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 Scientist

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 Member

Sophie Pinchinat [Associate Professor, HdR]

Technical Staff

Bastien Maubert [Software Engineer, started 1 September 2009, funded by the COMBEST european project]

PhD Student

Benoît Delahaye [Teaching Assistant, University of Rennes 1]
Rodrigue Djeumen [Funded by SCAC Yaoundé (Service de Coopération et d'Action Culturelle de l'Ambassade de France), part time in France]
Bernard Fotsing [Funded by AUF (Agence universitaire de la Francophonie), part time in France]
Rodrigue Tchougong Ngongang [Funded by SARIMA, part time in France]
Maurice Tchoupé [Funded by SCAC Yaoundé (Service de Coopération et d'Action Culturelle de l'Ambassade de France), part time in France]

Post-Doctoral Fellow

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

Visiting Scientist

Eike Best [Visiting scientist, February 2009]

Administrative Assistant

Laurence Dinh [TR, part-time in S4]

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:

- We have focused on developing a rich composition algebra for modal interfaces which meets certain methodological requirements. Modal interfaces [29], [28] correspond to deterministic automata whose transitions are both typed with may and must modalities and input or output attributes. In [20], [19], we have proposed a timed extension of modal specications, defined their notions of refinement and consistency, and established their decidability.
- we have developed InterSMV, a tool capable of analyzing modal interface specifications. The InterSMV tool is a front end to the open-source symbolic model-checker NuSMV, which explains its name. The tool reduces interface satisfaction and refinement problems into classic model-checking problems, to be solved by NuSMV.
- We have studied the *quasi-static scheduling* problem in which (uncontrollable) control flow branchings can influence scheduling decisions at run time [17]. We have proved that determining the existence of a scheduling policy that guarantees upper bounds on buffer capacities is undecidable for networks of sequential processes that communicate via point-to-point buffers. 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.

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 European project, Section 7.3).

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 tool for the verification of symbolic modal interfaces

Participants: Benoît Caillaud, Bastien Maubert.

In 2009, we have developed InterSMV, a tool capable of analyzing modal interface specifications [29], [28]. In order to support an interaction semantics based on shared variables, it has been necessary to extend the theory to convex acceptance interfaces, a trade-off between expressiveness, closure properties of the algebra, effectiveness of symbolic representations and algorithmic complexity. Convex acceptance interfaces strictly extend modal interfaces and are a subset of acceptance specifications, a formalism studied in J.-B. Raclet's PhD work [38], [37]. Convex acceptance interfaces have three advantages over modal interfaces: 1/ they allow for a synchronous reactive semantics where transition labels are valuations of several interface variables, 2/ it is possible to express non-blocking properties, 3/ composition operators have very simple mathematical definitions, making then somewhat easier to implement and 4/ composition operators have low algorithmic complexities on symbolic representations of convex acceptance interfaces.

The InterSMV tool is a front end to the open-source symbolic model-checker NuSMV, which explains its name. The tool reduces interface satisfaction and refinement problems into classic model-checking problems, to be solved by NuSMV. The tool proceeds by model transformations, where interface specifications are mapped to NuSMV code and relations to be checked are transformed into properties to be checked against the derived NuSMV models. The main benefit from this approach is that the power of the NuSMV symbolic model-checker can be used, without having to implement complex algorithms directly at the level of BDD (Binary Decision Diagrams) representations of interface specifications. This allows specifying interfaces with variables of finite but complex data-types.

6. New Results

6.1. Petri nets and their Synthesis

Participants: Eric Badouel, Philippe Darondeau.

6.1.1. Petri net synthesis

We have proposed in [30] a framework in which the synthesis of Petri nets from products of regular languages may be dealt with in a modular way, without evaluating any global language.For this purpose, we focused on distributed Petri nets, made of subnets residing in different sites of a communication network. The behaviour of each component is specified by a regular language on the union of the alphabets of this component and the components immediately upstream.

