INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team S4

System Synthesis and Supervision, Scenarios

Rennes - Bretagne-Atlantique

Theme : Embedded and Real Time Systems

Table of contents

S4 *is a joint project of* INRIA, CNRS *and the University of Rennes 1, within* IRISA *(UMR 6074).*

1. Team

Research Scientists

Benoît Caillaud [Team Leader, CR]
Eric Badouel [CR, HdR]
Albert Benveniste [Research Director (DR), part-time in S4, HdR]
Philippe Darondeau [Research Director (DR), HdR]
Axel Legay [CR]

Faculty Members

Guillaume Aucher [INRIA Chair Associate Professor, since October 2010]
Sophie Pinchinat [Professor, HdR]

Technical Staff

Cyrille Jégourel [Software Engineer, started October 2010, funded by the COMBEST european project]

PhD Students

Bastien Maubert [From October 2010]
Benoît Delahaye [Teaching Assistant, University of Rennes 1; Then Research Technologist, funded by the CESAR European project]
Bernard Fotsing [Funded by AUF (Agence universitaire de la Francophonie), part time in France]
Rodrigue Tchougong Ngongang [Funded by SARIMA, part time in France]
Valérie Murat [Started October 2010]

Post-Doctoral Fellows

Timothy Bourke [Post-Doctoral Fellow, started 16 November 2009, funded by the SYNCHRONICS large-scale initiative action of INRIA]
Laura Bozzelli [Post-Doctoral Fellow, until May 2010, funded by the COMBEST european project]
Uli Farhenberg [Post-Doctoral Fellow, started December 2010]

Visiting Scientists

Célestin Nkuimi-Jugnia [University of Yaoundé, visiting researcher from 20 December 2009 to 20 February 2010]
Shin-ya Katsumata [Research Institute for Mathematical Sciences, Kyoto university, visiting researcher in August 2010]
Andrzej Wasowski [IT University of Copenhagen, visiting researcher in July 2010]

Administrative Assistant

Laurence Dinh [TR, part-time in S4]

Others

Guillaume Madelaine [Research training from 1 June 2010 until 15 July 2010]
Lois Vanhée [ENS de Cachan, Antenne de Bretagne, Master Internship from 1 February 2010 to 30 July 2010]
Antoine Amarilli [ENS Researh training from 7 June 2010 until 14 August 2010]

2. Overall Objectives

2.1. Introduction

The objective of the project is the realization by algorithmic methods of reactive and distributed systems from partial and heterogeneous specifications. Methods, algorithms and tools are developed to synthesize reactive software from one or several incomplete descriptions of the system's expected behavior, regarding functionality (synchronization, conflicts, communication), control (safety, reachability, liveness), deployment architecture (mapping, partitioning, segregation), or even quantitative performances (response time, communication cost, throughput).

These techniques are better understood on fundamental models, such as automata, Petri nets, event structures and their timed extensions. The results obtained on these basic models are then adapted to those realistic but complex models commonly used to design embedded and telecommunication systems.

The behavioral views of the *Unified Modeling Language* (UML) (sequence diagrams and statecharts), the *High-Level Message Sequence Charts* (HMSC) and the synchronous reactive language Signal are the heart of the software prototypes being developed and the core of the technology transfer strategy of the project.

The scientific objectives of the project can be characterized by the following elements:

A focus on a precise type of applications: The design of real-time embedded software to be deployed over dedicated distributed architectures. Engineers in this field face two important challenges. The first one is related to system specification. Behavioral descriptions should be adaptable and composable. Specifications are expressed as requirements on the system to be designed. These requirements fall into four categories: (i) functional (synchronization, conflict, communication), (ii) control (safety, reachability, liveness), (iii) architectural (mapping, segregation) and (iv) quantitative (response time, communication cost, throughput, etc). The second challenge is the deployment of the design on a distributed architecture. Domain-specific software environments, known as *middleware* or *real-time operating systems* or *communication layers*, are now part of the usual software design process in industry. They provide a specialized and platform-independent distributed environment to higher-level software components. Deployment of software components and services should be done in a safe and efficient manner.

A specific methodology: The development of methods and tools which assist engineers since the very first design steps of reactive distributive software. The main difficulty is the adequacy of the proposed methods with standard design methods based on components and model engineering, which most often rely on heterogeneous formalisms and require correct-by-construction component assembly.

A set of scientific and technological foundations: Those models and methods which encompass (i) the distributed nature of the systems being considered, (ii) true concurrency, and (iii) real-time.

The contribution of the S4 Project-Team consists of algorithms and tools producing distributed reactive software from partial heterogeneous specifications of the system to be synthesized (functionality, control, architecture, quantitative performances). This means that several heterogeneous specifications (for instance, sequence diagrams and state machines) can be combined, analyzed (are the specifications consistent?) and mapped to lower-level specifications (for instance, communicating automata, or Petri nets).

The scientific approach of Team S4 begins with a rigorous modeling of problems and the development of sound theoretical foundations. This not only allows to prove the correctness (functionality and control) of the proposed transformations or analysis; but this can also guarantee the optimality of the quantitative performances of the systems produced with our methods (communication cost, response time).

Synthesis and verification methods are best studied within fundamental models, such as automata, Petri nets, event structures, synchronous transition systems. Then, results can be adapted to more realistic but complex formalisms, such as the UML. The research work of Team S4 is divided in four main tracks:

Petri net synthesis: This track follows up the main research theme of the former Team PARAGRAPHE at INRIA Rennes on the synthesis of Petri net models using the theory of regions.

