cooperating services. To achieve a common goal, the services need to agree upon a strategy that allows them to react appropriately regardless of the interactions with the environment. Clearly, the notions of opponents and strategies fall within *game theory*, which is naturally one of our main tools in exploring interaction. We will apply to our problems techniques and results developed in the domains of distributed games and of games with partial information. We will consider also new problems on games that arise from our applications.

3.2.2. Distributed Control

Participants: Benedikt Bollig, Thomas Chatain, Paul Gastin, Stefan Haar.

Program synthesis, as introduced by Church [52] aims at deriving directly an implementation from a specification, allowing the implementation to be correct by design. When the implementation is already at hand but choices remain to be resolved at run time then the problem becomes controller synthesis. Both program and controller synthesis have been extensively studied for sequential systems. In a distributed setting, we need to synthesize a distributed program or distributed controllers that interact locally with the system components. The main difficulty comes from the fact that the local controllers/programs have only a partial view of the entire system. This is also an old problem largely considered undecidable in most settings [70], [65], [69], [56], [58]. Actually, the main undecidability sources come from the fact that this problem was addressed in a synchronous setting using global runs viewed as sequences. In a truly distributed system where interactions are asynchronous we have recently obtained encouraging decidability results [57],[17]. This is a clear witness where concurrency may be exploited to obtain positive results. It is essential to specify expected properties directly in terms of causality revealed by partial order models of executions (MSCs or Mazurkiewicz traces). We intend to develop this line of research with the ambitious aim to obtain decidability for all natural systems and specifications. More precisely, we will identify natural hypotheses both on the architecture of our distributed system and on the specifications under which the distributed program/controller synthesis problem is decidable. This should open the way to important applications, e.g., for distributed control of embedded systems.

3.2.3. Adaptation and Grey box management

Participants: Benedikt Bollig, Stefan Haar, Serge Haddad.

Contrary to mainframe systems or monolithic applications of the past, we are experiencing and using an increasing number of services that are performed not by one provider but rather by the interaction and cooperation of many specialized components. As these components come from different providers, one can no longer assume all of their internal technologies to be known (as it is the case with proprietary technology). Thus, in order to compose e.g. orchestrated services over the web, to determine violations of specifications or contracts, to adapt existing services to new situations etc, one needs to analyze the interaction behavior of *boxes* that are known only through their public interfaces. For their semi-transparent-semi-opaque nature, we shall refer to them as **grey boxes**. While the concrete nature of these boxes can range from vehicles in a highway section to hotel reservation systems, the tasks of *grey box management* have universal features allowing for generalized approaches with formal methods. Two central issues emerge:

- Abstraction: From the designer point of view, there is a need for a trade-off between transparency (no abstraction) in order to integrate the box in different contexts and opacity (full abstraction) for security reasons.
- Adaptation: Since a grey box gives a partial view about the behavior of the component, even if it is not immediately useable in some context, the design of an adaptator is possible. Thus the goal is the synthesis of such an adaptator from a formal specification of the component and the environment.

3.3. Management of Quantitative Behavior

Participants: Benedikt Bollig, Thomas Chatain, Paul Gastin, Stefan Haar, Serge Haddad.

3.3.1. Introduction

Besides the logical functionalities of programs, the *quantitative* aspects of component behavior and interaction play an increasingly important role.

- *Real-time* properties cannot be neglected even if time is not an explicit functional issue, since transmission delays, parallelism, etc, can lead to time-outs striking, and thus change even the logical course of processes. Again, this phenomenon arises in telecommunications and web services, but also in transport systems.
- In the same contexts, *probabilities* need to be taken into account, for many diverse reasons such as unpredictable functionalities, or because the outcome of a computation may be governed by race conditions.

- Last but not least, constraints on *cost* cannot be ignored, be it in terms of money or any other limited resource, such as memory space or available CPU time.

Traditional mainframe systems were proprietary and (essentially) localized; therefore, impact of delays, unforeseen failures, etc. could be considered under the control of the system manager. It was therefore natural, in verification and control of systems, to focus on *functional* behavior entirely. With the increase in size of computing system and the growing degree of compositionality and distribution, quantitative factors enter the stage:

- calling remote services and transmitting data over the web creates *delays*;
- remote or non-proprietary components are not "deterministic", in the sense that their behavior is uncertain.

Time and *probability* are thus parameters that management of distributed systems must be able to handle; along with both, the *cost* of operations is often subject to restrictions, or its minimization is at least desired. The mathematical treatment of these features in distributed systems is an important challenge, which *MExICo* is addressing; the following describes our activities concerning probabilistic and timed systems. Note that cost optimization is not a current activity but enters the picture in several intended activities.

3.3.2. Probabilistic distributed Systems

Participants: Stefan Haar, Serge Haddad.

3.3.2.1. Non-sequential probabilistic processes.

Practical fault diagnosis requires to select explanations of *maximal likelihood*; this leads therefore to the question what the probability of a given partially ordered execution is. In Benveniste et al. [38], [33], we presented a model of stochastic processes, whose trajectories are partially ordered, based on local branching in Petri net unfoldings; an alternative and complementary model based on Markov fields is developed in [60], which takes a different view on the semantics and overcomes the first model's restrictions on applicability.

Both approaches abstract away from real time progress and randomize choices in *logical* time. On the other hand, the relative speed - and thus, indirectly, the real-time behavior of the system's local processes - are crucial factors determining the outcome of probabilistic choices, even if non-determinism is absent from the system.

Recently, we started a new line of research with Anne Bouillard, Sidney Rosario, and Albert Benveniste in the DistribCom team at INRIA Rennes, studying the likelihood of occurrence of non-sequential runs under random durations in a stochastic Petri net setting.

Once the properties of the probability measures thus obtained are understood, it will be interesting to relate them with the two above models in logical time, and understand their differences. Another mid-term goal, in parallel, is the transfer to diagnosis with possible cooperation with René Boel's group in Ghent/Belgium.

3.3.2.2. Distributed Markov Decision Processes

Participant:

Distributed systems featuring non-deterministic and probabilistic aspects are usually hard to analyze and, more specifically, to optimize. Furthermore, high complexity theoretical lower bounds have been established for models like partially observed Markovian decision processes and distributed partially observed Markovian decision processes. We believe that these negative results are consequences of the choice of the models rather than the intrinsic complexity of problems to be solved. Thus we plan to introduce new models in which the associated optimization problems can be solved in a more efficient way. More precisely, we start by studying connection protocols weighted by costs and we look for online and offline strategies for optimizing the mean cost to achieve the protocol. We cooperate on this subject with Eric Fabre in the DistribCom team at INRIA Rennes, in the context of the DISC project.

3.3.3. Real time distributed systems

Nowadays, software systems largely depend on complex timing constraints and usually consist of many interacting local components. Among them, railway crossings, traffic control units, mobile phones, computer servers, and many more safety-critical systems are subject to particular quality standards. It is therefore becoming increasingly important to look at networks of timed systems, which allow real-time systems to operate in a distributed manner.

Timed automata are a well-studied formalism to describe reactive systems that come with timing constraints. For modeling distributed real-time systems, networks of timed automata have been considered, where the local clocks of the processes usually evolve at the same rate [68] [46]. It is, however, not always adequate to assume that distributed components of a system obey a global time. Actually, there is generally no reason to assume that different timed systems in the networks refer to the same time or evolve at the same rate. Any component is rather determined by local influences such as temperature and workload.

3.3.3.1. Distributed timed systems with independently evolving clocks

Participants: Benedikt Bollig, Paul Gastin.