The unconstrained step semantics of Petri nets is impractical for simulating and modelling applications. In the past, this inadequacy has been alleviated by introducing various flavours of maximally concurrent semantics, as well as priority orders. We have introduced in [18] a general way of controlling step semantics of Petri nets through step firing policies that restrict the concurrent behaviour of Petri nets and so improve their execution and modelling features. In a nutshell, a step firing policy disables at each marking a subset of enabled steps which could otherwise be executed. We discuss various examples of step firing policies and then investigate the synthesis problem for Petri nets controlled by such policies. Using generalised regions of step transition systems, we provide an axiomatic characterisation of those transition systems which can be realised as reachability graphs of Petri nets controlled by a given step firing policy. We also provide two different decision and synthesis algorithms for PT-nets and step firing policies based on linear rewards of 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.

6.1.2. Petri net decomposition

We consider finite labelled transition systems. We show in [12] that if such transition systems are deterministic, persistent and weakly periodic, then they can be decomposed in the following sense. There exists a finite set of label disjoint cycles such that any other cycle is Parikh equivalent to a multiset from this set.

6.2. Heterogeneous systems

Participants: Eric Badouel, Rodrigue Djeumen, Bernard Fotsing, Rodrigue Tchougong Ngongang, Maurice Tchoupé.

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. Extending the preliminary work in [36], [35], we have in [11] considered manipulation of hierarchically-structured documents within a complex workflow system. Such a system may consist of several subsystems distributed over a computer network. These subsystems can concurrently update partial views of the document. At some points in time we need to reconcile the various local updates by merging the partial views into a coherent global document. For that purpose, we represent the potentially-infinite set of documents compatible with a given partial view as a coinductive data structure. This set is a regular set of trees that can be obtained as the image of the partial view of the document by the canonical morphism (anamorphism) associated with a coalgebra (some kind of tree automaton). Merging partial views then amounts to computing the intersection of the corresponding regular sets of trees which can be obtained using a synchronization operation on coalgebras.

6.3. Reactive components

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

Interfaces offer two fundamental properties that are essential to component based engineering, namely stepwise refinement and substituability. Stepwise refinement allows to replace, in any context, an interface by a more detailed version of it - a refinement. Substituability (also referred to as "independent implementability") allows to implement every interface regardless of its context of use.

In 2001, de Alfaro and Henzinger introduced Interface Automata which are I/O automata expressing assumptions on the environment and guarantees on the component's behaviour. Refinement is by alternating simulation, which amounts to getting more permissive regarding the environment and more constrained regarding its behaviour. Using background work on modal automata, Kim Larsen and coworkers have shown that the framework of Interface Automata is naturally embedded into that of Modal I/O Automata where alternating simulation appears as a particular case of modal refinement. Roughly speaking, Modal Automata are automata in which some transitions are labelled must and other are labelled may. Any implementation should always enable any must transition, whereas may transitions may or may not be enabled. Modal refinement is a simulation relation reflecting inclusion of implementations. With Modal I/O Automata one can express liveness properties, such that the trivial implementation that exhibits no behaviour at all can be disallowed. Of course we can equivalently adopt a language theoretic approach considering modal specifications rather than modal automata. In his thesis Jean-Baptiste Raclet introduced a residuation on modal specifications so that the implementations of the residual specification S/S' are those automata that when composed with an arbitrary implementation of S' constitute an implementation of S. The residuation of specifications approach to component reuse allows contract-based reasoning at system design level, refinement (specialization preorder), top-down design (quotient), bottom-up design (product), and shared refinement (greatest lower bound).

6.3.1. Modal Interfaces

We have focused on developing a rich composition algebra for modal interfaces which meets certain methodological requirements. We have experimented this methodology with InterSMV (Section 5.1), a symbolic modal interface frontend to NuSMV.This work was essentially done in the context of the SPEEDS and COMBEST european projects.

