INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team S4

System Synthesis and Supervision, Scenarios

Rennes - Bretagne-Atlantique

THEME COM

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]
Éric Badouel [CR, HdR]
Albert Benveniste [Research Director (DR), part-time in S4, HdR]
Philippe Darondeau [Research Director (DR), HdR]
Sophie Pinchinat [Lecturer, University of Rennes 1, In delegation at INRIA until January 2008. Funded by a Marie Curie European grant, HdR]

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]
Mateus Oliveira [Funded by the SPEEDS european project]
Rodrigue Tchougong [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]

Visiting Scientist

Laurie Ricker [Visiting scientist on sabbatical leave from Mount Allison University, Sackville, Canada, from September 2007 to June 2008]

Administrative Assistant

Laurence Dinh [TR, part-time in S4]

2. Overall Objectives

2.1. Overall Objectives

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) [30] (sequence diagrams and statecharts), the *High-Level Message Sequence Charts* (HMSC) [28] 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.

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* [30] or *Message Sequence Charts* [28]).

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* [28] 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 like *service adaptation* or *component interfacing*. Problems of this kind can be reduced 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 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 (Section 6.3) 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.

Our work on Petri net synthesis and distributed control (Section 5.1) has found applications in various domains such as automated production systems (in particular, flexible production cells, in collaboration with Team MACSI of INRIA Lorraine) and work-flow engineering.

5. New Results

5.1. Petri nets and their Synthesis

Keywords: *Petri net*, *synthesis of Petri nets*.

Participant: Philippe Darondeau.

5.1.1. Analysis of Cycles of Petri Nets

We have shown in [17] the following property of the reachability graphs of bounded and persistent Petri nets: there exists in such reachability graphs a finite set of cycles over disjoint subsets of transitions, such that any cycle through any reachable marking is Parikh-equivalent to a multiset of cycles from this set. These results have been obtained in a cooperation with Eike Best (University Carl von Ossietzky, Oldenburg, DE).

5.1.2. Synthesis of Petri nets with Step Firing

The unconstrained step semantics of Petri nets is impractical for simulating and modelling applications, owing to the combinatorial explosion of possible steps. In the past, this inadequacy has been alleviated by introducing various flavours of maximally concurrent semantics, as well as priority orders.

In [21] we have proposed a generic way of controlling step semantics of Petri nets by imposing step firing policies that restrict their concurrent behaviour, and thus improve their execution and modelling features, e.g. for membrane systems. In a nutshell, a step firing policy disables at each marking a subset of enabled steps which could otherwise be executed. We have investigated the synthesis of Petri nets controlled by such policies. Using generalised regions of step transition systems, we have provided an axiomatic characterisation of those step transition systems which can be realised as the reachability graphs of Petri nets controlled by step firing policies. We have provided a decision and synthesis algorithm for PT-nets and step firing policies based on linear rewards of steps, where fixing rewards of transitions is part of the synthesis problem. We have also provided a net synthesis algorithm for the general case where the rewards of steps depend linearly on the markings. These results have been obtained in a cooperation with Maciej Koutny, Marta Pietkiewisz-Koutny and Alex Yakovlev (Newcastle University, Newcastle, UK).

Another way to define constrained step semantics of Petri nets was proposed ten years ago by Bruni and Montanari. The principle is to distinguish between two types of places of nets, zero-places and stable places. A marking in which every zero-place is empty is called a stable marking. A stable step is then a sequence of transitions from a stable marking to a stable marking, such that tokens produced and fed into stable places during the step are not used until the completion of the step. A possible field of application is Workflows in which resources (or documents or goods) are taken and released (or consumed and produced) by transactions. In [13], we have proposed a polynomial time synthesis algorithm for Zero-Safe nets from finite "linear step automata" $A = (Q, L, q_0)$ where Q is the set of states, q_0 is the initial state, and $L : Q \times Q \to \mathcal{P}(T^*)$ defines for all states q and q' a (possibly empty) regular set $L(q, q')$ of non-empty sequences of micro-steps from q to q'.