A first step towards formal models of distributed timed systems with independently evolving clocks was done in [34]. As the precise evolution of local clock rates is often too complex or even unknown, the authors study different semantics of a given system: The *existential semantics* exhibits all those behaviors that are possible under *some* time evolution. The *universal semantics* captures only those behaviors that are possible under *all* time evolutions. While emptiness and universality of the universal semantics are in general undecidable, the existential semantics is always regular and offers a way to check a given system against safety properties. A decidable under-approximation of the universal semantics, called *reactive semantics*, is introduced to check a system for liveness properties. It assumes the existence of a *global* controller that allows the system to react upon local time evolutions. A short term goal is to investigate a *distributed* reactive semantics where controllers are located at processes and only have local views of the system behaviors.

Several questions, however, have not yet been tackled in this previous work or remain open. In particular, we plan to exploit the power of synchronization via local clocks and to investigate the *synthesis problem*: For which (global) specifications $\mathcal{S}$ can we generate a distributed timed system with independently evolving clocks $\mathcal{A}$ (over some given system architecture) such that both the reactive and the existential semantics of $\mathcal{A}$ are precisely (the semantics of) $\mathcal{S}$? In this context, it will be favorable to have partial-order based specification languages and a partial-order semantics for distributed timed systems. The fact that clocks are not shared may allow us to apply partial-order–reduction techniques.

If, on the other hand, a system is already given and complemented with a specification, then one is usually interested in controlling the system in such a way that it meets its specification. The interaction between the actual *system* and the *environment* (i.e., the local time evolution) can now be understood as a 2-player game: the system's goal is to guarantee a behavior that conforms with the specification, while the environment aims at violating the specification. Thus, building a controller of a system actually amounts to computing winning strategies in imperfect-information games with infinitely many states where the unknown or unpredictable evolution of time reflects an imperfect information of the environment. Only few efforts have been made to tackle those kinds of games. One reason might be that, in the presence of imperfect information and infinitely many states, one is quickly confronted with undecidability of basic decision problems.

3.3.3.2. Equivalences between Models with Time and Concurrency (EMoTiCon)

Participants: Thomas Chatain, Stefan Haar, Serge Haddad.

This is the subject of a project of the Farman institute of ENS Cachan in collaboration with the LURPA (laboratory for automated production at ENS Cachan).

Due to the dramatic development of the techniques that aim at improving the security of automated systems (synthesis, verification, test...), several classes of models are often needed to study a complex system, either to give several views of the system or to study the same aspect of the system using several techniques. Thus one often needs to transform a model from one formalism to another or to compare models written in different formalisms (time Petri nets, networks of timed automata...), that have common features: they allow one to model both (dense) time and concurrency. These transformations are usually done by hand and rely on natural equivalences between the basic components of the models. For instance, a state of an automaton corresponds intuitively to a place of a Petri net; a transition of a Petri net corresponds to a tuple of synchronized transitions in a network of automata; the interval of possible delays associated with a transition of a time Petri net corresponds to a pair invariant/guard in a timed automaton. But these natural equivalences do not apply easily to general models. And since the transformations are usually built on case studies and for ad-hoc reasons, no effort is made to generalize them and most often the relations between the initial model and the transformed one are not formalized.

Nevertheless we see clearly that the transformations on case studies tend naturally to preserve concurrency. Moreover this property is appreciated because it improves the readability of the transformation and makes the transformed model faithful to the initial one and to the modeled system. But these ad hoc transformations are difficult generalize. Thus, not surprisingly, the first works about formal comparison of the expressiveness of different models [48], [47], [75] [49], [50], [51], [44], [45] did not take preservation of concurrency into account. These works make extensive use of tricks that destroy concurrency and focus only on the preservation of sequential (interleaving) timed semantics.

In contrast, we aim at formalizing and automating translations that preserve both the timed semantics and the concurrent semantics. This effort is crucial for extending concurrency-oriented methods for logical time. In fact, validation and management - in a broad sense - of distributed systems is not realistic in general without understanding and control of their real-time dependent features; the link between real-time and logical-time behaviors is thus crucial for many aspects of *MExICo*'s work.

3.3.4. Weighted Automata and Weighted Logics

Participants: Benedikt Bollig, Paul Gastin.

Time and probability are only two facets of quantitative phenomena. A generic concept of adding weights to qualitative systems is provided by the theory of weighted automata [32]. They allow one to treat probabilistic or also reward models in a unified framework. Unlike finite automata, which are based on the Boolean semiring, weighted automata build on more general structures such as the natural or real numbers (equipped with the usual addition and multiplication) or the probabilistic semiring. Hence, a weighted automaton associates with any possible behavior a weight beyond the usual Boolean classification of “acceptance” or “non-acceptance”. Automata with weights have produced a well-established theory and come, e.g., with a characterization in terms of rational expressions, which generalizes the famous theorem of Kleene in the unweighted setting. Equipped with a solid theoretical basis, weighted automata finally found their way into numerous application areas such as natural language processing and speech recognition, or digital image compression.

What is still missing in the theory of weighted automata are satisfactory connections with verification-related issues such as (temporal) logic and bisimulation that could lead to a general approach to corresponding satisfiability and model-checking problems. A first step towards a more satisfactory theory of weighted systems was done in [7]. That paper, however does not give final solutions to all the aforementioned problems. It identifies directions for future research that we will be tackling.

4. Application Domains

4.1. Panorama

MExICo's research is motivated by problems on system management in several domains:

- In the domain of service oriented computing, it is often necessary to insertion some Web service into an existing orchestrated business process, e.g. to replace another component after failures. This requires to ensure, often actively, conformance to the interaction protocol. One therefore needs to synthesize *adaptators* for every component in order to steer its interaction with the surrounding processes.
- Still in the domain of telecommunications, the supervision of a network tends to move from out-of-band technology, with a fixed dedicated supervision infrastructure, to in-band supervision where the supervision process uses the supervised network itself. This new setting requires to revisit the existing supervision techniques using control and diagnosis tools.
- Several recent Intelligent Transport Systems projects aim at providing assistance to drivers, in the way of (partially) automated motorways. We will focus on modeling and analysis of the collision avoidance problems in critical short sections of motorways.

This list is likely to grow over the next years as we continue our research.

4.2. Autonomous Telecommunications Systems: In-Band Supervision

Participants: Stefan Haar, Serge Haddad.

In the context of traditional hard-wired communication networks, supervision structures for managing faults, configuration, provisioning etc could be developed with a fixed infrastructure, and perform the communication between sensors, supervisors, policy enforcement points etc over a separate network using separate hardware. This rigid, **out-of-band** technology does not survive passing to today's and tomorrow's services and networks. In fact, the dynamic mobility of services combined across sites and domains cannot be captured unless the network used for supervision evolves in the same way and simultaneously, which rules out static solutions; but providing out-of-band infrastructure that grows with the networks to be supervised would be prohibitively expensive, if at all technically feasible. *Heterogeneity* is the other feature of modern networks that forces a change, since different domains are not likely to agree on a pervasive third-party supervision. Rather, the providers will keep control over the internal state and evolution of their domain, and accept only exchange through standardized outward interfaces.

Supervision has thus to be re-invented on an *in-band*, *autonomous* base: monitoring probes deployed on the web, dysfunctions on one peer node diagnosed by another peer in a network with changing configuration, enhanced supervisor and actor capacities of services, etc. *MExICo* will work on improving the interoperability of service components through continued application of e.g. distributed techniques for control and diagnosis.

4.3. Traffic Safety Control

Participant: Serge Haddad.

The *Intelligent Transport Systems* **(ITS)** community tries to deal with the numerous challenges that arise when designing secure and reliable software dedicated to automatic transport systems.

Several recent ITS projects aim at providing assistance to drivers and deal with partially automated motorways. The community investigated first a fully automated infrastructure and vehicles approach (as in the PATH project [63]) in the 1990's. That approach was then abandoned in favor of a new line of research and development activities, more centered on safety strategies to ensure properties such as Collision Avoidance or Safety Margin for Assistance Vehicles [64].