Modal interfaces [29], [28] correspond to deterministic automata whose transitions are both typed with may and must modalities and input or output attributes. A modal interface thus represents a set of models; informally, a must transition is available in every component that implements the modal interface, while a may transition needs not be. Satisfiability of modal interfaces is decidable. Refinement between modal interfaces coincides with model inclusion. Conjunction is effectively computed via a product-like construction. Inconsistencies between contradictory interfaces can be handled thought a simple mechanism. It can be shown that the conjunction of two modal specifications correspond to their greatest common refinement. Combination of modal specifications, handling synchronization products à la Arnold and Nivat, and the dual quotient combinators can be efficiently handled in this setting. When building a system by combining interfaces on dissimilar alphabets, we proceed by equalization of alphabets and modalities appear as an elegant solution offering the appropriate flexibility; alphabet extension is performed by setting the specific modalities for added self-loops, specifically: may for the case of the conjunction, and must for the case of the parallel composition. Last, by correction the bug of Larsen and al. in 2007, the compatibility issue for interface automata can be generalized to modal interfaces. This operation ensures independent implementability: two compatible modal interfaces may be implemented separately and then composed, the resulting system will be an implementation of the composed interfaces.

6.3.2. Timed aspects of Interface Theories

Time is a crucial aspect of systems e.g. in the area of embedded systems. And yet, only few results exist on the design of timed component-based systems. In [20], we have proposed a timed extension of modal specications, defined their notions of refinement and consistency, and established their decidability. In [19], we have considered the subclass of modal event-clock automata, where clock resets are easy to handle. We then have developed an entire theory with conjunction, product, and quotient, that promotes efficient incremental design techniques and that enables to reason in a compositional way about timed system.

Simulation preorder is one of the standard mathematical notions used to formalize the "refinement" relation between an abstract version (specification) and a concrete version (implementation) of the same system. In refinement checking, the goal is to ensure that the properties proved about the specification continue to hold in the refined version (i.e., the implementation). This scenario may arise either because the design is being carried out in an incremental fashion, or because the system is too complex and an abstraction needs to be used to verify its properties.

Next, in [22], we have addressed the problem of alternating simulation refinement for concurrent timed games. We have shown that checking timed alternating simulation between TG is EXPTIME-complete, and we have provide a logical characterization of this preorder in terms of a meaningful fragment of a new logic, TAMTL*. This logic is an action-based timed extension of standard alternating-time temporal logic ATL*, which allows to quantify on strategies where the designated player is not responsible for blocking time. While for full TAMTL*, model-checking timed games is undecidable, we show that for its fragment TAMTL, corresponding to the timed version of ATL, the problem is instead in EXPTIME.

6.3.3. Probabilistic aspects of Interface Theories

We introduced a new abstraction, namely Constraint Markov Chains, for probabilistic interfaces [32]. We use this model to construct a specification theory for Markov Chains. Constraint Markov Chains generalize previously known abstractions by allowing arbitrary constraints on probability distributions. Our theory is the first specification theory for Markov Chains closed under conjunction, parallel composition and synchronization. Moreover, all the operators and relations introduced are computable. CMCs could be considered as a finite representation of Assumptions and Guarantees as presented in the probabilistic Assume/Guarantee contracts formalism. This ongoing work is submitted. On the other hand, we have considered probabilistic Assume/Guarantee contracts, which consist in (i) a non deterministic model of components behaviour, and (ii) a stochastic and non deterministic model of systems faults. Two types of contracts capable of capturing reliability and availability properties are considered. We show that Satisfaction and Refinement can be checked by effective methods thanks to a reduction to classical verification problems. We finally present theorems supporting compositional reasoning and enabling the scalable analysis of complex systems.

6.4. Discrete event system synthesis and supervisory control

Participants: Philippe Darondeau, Bastien Maubert, Sophie Pinchinat.