Heterogeneous systems: This track contributes to the extension of the well-established synchronous paradigm to distributed systems. The aim is to provide a unified framework in which both synchronous systems, and particular asynchronous systems (so-called weakly-synchronous systems) can be expressed, combined, analyzed and transformed.

Reactive components: The design of reusable components calls for rich specification formalisms, with which the interactions of a component with its environment combines expectations with guarantees on its environment.We are investigating questions related to reactive component refinement and composition. We are also investigating the issues of coherence of views and modularity in complex specifications.

Discrete event system synthesis and supervisory control: Many synthesis and supervisory control problems can be expressed with full generality in the *quantified mu-calculus*, including the existence of optimal solutions to such problems. Algorithms computing winning strategies in parity games (associated with formulas in this logic) provide effective methods for solving such control problems. This framework offers means of classifying control problems, according to their decidability or undecidability, but also according to their algorithmic complexity.

2.2. Highlights of the year

Several achievements of the S4 team are worth being highlighted:

- The completion of the COMBEST European project on component based design of complex embedded systems (Section 7.1). Our overall contribution to the project has consisted in theories, algorithms and tools to support 1/ modular design methods for embedded systems and 2/ the statistical analysis methods for complex embedded systems.
- The extension of interface theories to timed and stochastic properties (Section 6.3).
- A new research direction on hybrid modeling has been opened in team-project S4 (Section 6.2). Our first contribution on this topic is a hybrid system semantics based on non-standard analysis.
- New contributions to the *quasi-static scheduling problem* and opacity control on systems under partial observation (Section 6.4).

3. Scientific Foundations

3.1. Scientific Foundations

The research work of the team is built on top of solid foundations, mainly, algebraic, combinatorial or logical theories of transition systems. These theories cover several sorts of systems which have been studied during the last thirty years: sequential, concurrent, synchronous or asynchronous. They aim at modeling the behavior of finite or infinite systems (usually by abstracting computations on data), with a particular focus on the control flow which rules state changes in these systems. Systems can be autonomous or reactive, that is, embedded in an environment with which the system interacts, both receiving an input flow, and emitting an output flow of events and data. System specifications can be explicit (for instance, when the system is specified by an automaton, extensively defined by a set of states and a set of transitions), or implicit (symbolic transition rules, usually parameterized by state or control variables; partially-synchronized products of finite transition systems; Petri nets; systems of equations constraining the transitions of synchronous reactive systems, according to their input flows; etc.). Specifications can be non-ambiguous, meaning that they fully define at most one system (this holds in the previous cases), or they can be ambiguous, in which case more than one system is conforming to the specification (for instance, when the system is described by logical formulas in the modal mu-calculus, or when the system is described by a set of scenario diagrams, such as *Sequence Diagrams* or *Message Sequence Charts*).

Systems can be described in two ways: either the state structure is described, or only the behavior is described. Both descriptions are often possible (this is the case for formal languages, automata, products of automata, or Petri nets), and moving from one representation to the other is achieved by folding/unfolding operations.

Another taxonomy criteria is the concurrency these models can encompass. Automata usually describe sequential systems. Concurrency in synchronous systems is usually not considered. In contrast, Petri nets or partially-synchronized products of automata are concurrent. When these models are transformed, concurrency can be either preserved, reflected or even, infused. An interesting case is whenever the target architecture requires distributing events among several processes. There, communication-efficient implementations require that concurrency is preserved as far as possible and that, at the same time, causality relations are also preserved. These notions of causality and independence are best studied in models such as concurrent automata, Petri nets or Mazurkiewicz trace languages.

Here are our sources of inspiration regarding formal mathematical tools:

1. Jan van Leeuwen (ed.), *Handbook of Theoretical Computer Science - Volume B: Formal Models and Semantics*, Elsevier, 1990.
2. Jörg Desel, Wolfgang Reisig and Grzegorz Rozenberg (eds.), *Lectures on Concurrency and Petri nets*, Lecture Notes in Computer Science, Vol. 3098, Springer, 2004.
3. Volker Diekert and Grzegorz Rozenberg (eds.), *The Book of Traces*, World Scientific, 1995.
4. André Arnold and Damian Niwinski, *Rudiments of Mu-Calculus*, North-Holland, 2001.
5. Gérard Berry, *Synchronous languages for hardware and software reactive systems - Hardware Description Languages and their Applications*, Chapman and Hall, 1997.

Our research exploits decidability or undecidability results on these models (for instance, inclusion of regular languages, bisimilarity on automata, reachability on Petri nets, validity of a formula in the mu-calculus, etc.) and also, representation theorems which provide effective translations from one model to another. For instance, Zielonka's theorem yields an algorithm which maps regular trace languages to partially-synchronized products of finite automata. Another example is the theory of regions, which provides methods for mapping finite or infinite automata, languages, or even *High-Level Message Sequence Charts* to Petri nets. A further example concerns the mu-calculus, in which algorithms computing winning strategies for parity games can be used to synthesize supervisory control of discrete event systems.

Our research aims at providing effective representation theorems, with a particular emphasis on algorithms and tools which, given an instance of one model, synthesize an instance of another model. In particular we have contributed a theory, several algorithms and a tool for synthesizing Petri nets from finite or infinite automata, regular languages, or languages of *High-Level Message Sequence Charts*. This also applies to our work on supervisory control of discrete event systems. In this framework, the problem is to compute a system (the controller) such that its partially-synchronized product with a given system (the plant) satisfies a given behavioral property (control objective, such as a regular language or satisfaction of a mu-calculus formula).

Software engineers often face problems similar to *service adaptation* or *component interfacing*, which in turn, often reduce to particular instances of system synthesis or supervisory control problems.