5.2. Heterogeneous systems

Keywords: *Heterogeneous systems*, *coherence of views*, *desynchronization*, *endochrony*, *isochrony*, *partial views*.

Participants: Éric Badouel, Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Philippe Darondeau, Rodrigue Djeumen, Bernard Fotsing, Rodrigue Tchougong, Maurice Tchoupé.

5.2.1. Loosely Time Triggered Architectures

The Time Triggered Architecture (TTA) proposed by H. Kopetz for the communication mechanism of a distributed platform offers a direct mapping that would preserve the semantics of the specification. However, its exact implementation may, at times, be problematic as it requires the distributed platform to have the clocks of its components synchronized with great precision. We propose as implementation architecture a relaxation of TTA called Loosely Time-Triggered Architecture (LTTA), in which writes and reads on distributed shared variables are scheduled by quasi-periodic, but non synchronized local clocks. LTTA offers some of the advantages of TTA with lower hardware cost and greater flexibility.

In previous work, LTTA has been studied on uni-directional communications and two-nodes architectures only. In [11], the authors address the problem of mapping a set of processes which communicate synchronously on a distributed platform. The authors propose a design flow that ensures semantics preservation for LTT communication networks with arbitrary topology. The proposed approach uses an Event Graph abstraction and a token based mechanism. No assumption regarding the local clocks is needed. This work bears close resemblance with the nition of *elastic circuits* introduced by Cortadella [37], [36] in the EDA area.

In [18] a different direction was considered in which physical time is used, however in a relaxed way as compared to Kopetz' strict TTA discipline. By relying on assumptions on the relative drift of clocks and using counter based local protocols, the application semantics can be preserved. This is close to the approach followed at Airbus.

While the first approach is fully adaptive and quite efficient in terms of throughput, the second approach is much better at offering fault isolation and fault-tolerance.

5.2.2. Partial Views

5.2.2.1. Message Sequence Charts

In 2005, we addressed the problem of assembling partial views of the behavior of a distributed system, and we introduced for this purpose an operation of controlled product of languages of (compositional) message sequence charts (MSC). This product composition amounts to interleave separately for each process the send/receive events of the two components, and to amalgamate their synchronized events, without ever introducing circularity in the resulting flow. In this context, we focussed on existentially bounded MSCs, that seem especially interesting since no implementation can reasonably be envisaged for MSCs outside this class. In [19], we have answered the main decision problems concerning products of MSCs and existential boundedness. Whether the controlled product of two existentially bounded languages of MSCs is existentially bounded is undecidable. However, this problem is decidable when the two component MSCs synchronize on exactly one process. The construction of a (c)MSC graph generating the product of two MSC languages has been fully defined in this case. These results have been obtained in a cooperation with Blaise Genest and Loïc Hélouët (DISTRIBCOM team-project).

5.2.2.2. The Design of Heterogeneous 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. Extending the preliminary work in [12], we have in [15] considered the 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.

These documents are represented as tree decorated with attributes. Functional dependencies between attributes can be used when synchronizing the partial views to allows automatic flow of information between the various aspects. Evaluation of attributes w.r.t. an attribute grammar can be obtained by inductively computing a function expressing the dependencies of the synthesized attributes on inherited attributes. This higher-order functional approach to attribute grammars leads to a straightforward implementation using a higher-order lazy functional language like Haskell. The resulting evaluation functions are, however, not easily amenable to optimization rules. We have presented in [14] an alternative first-order functional interpretation of attribute grammars where the input tree is replaced with an extended cyclic tree each node of which is aware of its context viewed as an additional child tree. By the way, we demonstrated that these cyclic representations of zippers (trees with their context) are natural generalizations of doubly-linked lists to trees over an arbitrary signature.

Local views of documents are updated through dedicated editors in combination with domain specific tools. An editor is an interactive program which records the various modifications on the edited documents during its execution. In [23] we use the state monad in combination with the monad of input/output to define an editor in a functional manner. We also define a domain-specific language (DSL) embedded in Haskell (in the form of a set of functional operators) which allows to combine tree-structured data editors. The data can be edited through an abstract view obtained by projection of the concrete structure. The modification of the abstract view implies the propagation of the updates on the concrete representation of the document.