This vision relies on cooperative systems where *"road operators, infrastructure, vehicles, their drivers and other road users will cooperate to deliver the most efficient, safe, secure and comfortable journeys"* [42]. Implementing such a system then follows a peer-to-peer organization where each vehicle must fully cooperate in a time-constrained and safety-critical environment.

In that context, many projects are dealing with safety-oriented applications based on sensors, communication devices and protocols as well as distributed traffic management systems involving cooperation between the infrastructure and vehicles [41], [40], [74]. Thus, reliability, flexibility in the design as well as safety are primary issues. Such systems are even more complex to analyze than previous distributed systems. Consequently, there is a need for a specific methodology and tools to design and analyze them.

We will focus on an approach for the modeling and analysis of the collision avoidance problems in critical short sections of motorways with the aim to check whether a control strategy exists depending on the parameters (speed, safety distances, etc.). We intend to cope with the undecidability of such problems by appropriate discretizations and with high complexity of the obtained systems by using elaborated data structures based on decision diagrams.

4.4. Web Services

Participants: Stefan Haar, Serge Haddad.

Specific applications targeted by *MExICo* include the problem of adaptation in Service-Oriented Computing (SOC). The challenge is here twofold, stemming both from the distributed nature of services (scattered over the entire web) and their heterogeneous origins.

4.4.1. Context

Web services have become the most frequently used model of design and programming based on components for business applications. Web service languages like BPEL have useful constructors that manage for instance exceptions, (timed guarded) waiting of messages, parallel execution of processes, distant service invocations, etc. Interoperability of components is based on interaction protocols associated with them and often published on public or private registers. In the framework of Web services, these protocols are called abstract processes by contrast with business processes (i.e. services). Composition of components must be analyzed for several reasons and at least to avoid deadlocks during execution. This has led to numerous works that focus on compositional verification, substitution of a component by another one, synthesis of adaptators, etc., and triggered a push towards a unifying theoretical framework (see e.g. [71], [76])

4.4.2. Problems

Interoperability requires that when a user or a program wants to interact with the component, the knowledge of the interaction protocol is enough. Our previous works have shown that the interaction protocols can be inherently ambiguous: no client can conduct a correct interaction with the component in every scenario. This problem is even more complex when the protocol can evolve during execution due to adaptation requirements. The composition of components also raises interesting problems. When composing optimal components (w.r.t. the number of states for instance) the global component can be non optimal. So one aims at reducing a posteriori or better on the fly the global component. At last, the dynamical insertion of a component in a business process requires to check whether this insertion is behaviorally consistent [77], [66]

We do not intend to check global properties based on a modular verification technique. Rather, given an interaction protocol per component and a global property to ensure, we want to synthesize an adaptator per component such that this property is fulfilled or to detect that there cannot exist such adaptators [62]. In another research direction, one can introduce the concept of utility of a service and then optimize a system i.e. keeping the same utility value while reducing the resources (states, transitions, clocks, etc.).

5. Software

5.1. Software

5.1.1. libalf: the Automata Learning Framework

Participant: Benedikt Bollig [correspondant].

libalf is a comprehensive, open-source library for learning finite-state automata covering various well-known learning techniques (such as, Angluin s L^*, Biermann, and RPNI, as well as a novel learning algorithm for NFA. libalf is highly flexible and allows for facilely interchanging learning algorithms and combining domain-specific features in a plug-and-play fashion. Its modular design and its implementation in C++ make it a flexible platform for adding and engineering further, efficient learning algorithms for new target models (e.g., Büchi automata).

Details on libalf can be found at http://libalf.informatik.rwth-aachen.de/

5.1.2. Mole: an unfolder for Petri Nets

Participant: Stefan Schwoon [correspondant].

Mole computes, given a safe Petri net, a finite prefix of its unfolding. It is designed to be compatible with other tools, such as PEP and the Model-Checking Kit, which are using the resulting unfolding for reachability checking and other analyses. The tool Mole arose out of earlier work on Petri nets. In the context of MExiCo, we are extending it to handle contextual Petri nets. A preliminary (but inefficient) implementation has been achieved, which we intend to improve to obtain a viable, efficient tool.

Details on Mole can be found at http://www.fmi.uni-stuttgart.de/szs/tools/mole/

6. New Results

6.1. Quantitative systems

6.1.1. Real-time probabilistic systems

This year, new results were obtained in the cooperation of Stefan Haar with the DistribCom team in Rennes:

- *Monotonicity*: We established criteria to check whether and when global latencies depend in a monotonic way on local latencies, see [13].
- *Criticality*: parallelism, branching and synchronization in asynchronous compositions may either hide or accentuate individual delays. Some components' behavior is critical for global performance while the variation in performance of other components is without effect on the composite performance. Combining the analysis of partially ordered structure and of probabilistic delays we show in [12] how to identify critical components, and thus to allocate resources efficiently in view of improving performance.

Moreover this year, Serge Haddad developped jointly an efficient (approximate) method for the quantitative model checking of PCTL formula for interval-valued discrete time Markov chains (IMC). Whilst the exact model checking has been shown to be NP and co-NP hard, our method is polynomial-time. Furthermore, this procedure returns more refined answers than traditional ones: YES, NO, DON'T KNOW. Thus we may provide useful partial information for modelers in the "DON'T KNOW" case.

6.1.2. Weighted Automata

New results were obtained on weighted pushdown automata, which form the basis for interprocedural dataflow analysis. The problem is related to studying fixed-point equations on idempotent semirings. The efficiency with which these problems can be solved depends on certain algebraic properties of the weights involved. In comparison with previous work, we managed to establish more general properties that still enable an efficient solution [22].

6.1.3. Weighted versus Probabilistic Logics

Benedikt Bollig and Paul Gastin have introduced for arbitrary semirings weighted versions of MSO and CTL that generalize both the boolean logics and their probabilistic versions. Being able to use arbitrary semirings to specify quantitative properties of systems allows to deal with new quantities such as energy, rewards, expected rewards, etc. They established expressiveness results giving translations from weighted and probabilistic CTL into weighted MSO. This should open the way to more general quantitative verification of systems. This work was presented in the invited lecture [7].

6.2. Concurrency

6.2.1. Concurrent Recursive Programs

We introduced a new automata class, called *concurrent visibly pushdown automata*, to model recursive programs that communicate via shared memory. While most previous approaches to those kind of systems restrict to the analysis of an already existing program, our new model allows one to tackle the *synthesis problem*: the automatic generation of programs from global specifications. Our results extend classical theorems from the theory of Mazurkiewciz traces to the recursive setting [9].

6.2.2. Local Safety and Local Liveness for Distributed Systems

Volker Diekert and Paul Gastin have defined local versions of the classical *safety* and *liveness* properties. By *local* we mean that we are interested in the local views of processes and not in *global* configurations or snapshots. The advantage is to obtain reasonable complexity, i.e., PSPACE as for sequential systems. They charachterized local safety by local closure and local liveness by local density. Restricting to first-order definable properties, they proved a decomposition theorem in the spirit of the separation theorem for linear temporal logic. They also characterized local safety and local liveness by means of canonical local temporal logic formulae.

6.2.3. Diagnosis

Refinements were obtained in the analysis of unfoldings for the purposes of diagnosis. Stefan Haar's study of the *reveals* relation in occurrence nets has led to novel criteria on observability in labeled Petri nets, presented in [19].

6.2.4. Automata theory

We made a detailed re-investigation of the emptiness problem for Büchi automata, which is at the heart of LTL model checking [18]. More precisely, we studied on-the-fly algorithms of the type that are used in the acclaimed Spin tool. We compared existing and new algorithms experimentally suite and proposed improvements. Compared with the algorithm implemented in Spin, our best algorithm is faster by about 33 per cent on average. The paper was named among the best six at *MEMICS* and invited for a journal publication.