4. Application Domains

4.1. Application Domains

Results obtained in Team S4 apply to the design of real-time systems consisting of a distributed hardware architecture, and software to be deployed over that architecture. A particular emphasis is put on *embedded* systems (automotive, avionics, production systems, *etc.*), and also, to a lesser extent, *telecommunication* and *production* systems.

Our work on contract-based modular reasoning has found applications in embedded system design, by supporting and controlling concurrent design activities in aeronautics (see the SPEEDS or COMBEST European projects, Section 7.1).

Our work on heterogeneous reactive systems facilitates the mapping of pure synchronous designs onto a distributed architecture where communication is done by non-instantaneous message passing. These architectures can be usual *asynchronous* distributed systems or, more interestingly, *loosely time-triggered architectures* (LTTA), such as those found on board of recent Airbus aircrafts. In the latter, communication is done by periodically reading or writing (according to local inaccurate real-time clocks) distributed shared variables, without any means of synchronizing these operations. The consequence is that values may be lost or duplicated, and software designed for such specific architectures must resist losses or duplications of messages. In the context of the IST European network of excellence ARTIST we have developed a theoretical and methodological framework in which the correct mapping of synchronous designs to such particular distributed architectures can be best understood, at a high level of abstraction.

5. Software

5.1. InterSMV: A Symbolic Modal Interface Verification Tool

Participants: Benoît Caillaud, Bastien Maubert.

Interfaces have emerged as an essential concept for component-based system engineering. Industrial needs, and requirements applicable to interface theories in the context of embedded system design have been analyzed in [15]. Thanks to its rich composition algebra, the modal interface theory, developed by INRIA in the realm of the Combest project, matches these requirements, while retaining a low computational complexity. It enables both modularity in design, an assume-guarantee style of reasoning and a better separation of concerns and responsibilities between design teams. It is actually a unification of the Assume/Guarantee Contract theory [29], developed in the SPEEDS project and of the Interface Automata theory [31], therefore allowing to express requirements using any of these two styles.

However, Modal Modal lack expressivity: Deadlock freedom or termination can not be expressed and it can not support a relational (aka synchronous) semantics, as seen in synchronous programming languages and the SMV/NuSMV model-checkers. This is not the case of acceptance specifications [30]. However, acceptance specifications are computationally untractable, as sets of sets of labels have to be handled. This becomes even worse with a relational semantics, where labels are mappings and no efficient symbolic representation is known.

Adding marked states (or language) to modal specifications provides the required expressivity to capture both deadlock freedom, some liveness properties, and most interestingly, a relational semantics. The good news about marked modal specifications is that computational complexity is as good as for modal specifications. Hence, we gain expressivity without paying a price.

Thanks to the introduction of marked states and to an ad-hoc encoding, interfaces have been extended to finite data types and the full power of a synchronous interaction semantics. This is implemented, in part, in the InterSMV modal interface verifier. This tool is capable of computing compositions of interfaces (conjunction, product and quotient) and can decide satisfaction and refinement relations. The key algorithm is a symbolic least fix-point based reduction of marked modal specification that has to be applied whenever a composition operator is applied to several terms. This is followed by a satisfaction or refinement checking, using also a symbolic least fix-point computation algorithm.

6. New Results

6.1. Petri Nets and their Synthesis

Participant: Philippe Darondeau.

6.1.1. Separability in Petri Nets

We have proved in [19] that plain, bounded, persistent and reversible Petri nets are strongly separable, which means that any such net with an initial marking $M = k \times (M/k)$ may be simulated by the parallel composition of k independent copies of the net with initial marking M/k, and conversely. We have also proved that given such a net with initial marking $M = k \times (M/k)$, any copy of the net with a smaller initial marking $k' \times (M/k)$ is again bounded, persistent and reversible. These results may be of some help for simulating massively parallel systems, and in particular workflow systems.

6.1.2. Non-Interference in Petri Nets

We have studied in [20] non-interference over unbounded (Place/Transition) Petri nets. The definitions of transitive or intransitive non-interference have been adapted from similar definitions given earlier for labelled transition systems. The interpretation of intransitive non-interference which we propose for Petri nets is as follows. A Petri net represents the composition of a controlled and a controller systems, possibly sharing places and transitions. Low transitions represent local actions of the controlled system, high transitions represent local decisions of the controller, and downgrading transitions represent synchronized actions of both components. Intransitive non-interference means the impossibility for the controlled system to follow any local strategy that would force or dodge synchronized actions depending upon the decisions taken by the controller after the last synchronized action. The fact that both language equivalence and bisimulation equivalence are undecidable for unbounded labelled Petri nets might be seen as an indication that non-interference properties based on these equivalences cannot be decided. We prove the opposite, providing results of decidability of non-interference over a representative class of infinite state systems.

Steps, where the reward for firing a single transition is either fixed or it depends on the current net marking. The simplicity of the algorithms supports our claim that the proposed approach is practical.

6.2. Heterogeneous Systems

Participants: Eric Badouel, Albert Benveniste, Timothy Bourke, Benoît Caillaud, Bernard Fotsing, Rodrigue Tchougong Ngongang.

6.2.1. Hybrid Modeling

In the realm of Synchronics large scale initiative (http://synchronics.inria.fr/), we have proposed to use *non-standard analysis* as a semantics domain for hybrid systems. Non standard analysis is an extension of classical analysis in which infinitesimal variables (the ε and η in the celebrated generic sentence $\forall\varepsilon\exists\eta\cdots$ in college mathematics) can be manipulated as first class citizens. This allows us to provide a denotational semantics and a constructive semantics for hybrid systems, thus establishing simulation engines on a sound but flexible mathematical foundation. These semantics offer a clean separation of concerns between the numerical analyst (solving differential equations) and the computer scientist (generating execution schemes). The semantics is presented in [18].