6.4.1. scheduling and supervisory control

6.4.1.1. 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 have studied the *quasi-static scheduling* problem in which (uncontrollable) control flow branchings can influence scheduling decisions at run time [17]. 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 have proved 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.

6.4.1.2. supervisory control

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 [16] 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 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.

6.4.2. Logic, games and opacity

6.4.2.1. Imperfect information

We describe in [25] the class of games with opacity condition, as an adequate model for some security aspects of computing systems. We study their theoretical properties, relate them to reachability perfect information games and exploit this relation to discuss a search approach with heuristics, based on the directing-word problem in automata theory.

Diagnosis problems of discrete-event systems consist in detecting unobservable defects during system execution. For finite-state systems, the theory is well understood and a number of effective solutions have been developed. For infinite-state systems, however, there are only few results, mostly identifying classes where the problem is undecidable. We consider in [27] higher-order pushdown systems and investigate two basic variants of diagnosis problems: the diagnosability, which consists in deciding whether defects can be detected within a finite delay, and the bounded-latency problem, which consists in determining a bound for the delay of detecting defects.

6.4.2.2. Logic and Game

RoCTL* was proposed to model robustness in concur- rent systems. RoCTL* extended CTL* with the addition of Obligatory and Robustly operators, which quantify over failure-free paths and paths with one more failure respec- tively. Whether RoCTL* is more expressive than CTL* has remained an open problem since the RoCTL* logic was proposed. We use the equivalence of LTL to counter-free automata to show that RoCTL* is

expressively equivalent to CTL* [26]; the translation to CTL* provides the first model checking procedure for RoCTL*. However, we show that RoCTL* is relatively succinct as all satisfaction preserving translations into CTL* are non-elementary in lengthThe RoCTL* Obligatory operator is similar to the Oblig- atory operator in Standard Deontic Logic (SDL), although in RoCTL* the operator quantifies over paths rather than worlds. SDL has many paradoxes. Some of these, such as the "Gentle Murderer" paradox spring from the inadequacy of SDL for dealing with obligations caused by acting contrary to duty such as "If you murder, you must murder gently". Contrary-to-Duty (CtD) obligations are important for modeling a robust system, as it is often important to state that the system should achieve some goal and also that if it fails it should in some way recover from the failure.

We analyse in [21] two basic approaches of extending classical logics with quantifiers interpreted via games: Propositional Game Logic of Parikh and Alternating-Time Temporal Logic of Alur, Henzinger, and Kupferman. Although the two approaches are historically remote and they incorporate operationally orthogonal paradigms, we trace the formalisms back to common foundations and argue that they share remarkable similarities in terms of expressive power.

7. Other Grants and Activities

7.1. Synchronics: Language Platform for Embedded System Design

Participants: Timothy Bourke, Benoît Caillaud.

Synchronics is an INRIA large-scale initiative action, started January 2008. Partner team/laboratories are: ALCHEMY, PROVAL (INRIA Saclay - Île-de-France), POPART (INRIA Grenoble - Rhône-Alpes), S4 (INRIA Rennes - Bretagne Atlantique) and VERIMAG. http://synchronics.wiki.irisa.fr/. Synchronics capitalizes on recent extensions of data-flow synchronous languages (mode automata, Lucid Synchrone, Signal, Lustre, Reactive ML, 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. Synchronous languages have demonstrated in the past their pertinence, both from the theoretical point of view and from the industrial point of view.

Nonetheless, the current industrial application domain of synchronous languages is still limited and could be applied to a much wider range of applications, provided that we give answer to several questions, including the co-simulation of mixed discrete-continuous specifications, and more generally the co-simulation of programs and properties (partial specifications, either discrete or continuous). This in turn raises the question of the interaction between programs and various kinds of differential equations solvers, the adequate module systems and typing disciplines to structure the whole system and way to get both efficient code and efficient simulation such that the very same code can be used for both simulation and code generation.