6.3. Interaction

6.3.1. Distributed Control

Thomas Chatain, Paul Gastin, Nathalie Sznajder have defined a new and promising framework to study distributed synthesis. The system is asynchronous and communication is with signals. They proved the decidability of distributed synthesis for strongly connected architectures and external specifications with natural closure properties. In a fascinating domain where most decidability results are negative, this is an interesting step forwards which may open the way to many more positive results for distributed synthesis.

6.3.2. Web Services

Serge Haddad with several other researchers has proposed an extension of the BPEL language in order to mix BPEL specifications with JAVA code in a natural way [3].

7. Contracts and Grants with Industry

7.1. Industry

So far, several contacts with industry have been established, but contracts have not yet materialized. Cooperations may result from the EU project proposals that have been submitted (see below) which involve different industrial partners; this will be clearer in 2010.

8. Other Grants and Activities

8.1. Regional Actions

8.1.1. Farman Project EMoTiCon

Participants: Thomas Chatain, Stefan Haar, Serge Haddad.

The EMoTiCon project is a collaboration between researchers from *MExICo* and from the laboratory of automated production at ENS Cachan. As such, this collaboration is funded by the Farman institute of ENS Cachan.

The scentific context of the EMoTiCon project is the variety of formalisms used to model distributed real-time systems. We aim at formalizing and automating translations between these models, that preserve both the timed semantics and the concurrent semantics. This problem is described in Section 3.3.3.2.

8.2. National Actions

8.2.1. ANR DOTS

Participants: Benedikt Bollig, Thomas Chatain, Paul Gastin, Serge Haddad, Marc Zeitoun.

The DOTS project is a collaboration with researchers from IRCCyN, IRISA, LAMSADE, LaBRI and LSV.

The scientific context of the DOTS project is specification, verification and design of information systems. Complex systems, such as embedded systems that are widely used nowadays (telecommunication, transport, automation), are often distributed –composed of several components that communicate together–, timed –contain timing constraints–, and open –interact with their environment. Each of these aspects considered separately is now relatively well understood and corresponds to an active research area. The big challenge is to deal with systems which present several of these features.

The aim of the DOTS project is to associate researchers specialized in verification of different aspects mentioned above in order to tackle problems that emerge when considering several features simultaneously. In this way we plan to significantly advance both theory as well as algorithmics of design and verification of distributed, open and timed systems.

The research of *MExICo* about distributed control (Section 3.2.2) and real time distributed systems (Section 3.3.3) take place in the DOTS project.

8.2.2. ANR CHECKBOUND ANR-06-SETI-002

Participants: Hilal Djafri, Serge Haddad.

The increasing use of computerised systems in all aspects of our lives gives an increasing importance on the need for them to function correctly. The presence of such systems in safety-critical applications, coupled with their increasing complexity, makes indispensable their verification to see if they behaves as required . Thus the model checking which is the automated manner of formal verification techniques is of particular interest. Since verification techniques have becomemore efficient and more prevalent, the natural extension is to extend the range of models and specification formalisms to which model checking can be applied. Indeed the behaviour of many real-life processes is inherently stochastic, thus the formalism has been extended to probabilistic model checking. Therefore, different formalisms in which the underlying system has been modelled by Markovian models have been proposed.

Stochastic model checking can be performed by numerical or statistical methods. In model checking formalism, models are checked to see if the considered measures are guaranteed or not, bounding techniques become useful.We propose to apply Stochastic Comparison technique for numerical stochastic model checking. The main advantage of this approach is the possibility to derive transient and steady-state bounding distributions as well as the possibility to avoid the state space explosion problem. For the statistical model checking we propose to study the application of perfect simulation by coupling in the past. This method has been shown that to be efficient when the underlying system is monotonous for the exact steady-state distribution sampling. We consider to extend this approach for transient analysis and to model checking by means of bounding models and the stochastic monotonicity. One of difficult problems for model checking formalism, we envisage to study is when the state space is infinite. In some cases, it would be possible to consider bounding models defined in finite state space.

Indeed, formal verification using model checking and performance and dependability evaluation have a lot of things in common. We think that it would be interesting to apply the methods that we have a large experience in quantitative evaluation in the context of stochastic model checking.

8.2.3. DIGITEO 2009-27HD CoChaT: Covert Channels in Timed Systems

Participant: Serge Haddad.

Attacks with timing channels have been described and simulated for instance on TCP/IP protocols, Web communications or cryptographic operations. The scientific objective of the CoChaT project is to study the conditions underwhich such attacks can occur in timed systems, with two main directions. a. The first step consists in defining a theoretical framework, in which timing channels can be formally described. b. A second part of thework concerns the design of detection and verification algorithms, for which decidability issues are involved. Progress in both steps will have to take into account practical examples like the case studies mentionned above, in order to validate the formal approach.

8.2.4. Submissions: INRIA Associated teams

Participants: Serge Haddad, Stefan Haar.

Serge Haddad and Stefan Haar participate in the associated team FOSSA led by the DistribCom team at INRIA Rennes, with the University of Texas, Austin. The objective is *Formalizing Orchestration and Secure Services Analysis*; see http://www.irisa.fr/distribcom/FOSSA2010/Fossa10.html.

8.3. Actions Funded by the EC

8.3.1. Current Projects

8.3.1.1. DISC Grant Agreement 224498

Project Distributed Supervisory Control of Large Plants - DISC (acronym) The European Commission supports the project financially by the EU.ICT program, Challenge ICT-2007.3.3 (Information and Communication Technologies (ICT)). 1 September 2008 - 1 Septemer 2011

Within the DISC project's WP 2 on diagnosis, Stefan Haar started a cooperation with Rene Boel, Universtiy of Ghent, Belgium.

8.3.2. Submissions

Serge Haddad and Stefan Haar participate in the Network of Excellence proposal *Highly-complex and networked control systems (HYCON2)*, FP7-ICT-2009-5, as well as the large-scale integrating project (IP) proposal *Univerself*, FP7-ICT-2009-05.

8.4. International Actions

8.4.1. Current Projects

8.4.1.1. EGIDE/Procope Project Smyle

Participant: Benedikt Bollig.

Smyle (**S**ynthesizing **M**odels b**y** **L**earning from **E**xamples) is a framework for synthesizing design models. The novel aspect of our approach is to exploit *learning* algorithms. The technical heart of *Smyle* is a procedure that interactively infers a design model from a given set of positive and negative scenarios provided as message sequence charts (MSCs). Furthermore, on establishing the inconsistency of a set of MSCs, our approach mechanically provides *diagnostic feedback* (in the form of a counterexample) that can guide the engineer to evolve his requirements. (Project with RWTH Aachen and TU Munich, Germany, funded in 2008 and 2009.)

Details on *Smyle* can be found at http://www.smyle-tool.org/

8.4.1.2. ARCUS Inde

Most participants of the team participate in the sub-project 4, Formal approaches for computer systems, of the Ile-de-France/Inde project of the ARCUS program (Region Ile-de-France and Foreign Affairs Ministry, France), funded for 3 years (2008 – 2010).

9. Dissemination

9.1. Scientific animation

9.1.1. Benedikt Bollig

was a program committee member of the Young Researchers Workshop on Concurrency Theory (*YR-CONCUR 2009*), co-located with *CONCUR 2009*. He will be involved in the organization of the conference *CONCUR 2010*. He is/was a reviewer for the international journals *Logical Methods in Computer Science*, *TOSEM*, *Theory of Computing Systems*, *Foundations of Computer Science*, *Fundamenta Informaticae*, and the international conferences FSTTCS'09, CONCUR'09, ATVA'09, ICTAC'09, FOSSACS'10.