Also, we have studied a number of practical and fundamental issues in hybrid system modelers that give raise to non reproducibility of results, non determinism, and undesirable side effects arising from numerical inaccuracy. To address part of these problems we have proposed a simple single assignment language in which *continuous time* and *discrete time* are handled as types. The proposed type system takes the form of a Hindley-Milner type systems with a simple form of effects. Functions are tagged to be either discrete (tag D), continuous (tag C) or polymorphic (tag A). A function with tag D must be activated at discrete instants defined as zero-crossing events while a function with tag C must be exercised by the solver (because it defines a signal with a continuous time domain). We have defined a compilation technique to separate the code given to the numerical solver from the discrete part evolving at discrete instants [28].

6.2.2. Complex Workflow Systems

Complex system design includes various aspects involving different teams with different skills using heterogeneous techniques and tools. This process can be handled in the context of a distributed workflow system where structured documents, associated with the several aspects or viewpoints for the same system, are used as interface between the various teams. Information between a component and its context can flow through inherited and synthesized attributes respectively. We thus focus on the classical model of attribute grammars which we use for the design and modular construction of domain specific languages [12]. We also established that the descriptional composition of attribute grammars can be implemented as an ordinary tree transducer composition acting on cyclic data structures.

6.3. Component-Based Design

Participants: Eric Badouel, Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Sophie Pinchinat, Axel Legay.

6.3.1. Modal Interface Theory

In [15], we present the *modal interface* theory, a unification of *interface automata* and *modal specifications*, two radically dissimilar models for interface theories. Interface automata is a game-based model, which allows the designer to express assumptions on the environment and which uses an optimistic view of composition: *two components can be composed if there is an environment where they can work together*. Modal specifications are a language theoretic account of a fragment of the modal mu-calculus logic with a rich composition algebra which meets certain methodological requirements but which does not allow the environment and the component to be distinguished. The present paper contributes a more thorough unification of the two theories by correcting a first attempt in this direction by Larsen et al., drawing a complete picture of the modal interface algebra, and pushing the comparison between interface automata, modal automata and modal interfaces even further.

6.3.2. Timed Interface Theories

Time is a crucial aspect of systems and more typically for embedded systems. We have proposed two timed extensions of modal specifications: modal event-clock automata and timed I/O interfaces. The latter are timed automata whose transitions are equipped with Input (environment) and Output (system) modalities. We defined satisfaction, refinement, composition, and conjunction. We also proposed an optimistic game-based approach to decide whether a specification admits at least one implementation. The theory comes together with an algorithm to synthesize an interface automaton from two specifications. This year, our approach has been implemented as an extension of the well-known UPPAAL toolset. The resulting tool, which we call ECDAR, has been applied on several case studies.

We have also addressed the problem of alternating simulation refinement for concurrent timed games (TG). We show that checking timed alternating simulation between TG is EXPTIME-complete, and provide a logical characterization of this preorder in terms of a meaningful fragment of a new logic, TAMTL$_*$. TAMTL$_*$ is an action-based timed extension of standard alternating-time temporal logic ATL$_*$, which allows to quantify over strategies where the designated coalition of players is not responsible for blocking time. While for full TAMTL$_*$, model-checking TG is undecidable, we show that for its fragment TAMTL, corresponding to the timed version of ATL, the problem is instead decidable and in EXPTIME.

6.3.3. Stochastic Interface Theories

Many of the results given in this section are synthesized in the Ph.D. of Benoît Delahaye [11]. A *contract* allows to distinguish hypotheses made on a system (the guarantees) from those made on its environment (the assumptions). In [24], we focus on models of Assume/Guarantee contracts for (stochastic) systems. We consider contracts capable of capturing reliability and availability properties of such systems. We also show that classical notions of Satisfaction and Refinement can be checked by effective methods thanks to a reduction to classical verification problems. Finally, theorems supporting compositional reasoning and enabling the scalable analysis of complex systems are also studied.

Notions of specification, implementation, satisfaction, and refinement, together with operators supporting stepwise design, constitute a specification theory. In [22], we construct such a theory for Markov Chains (MCs) employing a new abstraction of a Constraint MC. Constraint MCs permit rich constraints on probability distributions and thus generalize prior abstractions such as Interval MCs. Linear (polynomial) constraints suffice for closure under conjunction (respectively parallel composition). This is the first specification theory for MCs with such closure properties. We discuss its relation to simpler operators for known languages such as probabilistic process algebra. Despite the generality, all operators and relations are computable.

6.3.4. Stochastic Model Checking

We are primarily interested in the Statistical Model Checking approach (SMC) SMC have recently been proposed as an alternative to avoid an exhaustive exploration of the state-space of a system under verification. The core idea of the approach is to conduct some simulations of the system and then use results from the statistic area in order to decide whether the system satisfies the property with respect to a given probability. The answer is correct up to some confidence. SMC is generally much faster (but less precise) than formal verification techniques.