7.2. DOTS: Distributed Open and Timed Systems

Participant: Sophie Pinchinat.

Started in 2007, this is a four year long ANR project with the following partner laboratories: IRISA Rennes, IRCCyN Nantes, LaBRI Bordeaux, LAMSADE Paris-Dauphine, LSV ENS Cachan. http://www.lsv.ens-cachan.fr/anr-dots/?l=en.

The aim of the DOTS project is to associate researchers specialized in verification of different aspects (timing constraints, communication, interaction with an environment,...) in order to tackle problems that emerge when considering several aspects 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.

An important characteristic of the DOTS project is our choice of methods and tools to address the problems mentioned above. We plan to use games to cope with interactive aspects and partial orders to deal with the distributed aspect.

7.3. SPEEDS: Speculative and Exploratory Design in Systems Engineering

Participants: Albert Benveniste, Benoît Caillaud, Bastien Maubert.

Started in May 2006, the Speeds project is a FP6 European integrated project. Speeds is a concerted effort to define a new generation of end-to-end methodologies, processes and supporting tools for safety-critical embedded system design. They will enable the European systems industry to evolve from model-based design of hardware/software systems, towards integrated, component-based construction of complete virtual system models.

Core partners of the project come from both academia (OFFIS, PARADES, Verimag and INRIA), aeronautics (Airbus, SAAB and IAI), the automotive industry (Daimler-Chrysler, Bosch, Knorr Bremse, Magna Powertrain) and tool vendors (Esterel Technologies, Extessy, Telelogic and TNI).

The main objective of the Speeds project is to develop a modeling formalism, the Heterogeneous Rich Component formalism (HRC) [31], capable of supporting not only scalable modular analysis methods for component based design, but also speculative design processes where several teams work in parallel on a design, one team making assumptions about the subsystem designed by another team.

In 2009 we have focused our effort on the development of the InterSMV tool (Section 5.1), a symbolic modal interface frontend to NuSMV with which discrete HRC contracts can be combined and verified (verification of consistency, satisfaction and refinement).

7.4. Combest: Component-Based Embedded Systems Design Techniques

Participants: Eric Badouel, Albert Benveniste, Benoît Caillaud, Philippe Darondeau, Benoît Delahaye, Sophie Pinchinat, Axel Legay, Laura Bozzelli.

IST STREP 215543 Combest (January 2008 to December 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 (see section 7.3, both regarding heterogeneous rich components and compositional analysis methods.

In 2009, most of our research activity on reactive components has taken place in the framework of the Combest project. This includes research on interface theories, in collaboration with Tom Henzinger at IST, Wien, Austria, Kim Larsen at Aalborg University, and Roberto Passerone at University of Trento (see section 6.3.1), probabilistic models of contracts (see section 6.3.3) and timed model of interfaces (see section 6.3.2). Axel Legay has also developed together with IST Austria a new theory to check the robustness of hardware components. Benoit Caillaud and Bastien Maubert have developed the InterSMV modal-interface verification toolset (Section 5.1). Axel Legay 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.

7.5. DISC: Distributed Supervisory Control of Large Plants

Participant: Philippe Darondeau.

ICT STREP 224498 Disc (September 2008 to October 2011), http://www.combest.eu/home/

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).

We plan to use several techniques to reduce the computational complexity that is one of the major obstacles to the technology transfer of supervisory control methodologies to distributed plants. These techniques are: modularity in the modelling and control design phases; coordinating control; modular state identification and modular fault detection based on the design of decentralized observers.

7.6. FAST: Planning Approaches and Software Verification

Participant: Sophie Pinchinat.

This is a Franco-Australian cooperation on Science and Technology (FAST) with Dr Sylvie Thiebaux, Research School Information Sciences and Engineering, Australian National University. This two year long cooperation started in 2008 and is funded on the Hubert Curien program.