9.1.2. Thomas Chatain

was a member of the jury of the PhD of Medesu Sogbohossou, under supervision of Olivier H. Roux at IRCCyN. He is a member of the program committee of the *International Conference on Application of Concurrency to System Design (ACSD) 2010*. He will participate in the organization of the conference *CONCUR 2010* in Paris.

9.1.3. Paul Gastin

is an associate editor of the *Journal of Automata, Languages and Combinatorics*. He was on the scientific committee of the international workshop on *Automata, Concurrency and Timed Systems (ACTS)* in Chennai, January 2009. He was an invited speaker at the *International Conference on Developments in Language Theory (DLT'09)* in Stuttgart, July 2009. He was/is a member of the program committees of ICTAC'09 (*6th International Colloquium on Theoretical Aspects of Computing*), LATA'10 (*4th International Conference on Language and Automata Theory and Applications*), DLT'10 (*14th International Conference on Development in Language Theory*). He co-organizes the international workshops on *Quantitative Models: Ex- pressiveness and Analysis* in Dagstuhl, January 2010; on *Automata, Concurrency and Timed Systems (ACTS)* in Chennai, February 2010; on *Weighted Automata: Theory and Applications (WATA'10)* in Leipzig, May 2010. He will organize (co-chair with François Laroussinie) the *International Conference on Concurrency Theory (CONCUR'10)* in Paris, September 2010. He was on the defense committee for the PhD thesis of Nathalie Sznajder (as supervisor) and Aldric Degorre (as examinator). He is the head of the computer science department of ENS Cachan. He was the head of the Parisian Master of Research in Computer Science (until August 2009).

9.1.4. Stefan Haar

is an associate editor of *IEEE Transactions on Automatic Control* until the end of 2009, and has accepted to become an associate editor for *Discrete event dynamic systems: theory and application* in 2010. He will participate in the organization (publicity) of the conference *CONCUR 2010* in Paris, and is a member of the program committee of *Workshop On Discrete Event Systems (WODES) 2010* in Berlin. He is also the correspondent of the *DRI* (international relations service) of INRIA for the Saclay center.

9.1.5. Serge Haddad

Serge Haddad has been a member of the editorial board of the journal *Technique et Science Informatiques* since 2007, and a member of the steering committee of the *International Conference on Applications and Theory of Petri Nets (ICATPN)* since 2001.

In 2009, he was the workshops and tutorials chair of the organizing committee of *ICATPN 2009* held in Paris, and was also vice-president of the organizing committee of the french speaking school for young researchers *"Ecole d'été Temps-Réel (ETR'2009)"* in Paris.

In 2009, he has also been member of the program committees of the following conferences and events:

- 30th International Conference on Application and Theory of Petri nets (ICATPN 2009), Paris, France.
- Workshops PNSE et TISTO associated with ICATPN 2009.
- Seventh "Colloque Francophone sur la Modélisation des Systèmes Réactifs" MSR 2009, Nantes, France.
- Ecole d'été Temps Réel ETR'09, Paris, France.

He was on the defense committee for the HdR of E. Grolleau (Poitiers), F. Pommereau (Créteil) as reviewer, and N. Sabouret (Paris 6) as president. He was on the defense committee for PhD of of F. Bouchy (Cachan) as prsident, S. Youcef (Dauphine), D. Coulibaly (Dauphine) as supervisor and T.Q. Tran (Bordeaux) as reviewer.

He is responsible of years L3 and M1 of the computer science department of ENS Cachan.

9.1.6. Stefan Schwoon

was a member of the program committee of CIAA 2009 (International Conference on Implementation and Application of Automata) and reviewer of a dozen other articles for international journals and conferences. He will participate in the organisation of CONCUR 2010, and has participated in the Dagstuhl Seminar on Graph Search Engineering (Nov/Dec 2009) with a talk on *Comparison of algorithms for checking emptiness on Büchi automata.*

9.2. Visits and Visitors

9.2.1. Visits received

9.2.1.1. Rolf HENNECKER

Rolf Hennecker, professor at Ludwig-Maximilians-Universität München, Germany, visited MExICo from March 30 to April 10th, 2009. He worked with Dorsaf El Hog and Serge Haddad on modal specifications for asynchronous composition of systems. In return, Dorsaf El Hog has visited R. Hennicker during two weeks in July 2009.

9.2.1.2. Marc ZEITOUN

Marc ZEITOUN, PR University Bordeaux 1, joined the LSV and MExIcO on a 1 year INRIA position (*délégation INRIA*, from Sept 1st 2009 to August 31, 2010).

His main research interests concern the verification of distributed systems, with a focus on the control and synthesis problems. He will also be involved in the diagnosis for concurrent systems. Finally, he is interested in developing tools and specification languages adapted to checking quantitative aspects of systems.

9.2.1.3. Volker DIEKERT

Volker DIEKERT, professor at the Univesity of Stuttgart. He worked with Paul Gastin on control and synthesis for distributed systems, 2 weeks in October 2009.

9.2.1.4. Madhavan MUKUND and K. Narayan KUMAR

Madhavan MUKUND and K. Narayan KUMAR, professors at the Chennai Mathematical Institute, worked with S. Akshay, Benedikt Bollig and Paul Gastin on distributed and timed systems, 2 weeks in June 2009.

9.2.1.5. Aiswarya CYRIAC

Aiswarya CYRIAC, student from IMSc in Chennai. Research internship with Stéphane Demri and Paul Gastin on focus games for LTL specifications, 7 weeks, June-July 2009.

9.2.1.6. Michael WALDVOGEL

Michael WALDVOGEL, student from Germany. Research internship with Benedikt Bollig and Paul Gastin on concurrent recursive programs, 2 months, April-May 2009.

9.2.1.7. Martin LEUCKER

Martin LEUCKER, professor at the Technische Universität München, worked with Benedikt Bollig on learning of alternating automata, 1 week in July 2009.

9.2.1.8. Paolo BALDAN

Paolo BALDAN, associate professor at the University of Padova, worked with Stefan Schwoon and Stefan Haar on contextual unfoldings of Petri nets, December 8 to 11, 2009.

9.2.2. Visits to other laboratories

Stefan Haar visited the group of Prof. *Rene BOEL* at Ghent University, Belgium, from June 2 to 4, 2009, and the group of Prof. *Luca Bernardinello*, Università di Milano, Italy, November 4 to 6, 2009.

9.3. Teaching

Benedikt Bollig is giving a lecture on weighted automata in the *Parisian Master of Research in Computer Science* (M2).

Paul Gastin is a professor and head of the computer science department at ENS Cachan. He was also head of the Parisian Master of Research in Computer Science until August 2009. He teaches courses on algorithms and on formal languages in the *Licence d'informatique* (L3), on basics of verification in the *Parisian Master of Research in Computer Science* (M1), and several topics in the *agrégation* program at ENS Cachan.

Serge Haddad is a professor at ENS Cachan. He currently teaches courses on Complexity and Advanced Algorithmicin the *Licence d'informatique* (L3), a course in the program on preparation for *agrégation* (with S. Haar) a course on Algorithms, and, in MPRI-M2 a course on probabilistic timed systems.

Thomas Chatain is maître de conférences at ENS Cachan. He teaches logic, graph theory, network programming, algorithms and C++ programming. He also prepares for the modélisation option in the *agrégation* program at ENS Cachan.

Stefan Schwoon is maître de conférences at ENS Cachan. He currently teaches a course on operating systems, a tutorial on logic (both L3) and a course on verification at the M1 level of the MPRI program.

10. Bibliography

Year Publications

Articles in International Peer-Reviewed Journal

[1] É. ANDRÉ, TH. CHATAIN, E. ENCRENAZ, L. FRIBOURG. *An Inverse Method for Parametric Timed Automata*, in "International Journal of Foundations of Computer Science", vol. 20, n^{o} 5, October 2009, p. 819-836, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/ACEF-ijfcs09.pdf.