We considered to used SMC to verifying applications working within a huge (more than 2^{3000} states) heterogeneous system, that is the cabin communication system of an airplane (HCS). Specifications of this system were provided by EADS – our industrial partner in COMBEST. The difficulty in this verification process comes from network communication which makes all applications interfering and therefore forces to explore the full state-space of the system. Unfortunately, SMC was not capable to compete with the size of the case study. This motivated the development of a new simulation-based technique that we call *stochastic abstraction*. Our technique (developed with VERIMAG Grenoble) starts by performing simulations of the system in order to learn the context/environment in where the application is used. Then, it creates a stochastic abstraction for the application, which takes the context information into account. This smaller model can be verified using efficient techniques such as statistical model checking. We have applied our approach to two industrial case studies that are beyond the scope of existing formal techniques: (1) The HCS case study (2) an *Avionics Full Duplex Switched Ethernet*. Our work has been published in [16], [17], [26].

6.4. Scheduling and Supervisory Control

Participants: Eric Badouel, Philippe Darondeau.

6.4.1. Scheduling

Good scheduling policies for distributed embedded applications are required for meeting hard real time constraints and for optimizing the use of computational resources. We study the *quasi-static scheduling* problem in which (uncontrollable) control flow branchings can influence scheduling decisions at run time [13]. Our abstracted distributed task model consists of a network of sequential processes that communicate via point-to-point buffers. In each round, the task gets activated by a request from the environment. When the task has finished computing the required responses, it reaches a pre-determined configuration and is ready to receive a new request from the environment. For such systems, we prove that determining the existence of a scheduling policy that guarantees upper bounds on buffer capacities is undecidable. However, we show that the problem is decidable for the important subclass of "data-branching" systems in which control flow branchings are exclusively due to data-dependent internal choices made by the sequential components. This decidability result exploits ideas derived from the Karp and Miller coverability tree for Petri nets as well as the existential boundedness notion of languages of message sequence charts.

6.4.2. Supervisory Control for Opacity

In the field of computer security, a problem that received little attention so far is the enforcement of confidentiality properties by supervisory control. Given a critical system G that may leak confidential information, the problem consists in designing a controller C, possibly disabling occurrences of a fixed subset of events of G, so that the closed-loop system G/C does not leak confidential information. We consider this problem [14] in the case where$G$ is a finite transition system with set of events Σ and an inquisitive user, called the adversary, observes a subset Σ_a of Σ. The confidential information is the fact (when it is true) that the trace of the execution of G on Σ^* belongs to a regular set $S \subseteq \Sigma^*$, called the secret. The secret S is said to be opaque w.r.t. G (resp. G/C) and Σ_a if the adversary cannot safely infer this fact from the trace of the execution of G (resp. G/C) on Σ_a^*. In the converse case, the secret can be disclosed. We present an effective algorithm for computing the most permissive controller C such that S is opaque w.r.t. G/C and Σ_a. This algorithm subsumes two earlier algorithms by the same authors, working under the strong assumption that the alphabet Σ_a of the adversary and the set of events that the controller can disable are comparable.

6.4.3. Supervisory Control for Services

In the service oriented architecture framework, Larsen's modal specifications enriched with final states may be used to formalise how a service should interact with its environments. A modal specification determines the events that the server *may* or *must allow* at each stage in an interactive session. Final states increase significantly the expressive power of modal specifications and serve to mark the possible *ends* of sessions. We investigate in [23] the adaptation of the supervisory control theory of Ramadge and Wonham to enforce a modal specification on a system modelled by a finite labelled transition system. We prove that there exists at most one most permissive solution to this control problem We also prove that this solutuion is regular and we present an algorithm for the construction of the corresponding controller.

6.4.4. Delay Controllers

Jan Komenda (Czech Academy of Sciences, Branch in Brno), Philippe Darondeau, Anne Bouillard, and Eric Badouel have started joint work and written a draft paper upon Residuation of Tropical Series: Rationality Issues. In this work, we show an effective decision of existence and computation of robust delay controllers for timed systems modelled with rational power series.

6.5. Games, Logic and System Synthesis

Participants: Axel Legay, Bastien Maubert, Sophie Pinchinat, Lois Vanhée.

6.5.1. Preorder Checking

We investigate [21] the complexity of preorder checking when the specification is a flat finite-state system whereas the implementation is either a non-flat finite-state systemor a standard timed automaton. In both cases, we show that simulation checking is EXP TIME-hard, and for the case ofa non-flat implementation, the result holds even if there is no synchronization between the parallel components and their alphabets of actions are pairwise disjoint. Moreover, we show that the considered problems become PSPACE-complete when the specification is assumed to be deterministic. Additionally, we establish that comparing a synchronous non-flat system with no hiding and a flat system is PSPACE-hard for any relation between trace containment and bisimulation equivalence.

6.5.2. Future Event Logic

We presented in [25] a sound and complete axiomatization of future event logic. Future event logic is a logic that generalizes a number of dynamic epistemic logics, by using a new operator that acts as a quantifier over the set of all refinements of a given model. (A refinement is like a bisimulation except that from the three relational requirements only 'atoms' and 'back' need to be satisfied.) Thus the logic combines the simplicity of modal logic with some powers of monadic second order quantification. We prove the axiomatization is sound and complete and discuss some extensions to the result.

6.5.3. Winning Coalition in Multi-Agent Systems

We consider in [27] the setting of multi-agent systems designed to achieve an objective e.g. a multi-node distributed system that has to perform a task in a bounded amount of time. Whatever the objective is, a way to enforce it can be to restrict the behaviour of a subset of its agents; we call this subset a coalition. Depending on the initial configuration of the system, the set of winning coalitions may vary. we propose an algorithm which computes the set of winning coalitions for any configuration.