This cooperation aims at bridging the gap between automated planning techniques and formal methods that are used for software verification. Although close, these research areas continue to develop almost independently, led by different scientific communities: artificial intelligence for the former and theoretical computer science for the latter. Cross-fertilisation between these fields will result in the injection of new ideas into each of them and in the development of high-performance verification and planning tools beyond the present days capabilities of either of the two fields alone.

8. Dissemination

8.1. Participation to editorial boards and program committees

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 of the organizing committee of CARI. He is a member of the editorial board of the ARIMA Journal.

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 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2009).

Philippe Darondeau Philippe Darondeau has served in the programme committees for APNOC (Abstractions for Petri Nets and Other Models of Concurrency), satellite workshop of ICATPN 2009, Paris, CONCUR 2009 (International Conference on Concurrency Theory), Bologna, and SOFSEM 2010 (International Conference on Current Trends in Theory and Practice of Computer Science), Spindleruv Mlyn. Philippe Darondeau is the secretary of the IFIP WG2.2 working group.

Axel Legay has served as the general chair for INFINITY 2009 (the *11th International Workshop on Verification of Infinite-State Systems*). He also takes part of the national working Groups : GT Jeux and GASICS.

Sophie Pinchinat is Assistant Director of the "Institut de Formation Supérieure en Informatique et Communication" (IFSIC) de l'université de Rennes I in charge of the International Relations of IFSIC. She is serving as public relations associate and science policy advisor on the Advisory Board of the Marie Curie Fellows Association. Sophie Pinchinat takes part to the organization of the 68NQRT seminar series of IRISA, dedicated to software engineering, theoretical computer science, discrete mathematics, and artificial intelligence.

8.2. Teaching

Teaching related to research undertaken in Team S4 is listed below:

- Eric Badouel has taught an advanced course on functional programming in the Second year of the Master of Research in Computer Science, University of Yaoundé 1, Cameroon. He gave a tutorial on Formal Models for Structured Document at a Summer School organized by CIMPA, UCTP and UNESCO at Yaoundé, Cameroon.
- Axel Legay has participated to the Formal Verification course taught by Sophie Pinchinat and Thomas Genet (20h). He was an invited Professor at TU Munich (8h).
- Sophie Pinchinat has taught advanced courses in first and second year of the Master of Research in Computer Science and Master of Bioinformatics of the University of Rennes 1, and theoretical aspects of Computer Science at École Normale Supériérieure de Cachan in Kerlann, Rennes: Formal Verification (12h), Biology and Algorithmics (22h), Computer Science and Game Theory (4h), Verification and Test of Embbeded Systems (10h), Preparation to the Agreggation of Mathematics (104h).

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", vol. 17, n^{o} 4, December 2007, p. 425-446, http://dx.doi.org/10.1007/s10626-007-0020-5.

[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.

[2] E. BADOUEL, M. BEDNARCZYK, P. DARONDEAU. *Generalized Automata and their Net Representations*, H. EHRIG, G. JUHÁS, J. PADBERG, G. ROZENBERG (editors), Lecture Notes in Computer Science, vol. 2128, Springer, 2001, 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", vol. 13, 2002, 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, vol. 1491, Springer, 1999, 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.", vol. 7, n^{o} 4, 2008, 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", vol. 163, 2000, 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, vol. 2067, Springer, 2001, 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", vol. 17, n^{o} 2, 2007, 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", vol. 78, n^{o} 1, 2007, 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, vol. 2747, Springer, aug 2003, p. 642–651, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2747&spage=642.

Year Publications

Doctoral Dissertations and Habilitation Theses

[11] M. TCHOUPÉ TCHENDJI. *Une approche grammaticale pour la fusion de réplicats partiels d'un document structuré : application à l'édition collaborative asynchrone*, Université de Rennes I et Université de Yaoundé I, août 2009, Ph. D. Thesis.