[2] B. BOLLIG, J.-P. KATOEN, C. KERN, M. LEUCKER. *Learning Communicating Automata from MSCs*, in "IEEE Transactions on Software Engineering", 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BKKL-tse09.pdf, To appear DE .

[3] C. BOUTROUS-SAAB, D. COULIBALY, S. HADDAD, T. MELLITI, P. MOREAUX, S. RAMPACEK. *An Integrated Framework for Web Services Orchestration*, in "International Journal of Web Services Research", vol. 6, n^o 4, September 2009, p. 1-29.

[4] P. BOUYER, S. HADDAD, P.-A. REYNIER. *Undecidability Results for Timed Automata with Silent Transitions*, in "Fundamenta Informaticae", vol. 92, n^o 1-2, 2009, p. 1-25, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2007-12.pdf.

[5] S. DONATELLI, S. HADDAD, J. SPROSTON. *Model Checking Timed and Stochastic Properties with CSL^{TA}*, in "IEEE Transactions on Software Engineering", vol. 35, n^o 2, March-April 2009, p. 224-240, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DHS-tose09.pdf GB IT .

[6] P. GASTIN, N. SZNAJDER, M. ZEITOUN. *Distributed synthesis for well-connected architectures*, in "Formal Methods in System Design", vol. 34, n^o 3, June 2009, p. 215-237, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/GSZ-fmsd09.pdf.

Invited Conferences

[7] B. BOLLIG, P. GASTIN. *Weighted versus Probabilistic Logics*, in "Proceedings of the 13th International Conference on Developments in Language Theory (DLT'09), Stuttgart, Germany", V. DIEKERT, D. NOWOTKA (editors), Lecture Notes in Computer Science, vol. 5583, Springer, June-July 2009, p. 18-38, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BG-dlt09.pdf.

International Peer-Reviewed Conference/Proceedings

[8] É. ANDRÉ, TH. CHATAIN, O. DE SMET, L. FRIBOURG, S. RUEL. *Synthèse de contraintes temporisées pour une architecture d'automatisation en réseau*, in "Actes du 7ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'09), Nantes, France", O. H. ROUX (editor), Hermès, November 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/ACDFR-msr09.pdf, To appear.

[9] B. BOLLIG, M.-L. GRINDEI, P. HABERMEHL. *Realizability of Concurrent Recursive Programs*, in "Proceedings of the 12th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'09), York, UK", L. DE ALFARO (editor), Lecture Notes in Computer Science, vol. 5504, Springer, March 2009, p. 410-424, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BGH-fossacs09.pdf.

[10] B. BOLLIG, P. HABERMEHL, C. KERN, M. LEUCKER. *Angluin-Style Learning of NFA*, in "Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09), Pasadena, CA, USA", C. BOUTILIER (editor), AAAI Press, July 2009, p. 1004-1009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BHKL-ijcai09.pdf DE .

[11] B. BOLLIG, J.-P. KATOEN, C. KERN, M. LEUCKER. *SMA—The Smyle Modeling Approach*, in "Proceedings of the 3rd IFIP TC2 Central and East European Conference on Software Engineering Techniques (CEE-SET'08), Brno, Czech Republic", Z. HUZAR, B. MEYER (editors), Lecture Notes in Computer Science, Springer, 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BKKL-ceeset2008.pdf, To appear.

[12] A. BOUILLARD, S. HAAR, S. ROSARIO. *Critical paths in the Partial Order Unfolding of a Stochastic Petri Net*, in "Proceedings of the 7th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'09), Budapest, Hungary", J. OUAKNINE, F. VAANDRAGER (editors), Lecture Notes in Computer Science, vol. 5813, Springer, September 2009, p. 43-57, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BHR-formats09.pdf.

[13] A. BOUILLARD, S. ROSARIO, A. BENVENISTE, S. HAAR. *Monotonicity in Service Orchestrations*, in "Proceedings of the 30th International Conference on Applications and Theory of Petri Nets (ICATPN'09), Paris, France", G. FRANCESCHINIS, K. WOLF (editors), Lecture Notes in Computer Science, vol. 5606, Springer, June 2009, p. 263-282, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BRBH-atpn09.pdf.

[14] P. BULYCHEV, TH. CHATAIN, A. DAVID, K. G. LARSEN. *Checking simulation relation between timed game automata*, in "Proceedings of the 7th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'09), Budapest, Hungary", J. OUAKNINE, F. VAANDRAGER (editors), Lecture Notes in Computer Science, vol. 5813, Springer, September 2009, p. 73-87, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BCDL-formats09.pdfDKRU.

[15] B. BÉRARD, S. HADDAD. *Interrupt Timed Automata*, in "Proceedings of the 12th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'09), York, UK", L. DE ALFARO (editor), Lecture Notes in Computer Science, vol. 5504, Springer, March 2009, p. 197-211, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BH-fossacs09.pdf.

[16] TH. CHATAIN, A. DAVID, K. G. LARSEN. *Playing Games with Timed Games*, in "Proceedings of the 3rd IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'09), Zaragoza, Spain", A. GIUA, M. SILVA, J. ZAYTOON (editors), September 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/CDL-adhs09.pdf DK .

[17] TH. CHATAIN, P. GASTIN, N. SZNAJDER. *Natural Specifications Yield Decidability for Distributed Synthesis of Asynchronous Systems*, in "Proceedings of the 35th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'09), Špindlerův Mlýn, Czech Republic", M. NIELSEN, A. KUČERA, P. BRO MILTERSEN, C. PALAMIDESSI, P. TŮMA, F. VALENCIA (editors), Lecture Notes in Computer Science, vol. 5404, Springer, January 2009, p. 141-152, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/CGS-sofsem09.pdf.

[18] A. GAISER, S. SCHWOON. *Comparison of Algorithms for Checking Emptiness on Büchi Automata*, in "Proceedings of the 5th Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS), Znojmo, Czechia", November 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/GS-memics09.pdf DE .

[19] S. HAAR. *Qualitative Diagnosability of Labeled Petri Nets Revisited*, in "48th IEEE Conference on Decision and Control", 2009.

[20] S. HADDAD, F. KORDON, L. PETRUCCI, J.-F. PRADAT-PEYRE, N. TRÈVES. *Efficient State-Based Analysis by Introducing Bags in Petri Nets Color Domains*, in "Proceedings of the 28th American Control Conference (ACC'09), Saint Louis, MO, USA", June 2009, p. 5018-5025, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/HKPPT-acc09.pdf.

[21] S. HADDAD, N. PEKERGIN. *Using Stochastic Comparison for Efficient Model Checking of Uncertain Markov Chains*, in "Proceedings of the 6th International Conference on Quantitative Evaluation of Systems (QEST'09), Budapest, Hungary", IEEE Computer Society Press, September 2009, p. 177-186, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/HP-qest09.pdf.

[22] M. KÜHNRICH, S. SCHWOON, J. SRBA, S. KIEFER. *Interprocedural Dataflow Analysis over Weight Domains with Infinite Descending Chains*, in "Proceedings of the 12th International Conference on Foundations of Software Science and Computational Structures (FOSSACS), York, UK", L. DE ALFARO (editor), Lecture

Notes in Computer Science, vol. 5504, Springer, March 2009, p. 440–455, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/KSSK-fossacs09.pdf DE DK .

[23] H. M. ZARGAYOUNA, F. BALBO, S. HADDAD. *Agents Secure Interaction in Data Driven Languages*, in "Preliminary Proceedings of the 2nd Workshop on LAnguages, methodologies and Development tools for multi-agent systemS (LADS'09), Torino, Italy", M. DASTANI, A. EL FALLAH SEGHROUCHNI, J. LEITE, P. TORRONI (editors), September 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/ZBH-lads09.pdf.