6.5.4. Analysis of partially observed recursive discrete-event systems

Monitoring of recursive discrete-event systems under partial observation is an important issue with major applications such as the diagnosability of faulty behaviors and the detection of information flow. We consider regular discrete-event systems, that is recursive discrete-event systems definable by deterministic graph grammars. This setting is expressive enough to capture classical models of recursive systems such as the pushdown systems. Hence they are infinite-state in general and standard powerset constructions for monitoring do not apply anymore. We exhibit computable conditions on these grammars together with non-trivial transformations of graph grammars that enable us to construct a monitor. This construction is applied to diagnose faulty behaviors and to detect information flow in regular discrete-event systems.

7. Other Grants and Activities

7.1. Combest: Component-Based Embedded Systems Design Techniques

Participants: Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Axel Legay.

IST STREP 215543, January 1st 2008 - December 31st 2010 http://www.combest.eu/home/

The objective of the Combest project is to provide a formal framework for component based design of complex embedded systems. This framework will:

- Enable formal integration of heterogeneous components, such as with different models of communication or execution;
- Provide complete encapsulation of components both for functional and extra-functional properties and develop foundations and methods ensuring composability of components;
- Enable prediction of emergent key system characteristics such as performance and robustness (timing, safety) from such characterizations of its subcomponents;
- Provide certificates for guarantees of such key system characteristics when deployed on distributed HW-architectures

To achieve these objectives, Combest will:

- Develop a design theory for complex embedded systems, fully covering heterogeneity, interface specifications, composability, compositionality, and refinement for functional and extra-functional properties;
- Build on substantial highly recognized background results of the academic partners, partly carried out within the integrated project Speeds;
- Extend results of the Integrated Project Speeds, both regarding heterogeneous rich components and compositional analysis methods.

In 2010, some of our research activity on reactive components has taken place in the framework of the Combest Project. This include research on interface theories in collaboration with Kim Larsenn at Aalborg University and Andrzej Wasowski at ITU Copenhagen. Benoît Caillaud, Bastien Maubert, and Jean-Baptise Raclet have developed the InterSMV modal-interface verification toolset (Section 5.1). Axel Legay, Benoît Caillaud, and Benoit Delahaye have collaborated with VERIMAG and they have proposed a statistical model checking procedure to verify the HCS case study submitted by EADS to the consortium.

7.2. Disc: Distributed Supervisory Control of Large Plants

Participant: Philippe Darondeau.

ICT STREP 224498 Disc (September 2008 to October 2011), http://www.disc-project.eu

Started on 1 September 2008, Disc is a project supported by the ICT program of the European Union.

The aim of the project is to enable the supervisory control of networked embedded systems. These distributed plants are composed by several local agents that take concurrently decisions, based on information that may be local or received from neighbouring agents; they require scalable and self-organising platforms for advanced computing and control. The evolution is guided by the occurrence of asynchronous events, as opposed to other real-time models where the event occurrence is time-triggered.

The partners of the project come from academia (University of Cagliari, CWI - Amsterdam, Ghent University, Technical University of Berlin, University of Zaragoza, INRIA, Czech Academy of Sciences), from industry (Akhela s.r.l., Italy and CyBio AG, Germany), and from a governmental instance (Ministry of the Flemish Government, Belgium).

Philippe Darondeau has worked in this context with Eric Badouel, Anne Bouillard and Jan Komenda (Czech Academy of Sciences, Brno) on the synthesis of robust delay-controllers for timed systems modelled with rational power series. He works also towards applying the synthesis of distributable Petri nets to asynchronous and distributed supervisory control.

7.3. Synchronics: Language Platform for Embedded System Design

Participants: Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Axel Legay.

Large initiative action funded by INRIA. http://synchronics.inria.fr/

This project, started Jan 1st 2008, is supported by INRIA. It capitalizes on recent extensions of data-flow synchronous languages (mode automata, Lucid Synchrone, Signal, Lustre, ReactiveML, relaxed forms of synchronous composition or compilation techniques for various platforms). We aim to address the main challenges of embedded system design, starting from a single, semantically well founded programming language.

Our contribution in 2010 is detailed in Section 6.2. A detailed account of the work carried out in Synchronics can be found in the slides presented during the mid-term evaluation seminar of the action: http://synchronics.inria.fr/doku.php/mid-term-review

7.4. ARC (TP)I

Participants: Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Axel Legay.

Website of ARC TP(I): http://arctpi.inria.fr/

The objective of ARC (TP)I started in 2010 is to extend interfaces theories developed in the COMBEST project. This year, Axel Legay has collaborated with Kim Larsen and Andrzej Wasowski to propose new interfaces theories for timed systems. Our results have been implemented in a tool called ECDAR (see `http://www.cs.aau.dk/~adavid/ecdar/download.html`). In addition, the above mentioned researchers have also collaborated with Benoît Caillaud, Benoît Delahaye, and Joost-Pieter Katoen from Aachen to develop a new theory for stochastic system. Our theory is currently being implemented in a new tool called APAC (see `http://www.cs.aau.dk/~mikkelp/apac/`).

8. Dissemination

8.1. Teaching, Committee Work and Organization of Scientific Events

- Eric Badouel is the secretary of the steering committee of CARI, the African Conference on Research on Computer Science and Applied Mathematics. He takes part in the programme committee and in the organizing committee of CARI 2010. He is a member of the editorial board of the ARIMA Journal. He has taught an advanced course on functional programming and the manipulations of structured documents in the Second year of the Master of Research in Computer Science, Université Cheikh Anta Diop (UCAD) in Dakar, Senegal.
- Albert Benveniste is associated editor at large (AEAL) for the journal *IEEE Trans. on Automatic Control*. He is member of the Strategic Advisory Council of the Institute for Systems Research, Univ. of Maryland, College Park, USA. He belongs to the Scientific Advisory Board of INRIA, where he is in charge of the area of Embedded Systems.[1]
- Benoît Caillaud has been serving on the steering and program committees of the International Conference on Application of Concurrency to System Design (ACSD). He has also served on the program committee of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010).