Articles in International Peer-Reviewed Journal

[12] E. BEST, P. DARONDEAU. *A decomposition theorem for finite persistent transition systems*, in "Acta Informatica", vol. 46, n^{o} 3, 2009, p. 237–254, http://dx.doi.org/10.1007/s00236-009-0095-6DE.

[13] F. CANTIN, A. LEGAY, P. WOLPER. *Computing Convex hulls by automata iteration*, in "International Journal of Foundations of Computer Science", vol. 20, n^{o} 4, 2009, p. 647–667 BE .

[14] K. CHATTERJEE, L. DE ALFARO, M. FAELLA, A. LEGAY. *Qualitative Logics and Equivalences for Probabilistic Systems*, in "Logical Methods in Computer Science", 2009 US IT .

[15] E. M. CLARKE, A. DONZÉ, A. LEGAY. *On Simulation-Based Probabilistic Model Checking of Mixed-Analog Circuits*, in "Formal Methods in System Design", 2009 US .

[16] P. DARONDEAU, J. DUBREIL, H. MARCHAND. *Supervisory Control for Opacity*, in "IEEE-Transactions on Automatic Control", 2009, to appear.

[17] P. DARONDEAU, B. GENEST, P. THIAGARAJAN, S. YANG. *Quasi-Static Scheduling of Communicating Tasks*, in "Information and Computation", 2009, to appear SG CN .

[18] P. DARONDEAU, M. KOUTNY, M. PIETKIEWICZ-KOUTNY, A. YAKOVLEV. *Synthesis of Nets with Step Firing Policies*, in "Fundamenta Informaticae", vol. 94, n^{o} 3–4, 2009, p. 275–474 GB .

International Peer-Reviewed Conference/Proceedings

[19] N. BERTRAND, A. LEGAY, S. PINCHINAT, J.-B. RACLET. *A Compositional Approach on Modal Specifications for Timed Systems*, in "Proc. of the 11th International Conference on Formal Engineering Methods (ICFEM'09), Rio de Janeiro, Brazil", Lecture Notes in Computer Science, Springer, December 2009, To appear.

[20] N. BERTRAND, S. PINCHINAT, J.-B. RACLET. *Refinement and Consistency of Timed Modal Specifications*, in "Proc. of the 3rd International Conference on Language and Automata Theory and Applications (LATA'09)", Lecture Notes in Computer Science, vol. 5457, Springer, 2009, p. 152–163.

[21] D. BERWANGER, S. PINCHINAT. *Game Quantification Patterns*, in "Proceedings of the 3rd Indian Conference on Logic and its Applications, ICLA2009, Formal Definitions of Reason Fallacies to Aid Defect Exploration in Argument Gaming, Chennai, India", Lecture Notes in Artificial Intelligence, vol. 5378, Springer, January 2009.

[22] L. BOZZELLI, S. PINCHINAT, A. LEGAY. *On Timed Alternating Simulation for Concurrent Timed Games*, in "IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009), IIT Kanpur, India", Leibniz International Proceedings in Informatics, Schloss Dagstuhl - Leibniz Center of Informatics, December 2009, To appear IT .

[23] A. DONZÉ, G. CLERMONT, C. J. LANGMEAD, A. LEGAY. *Synthesis in Nonlinear Dynamical Systems: Application to Systems Biology*, in "Proc. 13th Annual International Conference on Research in Computational Molecular Biology", Lecture Notes in Computer Science, vol. 5541, Springer, 2009, p. 155–169 US .

[24] A. LEGAY, M. VISWANATHAN. *Simulation + Hypothesis Testing for Model Checking of Probabilistic Systems (tutorial)*, in "Proc. 6th International Conference on Quantitative Evaluation of Systems", IEEE, 2009 US .