National Peer-Reviewed Conference/Proceedings

[24] S. HADDAD, L. MOKDAD, S. YOUCEF. *Bornes du temps de réponse des services Web composites*, in "Actes du 7ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'09), Nantes, France", O. H. ROUX (editor), Hermès, November 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/HMY-msr09.pdf, To appear.

Scientific Books (or Scientific Book chapters)

[25] S. DEMRI, P. GASTIN. *Specification and Verification using Temporal Logics*, in "Modern applications of automata theory", D. D'SOUZA, P. SHANKAR (editors), IISc Research Monographs, vol. 2, World Scientific, 2009, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DG-iis09.pdf, To appear.

[26] V. DIEKERT, P. GASTIN. *Local safety and local liveness for distributed systems*, in "Perspectives in Concurrency Theory", K. LODAYA, M. MUKUND, R. RAMANUJAM (editors), IARCS-Universities, Universities Press, January 2009, p. 86-106, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DG-pct08.pdf DE .

[27] M. DROSTE, P. GASTIN. *Weighted automata and weighted logics*, in "Handbook of Weighted Automata", W. KUICH, H. VOGLER, M. DROSTE (editors), EATCS Monographs in Theoretical Computer Science, chap. 5, Springer, 2009, p. 175-211, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DG-hwa08.pdf DE .

[28] P. GASTIN, M. MUKUND, K. NARAYAN KUMAR. *Reachability and boundedness in time-constrained MSC graphs*, in "Perspectives in Concurrency Theory", K. LODAYA, M. MUKUND, R. RAMANUJAM (editors), IARCS-Universities, Universities Press, January 2009, p. 157-183, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/GMN-pct08.pdf IN .

Research Reports

[29] B. BÉRARD, S. HADDAD, M. SASSOLAS. *Verification on Interrupt Timed Automata*, n° LSV-09-16, Laboratoire Spécification et Vérification, ENS Cachan, France, July 2009, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2009-16.pdf, 16 pages, Research Report.

[30] A. GAISER, S. SCHWOON. *Comparison of Algorithms for Checking Emptiness on Büchi Automata*, n° 0910.3766, Computing Research Repository, 2009, Technical report.

[31] M. KÜHNRICH, S. SCHWOON, J. SRBA, S. KIEFER. *Interprocedural Dataflow Analysis over Weight Domains with Infinite Descending Chains*, n° 0901.0501, Computing Research Repository, 2009, Technical report.

References in notes

[32] W. KUICH, H. VOGLER, M. DROSTE (editors). *Handbook of Weighted Automata*, EATCS Monographs in Theoretical Computer Science, Springer, 2009 DE .

[33] S. ABBES, A. BENVENISTE, S. HAAR. *A Petri net model for distributed estimation*, in "Proc. MTNS 2004, Sixteenth International Symposium on Mathematical Theory of Networks and Systems, Louvain (Belgium), ISBN 90-5682-517-8", 2004.

[34] S. AKSHAY, B. BOLLIG, P. GASTIN, M. MUKUND, K. NARAYAN KUMAR. *Distributed Timed Automata with Independently Evolving Clocks*, in "Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada", F. VAN BREUGEL, M. CHECHIK (editors), Lecture Notes in Computer Science, vol. 5201, Springer, August 2008, p. 82-97, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/ABGMN-concur08.pdf.

[35] R. ALUR, K. ETESSAMI, M. YANNAKAKIS. *Realizability and Verification of MSC Graphs*, in "Theor. Comput. Sci.", vol. 331, n^{o} 1, 2005, p. 97–114.

[36] P. BALDAN, TH. CHATAIN, S. HAAR, B. KOENIG. *Unfolding-based Diagnosis of Systems with an Evolving Topology*, in "Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08), Toronto, Canada", F. VAN BREUGEL, M. CHECHIK (editors), Lecture Notes in Computer Science, vol. 5201, Springer, August 2008, p. 203-217, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BCHK-concur08.pdf, Extended version accepted for publication in: Information and Computation DE IT .

[37] P. BALDAN, S. HAAR, B. KOENIG.. *Distributed Unfolding of Petri Nets*, in "Proc.FOSSACS 2006", LNCS, vol. 3921, Springer, 2006, p. 126-141, Extended version: Technical Report CS-2006-1. Department of Computer Science, University Ca' Foscari of Venice. DE IT .

[38] A. BENVENISTE, E. FABRE, S. HAAR.. *Markov Nets: Probabilistic Models for distributed and concurrent Systems*, in "IEEE Transactions on Automatic Control", vol. 48 (11), 2003, p. 1936-1950, Extended version: IRISA Research Report 1538..

[39] P. BHATEJA, P. GASTIN, M. MUKUND, K. NARAYAN KUMAR. *Local testing of message sequence charts is difficult*, in "Proceedings of the 16th International Symposium on Fundamentals of Computation Theory (FCT'07), Budapest, Hungary", E. CSUHAJ-VARJÚ, Z. ÉSIK (editors), Lecture Notes in Computer Science, vol. 4639, Springer, August 2007, p. 76-87, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BGMN-fct07.pdf IN .

[40] R. BISHOP. *Intelligent Vehicle R&D: a review and contrast of programs worldwide and emerging trends.*, in "Annals of Telecommunications - Intelligent Transportation Systems", J. EHRLICH (editor), vol. 60, n^{o} 3-4, GET-Lavoisier, March-April 2005, p. 228–263.

[41] J.-M. BLOSSEVILLE. *Driving assistance systems and road safety: State-of-the-art and outlook*, in "Annals of Telecommunications - Intelligent Transportation Systems", J. EHRLICH (editor), vol. 60, n^{o} 3-4, GET-Lavoisier, March-April 2005, p. 281–298.

[42] P. BLY. *e-Safety - Co-operative Systems for Road Transport (IST Work Progamme 2005-2006)*, European Commission, 2004, Technical report.

[43] G. V. BOCHMANN, S. HAAR, C. JARD, G.-V. JOURDAN. *Testing Systems Specified as Partial Order Input/Output Automata.*, in "Proc. TESTCOM/Fates 08, 20th IFIP International Conference on Testing of Communicating Systems and 8th International Workshop on Formal Approaches to Testing of Software", LNCS, vol. 5047, Springer, 2008, p. 169-183 CA .

[44] P. BOUYER, S. HADDAD, P.-A. REYNIER. *Extended Timed Automata and Time Petri Nets*, in "Proceedings of the 6th International Conference on Application of Concurrency to System Design (ACSD'06), Turku, Finland", K. GOOSSENS, L. PETRUCCI (editors), IEEE Computer Society Press, June 2006, p. 91-100, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2006-01.pdf.

[45] P. BOUYER, S. HADDAD, P.-A. REYNIER. *Timed Petri Nets and Timed Automata: On the Discriminating Power of Zeno Sequences*, in "Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) — Part II, Venice, Italy", M. BUGLESI, B. PRENEEL, V. SASSONE, I. WEGENER (editors), Lecture Notes in Computer Science, vol. 4052, Springer, July 2006, p. 420-431, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BHR-icalp06.pdf.

[46] P. BOUYER, S. HADDAD, P.-A. REYNIER. *Timed Unfoldings for Networks of Timed Automata*, in "Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), Beijing, ROC", S. GRAF, W. ZHANG (editors), Lecture Notes in Computer Science, vol. 4218, Springer, October 2006, p. 292-306, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BHR-atva06.pdf.

[47] M. BOYER, M. DIAZ. *Non equivalence between time Petri nets and time stream Petri nets*, in "PNPM", IEEE Computer Society, 1999, p. 198-207.

[48] M. BOYER, O. H. ROUX. *Comparison of the Expressiveness of Arc, Place and Transition Time Petri Nets*, in "ICATPN", J. KLEIJN, A. YAKOVLEV (editors), Lecture Notes in Computer Science, vol. 4546, Springer, 2007, p. 63-82.