[1] Only facts related to the activities of Team S4 are mentioned. Other roles or duties concern the DistribCom or Sisthem teams, to which A. Benveniste also belongs.

- Philippe Darondeau gave an invited lecture entitled "The Synthesis of Petri Nets" at the workshop ART 2010 (Applications of Region Theory) organized by J"org Desel and Alex Yakovlev in June in Braga (Portugal) as a satellite of the conference ACSD 2010. He participated in the juries for the PhD defenses of Jeremy Dubreil (Inria Rennes) and Christelle Braun (Inria Orsay). He served on the programme committees for the workshops ART 2010 (Braga), APNOC 2010 (Braga), and Wodes 2010 (Berlin). He is the secretary of the IFIP WG2.2 working group.
- Axel Legay ...
- Sophie Pinchinat Organized a workshop on "Games with Imperfect Information for Privacy and Security" at IRISA in Rennes (GIPSy: http://www.irisa.fr/prive/Sophie.Pinchinat/GIPSy/gipsy10.html). She gave a talk on “A Theory of Interfaces” at the Workshop “Formal Theories of Communication”, Lorentz Center, Leiden, NL, February 2010, on “A Theory of Timed Interfaces” at the ANR DOTS Meeting, LaBRI, and on "Future event logic - axioms and complexity" at Advances in Modal Logic 2010, 24-27, August 2010. Bordeaux, March 2010. She was Lecturer at the the Logic Summer School, ANU Canberra (http://lss.rsise.anu.edu.au/), 5 hours on "Logic, Games and Automata". She acts as expert at the Europen Commision for a TECHNOLOGICAL DEVELOPMENT AND INNOVATION project. She was appointed as Recruitment Board Member for CR1 et CR2 INRIA Rennes. She took place in the Ph.D. jury of Mehdi Talbi (université de Rennes 1).

9. Bibliography

Major publications by the team in recent years

[1] E. BADOUEL, M. BEDNARCZYK, A. BORZYSZKOWSKI, B. CAILLAUD, P. DARONDEAU. *Concurrent Secrets*, in "Discrete Event Dynamic Systems", December 2007, vol. 17, n^{o} 4, p. 425-446, http://dx.doi.org/10.1007/s10626-007-0020-5.

[2] E. BADOUEL, M. BEDNARCZYK, P. DARONDEAU. *Generalized Automata and their Net Representations*, H. EHRIG, G. JUHÀS, J. PADBERG, G. ROZENBERG (editors), Lecture Notes in Computer Science, Springer, 2001, vol. 2128, p. 304–345, http://link.springer.de/link/service/series/0558/bibs/2128/21280304.htm.

[3] E. BADOUEL, B. CAILLAUD, P. DARONDEAU. *Distributing Finite Automata through Petri Net Synthesis*, in "Journal on Formal Aspects of Computing", 2002, vol. 13, p. 447–470, http://dx.doi.org/10.1007/s001650200022.

[4] E. BADOUEL, P. DARONDEAU. *Theory of regions*, in "Lectures on Petri Nets I: Basic Models", Lecture Notes in Computer Science, Springer, 1999, vol. 1491, p. 529–586.

[5] A. BENVENISTE, B. CAILLAUD, LUCA P. CARLONI, P. CASPI, ALBERTO L. SANGIOVANNI-VINCENTELLI. *Composing heterogeneous reactive systems*, in "ACM Trans. Embedded Comput. Syst.", 2008, vol. 7, n^{o} 4, http://doi.acm.org/10.1145/1376804.1376811.

[6] A. BENVENISTE, B. CAILLAUD, P. LE GUERNIC. *Compositionality in dataflow synchronous languages: specification and distributed code generation*, in "Information and Computation", 2000, vol. 163, p. 125-171.

[7] B. CAILLAUD, P. DARONDEAU, L. HÉLOUËT, G. LESVENTES. *HMSCs as specifications... with PN as completions*, F. CASSEZ, C. JARD, B. ROZOY, M. DERMOT (editors), Lecture Notes in Computer Science, Springer, 2001, vol. 2067, p. 125–152, http://www.irisa.fr/s4/download/papers/hmsc2pn_movep2k_lncs.ps.gz.

[8] G. FEUILLADE, S. PINCHINAT. *Modal Specifications for the Control Theory of Discrete-Event Systems*, in "Discrete Event Dynamic Systems", 2007, vol. 17, n^{o} 2, p. 211–232, http://dx.doi.org/10.1007/s10626-006-0008-6.

[9] D. POTOP-BUTUCARU, B. CAILLAUD. *Correct-by-Construction Asynchronous Implementation of Modular Synchronous Specifications*, in "Fundamenta Informaticae", 2007, vol. 78, n^{o} 1, p. 131–159.

[10] S. RIEDWEG, S. PINCHINAT. *Quantified Mu-Calculus for Control Synthesis*, in "MFCS 2003, 28th International Symposium on Mathematical Foundations of Computer Science", Lecture notes in computer science, Springer, aug 2003, vol. 2747, p. 642–651, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2747&spage=642.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] B. DELAHAYE. *Modular Specification and Compositional Analysis of Stochastic Systems*, Université de Rennes 1, 2010.

[12] B. T. FOTSING. *Les grammaires attribuées pour la conception et l'assemblage de langages dédiés*, Université de Rennes I et Université de Yaoundé I, december 2010.