[25] B. MAUBERT, S. PINCHINAT. *Games with Opacity Condition*, in "Proceedings of the 3rd INternational Workshop on Reachability Problem, Palaiseau, France", Lecture Notes in Computer Science, vol. 5797, Springer, 2009, p. 166–175, http://dx.doi.org/10.1007/978-3-642-04420-5_16.

[26] J. MCCABE-DANSTED, T. FRENCH, S. PINCHINAT, M. REYNOLDS. *On the Expressivity of RoCTL**, in "Proceedings of the 16th International Symposium on Temporal Representation and Reasoning, Brixen-Bressanone, Italy", July 2009 AU .

[27] C. MORVAN, S. PINCHINAT. *Diagnosability of pushdown systems*, in "Proceedings of the Haifa Verification Conference, HVC2009, Haifa, Israel", Lecture Notes in Computer Science, Springer, October 2009, To appear.

[28] J.-B. RACLET, E. BADOUEL, A. BENVENISTE, B. CAILLAUD, A. LEGAY, R. PASSERONE. *Modal Interfaces: Unifying Interface Automata and Modal Specifications*, in "Proc. of the 9th International Conference on Embedded Software (EMSOFT'09), Grenoble, France", ACM, October 2009 IT .

[29] J.-B. RACLET, E. BADOUEL, A. BENVENISTE, B. CAILLAUD, R. PASSERONE. *Why are modalities good for Interface Theories?*, in "Proc. of the 9th International Conference on Application of Concurrency to System Design (ACSD'09), Augsburg, Germany", IEEE Computer Society Press, 2009, p. 199–127 IT .

Scientific Books (or Scientific Book chapters)

[30] E. BADOUEL, P. DARONDEAU, L. PETRUCCI. *Modular Synthesis of Petri Nets from Regular Languages*, K. LODAYA, M. MUKUND, R. RAMANUJAM (editors), Universities Press (India) Private Limited, Himayatnagar, Hyderabad, 2009, p. 1–21, distributed by CRC Press, Taylor and Francis Group.

[31] A. BENVENISTE, B. CAILLAUD, R. PASSERONE. *Multi-Viewpoint State Machines for Rich Component Models*, P. MOSTERMAN, G. NICOLESCU (editors), CRC Press, 2009.

Research Reports

[32] B. CAILLAUD, B. DELAHAYE, K. G. LARSEN, A. LEGAY, M. L. PEDERSEN, A. WASOWSKI. *Compositional Design Methodology with Constraint Markov Chains*, n^{o} RR-6993, INRIA, 2009, http://hal.inria.fr/inria-00404304/en/, Research ReportDK.

[33] B. CAILLAUD, B. DELAHAYE, K. G. LARSEN, A. LEGAY, M. L. PEDERSEN, A. WASOWSKI. *Compositional Design Methodology with Constraint Markov Chains*, n^{o} RR-6993, INRIA, 2009, http://hal.inria.fr/inria-00404304/en/, Research ReportDK.

[34] B. DELAHAYE, B. CAILLAUD, A. LEGAY. *Compositional Reasoning on (Probabilistic) Contracts*, n^{o} RR-6970, INRIA, 2009, http://hal.inria.fr/inria-00398985/en/, Research Report.

References in notes

[35] E. BADOUEL, M. TCHOUPÉ. *Merging Hierarchically-Structured Documents in Workflow Systems*, in "Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science (CMCS 2008)", Electronic Notes in Theoretical Computer Science, vol. 203, n^{o} 5, Elsevier, June 2008, p. 3–24, http://dx.doi.org/10.1016/j.entcs.2008.05.017.

[36] E. BADOUEL, M. TCHOUPÉ. *Projections et cohérence de vues dans les grammaires algébriques*, in "Revue ARIMA", vol. 8, 2008, http://www-direction.inria.fr/international/arima/.

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

[38] J.-B. RACLET. *Residual for Component Specifications*, in "Proceedings of the 4th International Workshop on Formal Aspects of Component Software, Sophia-Antipolis, France", 2007.