[49] B. BÉRARD, F. CASSEZ, S. HADDAD, D. LIME, O. H. ROUX. *Comparison of Different Semantics for Time Petri Nets*, in "Proceedings of the 3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'05), Taipei, Taiwan, ROC", D. A. PELED, Y.-K. TSAY (editors), Lecture Notes in Computer Science, vol. 3707, Springer, October 2005, p. 293-307, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BCHLR05-atva.pdf.

[50] B. BÉRARD, F. CASSEZ, S. HADDAD, D. LIME, O. H. ROUX. *Comparison of the Expressiveness of Timed Automata and Time Petri Nets*, in "Proceedings of the 3rd International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'05), Uppsala, Sweden", P. PETTERSSON, W. YI (editors), Lecture Notes in Computer Science, vol. 3829, Springer, November 2005, p. 211-225, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BCHLR-formats2005.pdf.

[51] B. BÉRARD, F. CASSEZ, S. HADDAD, D. LIME, O. H. ROUX. *When are Timed Automata Weakly Timed Bisimilar to Time Petri Nets?*, in "Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05), Hyderabad, India", R. RAMANUJAM, S. SEN (editors), Lecture Notes in Computer Science, vol. 3821, Springer, December 2005, p. 273-284, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BCHLR-fsttcs05.pdf.

[52] A. CHURCH. *Logic, arithmetics, and automata*, in "Proc. of Int. Congr. of Mathematicians", 1962, p. 23–35.

[53] R. DEBOUK, D. TENEKETZIS. *Coordinated decentralized protocols for failure diagnosis of discrete-event systems*, in "Journal of Discrete Event Dynamical Systems: Theory and Application", vol. 10, 2000, p. 33–86.

[54] E. FABRE, A. BENVENISTE, C. JARD, S. HAAR. *Diagnosis of Asynchronous Discrete Event Systems, a Net Unfolding Approach.*, in "IEEE Trans. Aut. Control", vol. 48 (5), 2003, p. 714-727.

[55] E. FABRE, A. BENVENISTE, C. JARD, S. HAAR. *Distributed monitoring of concurrent and asynchronous systems.*, in "Discrete Event Dynamic Systems: theory and application", vol. 15 (1), 2005, p. 33-84, Preliminary version: Proc. CONCUR 2003, LNCS 2761, pp.1–28, Springer..

[56] B. FINKBEINER, S. SCHEWE. *Uniform distributed synthesis*, in "Proc. of the 20th IEEE Annual Symposium on Logic in Computer Science (LICS'05)", IEEE Computer Society Press, 2005, p. 321–330.

[57] P. GASTIN, B. LERMAN, M. ZEITOUN. *Distributed games with causal memory are decidable for series-parallel systems*, in "Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'04), Chennai, India", K. LODAYA, M. MAHAJAN (editors), Lecture Notes in Computer Science, vol. 3328, Springer, December 2004, p. 275-286, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/GLZ-fsttcs04.pdf.

[58] P. GASTIN, N. SZNAJDER, M. ZEITOUN. *Distributed synthesis for well-connected architectures*, in "Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06), Kolkata, India", N. GARG, S. ARUN-KUMAR (editors), Lecture Notes in Computer Science, vol. 4337, Springer, December 2006, p. 321-332, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/GSZ-fsttcs2006.pdf.

[59] S. HAAR, A. BENVENISTE, E. FABRE, C. JARD. *Partial Order Diagnosability Of Discrete Event Systems Using Petri Net Unfoldings.*, in "42nd IEEE Conference on Decision and Control (CDC)", 2003.

[60] S. HAAR. *Probabilistic Cluster Unfoldings*, in "Fundamenta Informaticae", vol. 53 (3-4), 2003, p. 281-314.

[61] S. HAAR, C. JARD, G.-V. JOURDAN. *Testing Input/Output Partial Order Automata.*, in "Proc. TESTCOM/FATES", LNCS, vol. 4581, SpringerLNCS 4581, pp. 171–185., 2007, p. 171-185 CA .

[62] S. HADDAD, P. POIZAT. *Transactional Reduction of Component Compositions*, in "Proceedings of 27th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'07), Tallinn, Estonia", J. DERRICK, J. VAIN (editors), Lecture Notes in Computer Science, vol. 4574, Springer, June 2007, p. 341-357, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/HP-forte07.pdf.

[63] R. HOROWITZ, P. VARAIYA. *Control Design of an Automated Highway System*, in "Proc. of the IEEE 88(7)", 2000.

[64] INTELLIGENT VEHICLE INITIATIVE. *Saving Lives through advanced vehicle safety technology*, September 2005.

[65] O. KUPFERMAN, M. Y. VARDI. *Synthesizing Distributed Systems*, in "Proc. of the 16th IEEE Annual Symposium on Logic in Computer Science (LICS'01)", IEEE Computer Society Press, 2001.

[66] D. KÖNIG, N. LOHMANN, S. MOSER, C. STAHL, K. WOLF. *Extending the compatibility notion for abstract WS-BPEL processes*, in "WWW", ACM, 2008, p. 785-794.

[67] S. LAFORTUNE, Y. WANG, T.-S. YOO. *Diagnostic Decentralisé Des Systèmes A Evénements Discrets*, in "Journal Europeen des Systèmes Automatisés (RS-JESA)", vol. 99, n° 99, August 2005, p. 95–110.

[68] K. G. LARSEN, P. PETTERSSON, W. YI. *Compositional and symbolic model-checking of real-time systems*, in "Proc. of RTSS 1995", IEEE Computer Society, 1995, p. 76-89.

[69] S. MOHALIK, I. WALUKIEWICZ. *Distributed Games*, in "Proc. of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'03)", LNCS, vol. 2914, Springer, 2003, p. 338–351.

[70] A. PNUELI, R. ROSNER. *Distributed reactive systems are hard to synthesize*, in "Proc. of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS'90)", vol. II, IEEE Computer Society Press, 1990, p. 746–757.

[71] W. REISIG. *Towards a Theory of Services*, in "UNISCON", Lecture Notes in Business Information Processing, vol. 5, Springer, 2008, p. 271-281.

[72] L. RICKER, K. RUDIE. *Know Means No: Incorporating Knowledge into Discrete-Event Control Systems*, in "IEEE Transactions on Automatic Control", vol. 45, n^{o} 9, September 2000, p. 1656–1668.

[73] L. RICKER, K. RUDIE. *Knowledge Is a Terrible Thing to Waste: Using Inference in Discrete-Event Control Problems*, in "IEEE Transactions on Automatic Control", vol. 52, n^{o} 3, MarchSeptember 2007, p. 428–441.

[74] Y. ROBIN-JOUAN, J. EHRLICH, B. GUILLAUMIN, M. DELARCHE, M. DUTECH. *Transport-specific communication services: Safety-based or critical applications for mobiles and cooperation with infrastructure networks*, in "Annals of Telecommunications - Intelligent Transportation Systems", J. EHRLICH (editor), vol. 60, n^{o} 3–4, GET-Lavoisier, March-April 2005, p. 405–440 DE .

[75] J. SRBA. *Timed-Arc Petri Nets vs. Networks of Timed Automata*, in "ICATPN", Lecture Notes in Computer Science, vol. 3536, Springer, 2005, p. 385-402.

[76] S. STAAB, W. VAN DER AALST, V. BENJAMINS, A. SHETH, J. MILLER, C. BUSSLER, A. MAEDCHE, D. FENSEL, D. GANNON. *Web Services: Been There, Done That?*, in "IEEE Intelligent Systems", vol. 18, 2003, p. 72-85.

[77] M. B. VAN RIEMSDIJK, R. HENNICKER, M. WIRSING, A. SCHROEDER. *Service Specification and Matchmaking Using Description Logic*, in "AMAST", Lecture Notes in Computer Science, vol. 5140, Springer, 2008, p. 392-406.