Articles in International Peer-Reviewed Journal

[13] P. DARONDEAU, B. GENEST, P. THIAGARAJAN, S. YANG. *Quasi-static scheduling of communicating tasks*, in "Information and Computation", 2010, vol. 208, n^{o} 10, p. 1154 – 1168, Special Issue: 19th International Conference on Concurrency Theory (CONCUR 2008).

[14] J. DUBREIL, P. DARONDEAU, H. MARCHAND. *Supervisory Control for Opacity*, in "IEEE Transactions on Automatic Control", May 2010, vol. 55, n^{o} 5, p. 1089–1100.

[15] J.-B. RACLET, E. BADOUEL, A. BENVENISTE, B. CAILLAUD, A. LEGAY, R. PASSERONE. *A Modal Interface Theory for Component-based Design*, in "Fundamenta Informaticae", 2010, to appear.

International Peer-Reviewed Conference/Proceedings

[16] A. BASU, S. BENSALEM, M. BOZGA, B. CAILLAUD, B. DELAHAYE, A. LEGAY. *Statistical Abstraction and Model-Checking of Large Heterogeneous Systems*, in "Formal Techniques for Distributed Systems, Joint 12th IFIP WG 6.1 International Conference, FMOODS 2010 and 30th IFIP WG 6.1 International Conference, FORTE 2010", Amsterdam, The Netherlands, J. HATCLIFF, E. ZUCCA (editors), Lecture Notes in Computer Science, Springer, June 7-9 2010, vol. 6117, p. 32–46.

[17] A. BASU, M. BOZGA, S. BENSALEM, B. DELAHAYE, A. LEGAY, E. SIFAKIS. *Verification of an AFDX Infrastructure using Simulations and Probabilities*, in "Proc. 1st International Conference on Runtime Verification", Malta, Lecture Notes in Computer Science, Springer-verlag, 2010.

[18] A. BENVENISTE, B. CAILLAUD, M. POUZET. *The Fundamentals of Hybrid Systems Modelers*, in "49th IEEE Conference on Decision and Control (CDC 2010)", IEEE Computer Society, 2010.

[19] E. BEST, P. DARONDEAU. *Separability in Persistent Petri Nets*, in "Applications and Theory of Petri Nets, ATPN 2010", Braga, Portugal, June 2010, p. 246–266.

[20] E. BEST, P. DARONDEAU, R. GORRIERI. *On the Decidability of Non Interference over Unbounded Petri Nets*, in "8th International Workshop on Security Issues in Concurrency, SecCo 2010", Paris, France, August 2010.

[21] L. BOZZELLI, S. PINCHINAT, A. LEGAY. *Hardness of preorder checking for basic formalisms*, in "Proceedings of the 16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR-16", Dakar, Senegal, Lecture Notes in Articifial Intellingence, Springer, 2010, vol. 6355.

[22] B. CAILLAUD, B. DELAHAYE, KIM G. LARSEN, A. LEGAY, M. L. PEDERSEN, A. WASOWSKI. *Compositional design methodology with constraint Markov chains*, in "Proceedings of the 7th International Conference on Quantitative Evaluation of SysTems (QEST) 2010", IEEE Computer Society, 2010.

[23] P. DARONDEAU, J. DUBREIL, H. MARCHAND. *Supervisory Control for Modal Specifications of Services*, in "Workshop on Discrete Event Systems, WODES 2010", Berlin, Germany, August 2010, p. 428–435.

[24] B. DELAHAYE, B. CAILLAUD, A. LEGAY. *Probabilistic Contracts : A Compositional Reasoning Methodology for the Design of Stochastic Systems.*, in "Proc. 10th International Conference on Application of Concurrency to System Design (ACSD)", Braga, Portugal, IEEE, 2010.

[25] H. VAN DITMARSCH, T. FRENCH, S. PINCHINAT. *Future Event Logic - axioms and complexity*, in "Proceedings of Advances in Modal Logic", M. RUSSIA (editor), Lecture Notes in Computer Science, College Publications, 2010.

Other Publications

[26] S. BENSALEM, B. DELAHAYE, A. LEGAY. *Statistical Model Checking: Present and Future: a tutorial*, in "Proc. 1th International Conference on Runtime Verification, Malta", Lecture Notes in Computer Science, Springer-verlag, 2010.

[27] L. VAHNE, S. PINCHINAT. *Computing the winning coalitions of a turn-based game with reachability objectives*, 2010, 2nd Workshop on Games for Design, Verification and Synthesis. colocated with CONCUR'10.

References in notes

[28] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *Dividing and cycling: types and compilation for a hybrid synchronous language*, in "Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, Tools and Theory for Embedded Systems (LCTES) 2011", 2011, to appear.

[29] A. BENVENISTE, B. CAILLAUD, A. FERRARI, L. MANGERUCA, R. PASSERONE, C. SOFRONIS. *Multiple Viewpoint Contract-Based Specification and Design*, in "Proceedings of the Software Technology Concertation on Formal Methods for Components and Objects (FMCO'07)", Amsterdam, The Netherlands, Revised Lectures, Lecture Notes in Computer Science, Springer, October 2008, vol. 5382.

[30] J.-B. RACLET. *Quotient de spécifications pour la réutilisation de composants*, École doctorale Matisse, université de Rennes 1, 2007.

[31] L. DE ALFARO, T. A. HENZINGER. *Interface Automata*, in "Proceedings of the Ninth Annual Symposium on Foundations of Software Engineering", ACM Press, 2001, p. 109–120.