5.3. Reactive components

Keywords: *behavioral type*, *interface*, *residual specification*.

Participants: Éric Badouel, Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Sophie Pinchinat.

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).

We develop a probabilistic model of contract and pursue the work on residuation of modal specifications in three directions:

- Handling multiple viewpoint design and synthesis as well as shared refinement.
- Extending the framework to time modal specifications.
- Investigating an algebraic formulation of residuation.

5.3.1. Probabilistic Models of Contracts

We have successfully built a probabilistic extension [26] of the Assume/Guarantee theory of contracts initiated in IP-Speeds. The probabilistic viewpoint is introduced by considering some of the uncontrolled ports of a contract (inputs for instance) as subject to probability distributions on their possible histories. As a consequence, the satisfaction relation is a quantification of component's runs that satisfy the behavior guaranteed by the contract. This allows to carry out precise reliability analyses at an early stage in system design. Mild restrictions on probabilistic contracts ensure that the probabilistic satisfaction and dominance operations are compositional, thus ensuring the scalability of reliability analyses in the framework of component-based system design methodologies.

A drawback of this approach is that it eludes the effective computation of the satisfaction and refinement probabilities. At best these are of a high computational complexity if we consider assumptions, guarantees and component implementations as regular languages of runs. We are currently developing a more pragmatic approach where contracts are Markov Decision Processes and implementations Open Transition Systems. In this case, computing satisfaction and refinement probabilities relies on the existence of pure optimal strategies in mean-payoff Markov Decision Processes, which is already addressed in [38]. We are still working on proving compositionnality for this formalism, but a preliminary version of the work has already been presented in MOVEP'08.

5.3.2. Residuation of Modal Specifications

5.3.2.1. Multiple Viewpoints

There are practical reasons for not being totally satisfied with the theory of interfaces as it has evolved. First, it is often desirable to reuse whenever possible a same implementation for two different interfaces. Or, more

generally, to refine two different interfaces by a common, more detailed interface; this is referred to as shared refinement. Second, complex system design includes various aspects — functional, timeliness, resource, safety and fault tolerance, etc. — involving different teams with different skills using heterogeneous techniques and tools. This leads to handling several aspects or viewpoints for a same sub-system or component. When contemplated through the framework of interface theory, the above two use cases require the possibility that a same interface refines (or implements) several interfaces, not just a single one.

In the context of a collaboration between INRIA, EPFL and the University of Trento we have further developed the approach on modal specifications to allow for multiple viewpoints and shared refinement. Modal specifications come equipped with several operations: parallel composition $\otimes$ and refinement order $\preceq$, which in turn induces the greatest lower bound (meet) $\wedge$. We showed that this meet allows addressing the two requirements of multi-viewpoint and shared refinement. To achieve this, we need to enhance modal specifications with inputs and outputs and we express contracts in the resulting model of Modal Interfaces. Assume/Guarantee reasoning as it is supported in the framework of the Speeds project (concept of rich component) is compared with Modal Interfaces.

5.3.2.2. Timed Modal Specifications

The increasing complexity of computer systems has led to methodologies almost systematically based on component assembling. Because in the system development process, some pieces may not be completed or are not yet available, analysis methodologies must rely on an abstract description of the components behaviour.

Logic-based formalisms, such as modal and temporal logics, are robust formal tools to express statements about the behaviours of computer systems. Unfortunately, logics do not relate well in general to compositional approaches; the description of a system as interacting components cannot be exploited. However, by conceding a loss of expressiveness, like confining to safety properties, satisfactory frameworks can be developed.

We introduce *timed modal specifications*, an automata-based formalism combining modal and timed aspects. As a stepping stone to compositional approaches of timed systems, we define the notions of refinement and consistency, and establish their decidability.

This is joint work with Nathalie Bertrand (Vertecs, INRIA, Rennes) and Jean-Bapitste Raclet (Popart, INRIALPES); it is submitted for publication.

Modal specification-based approaches seem promising to develop formal tools in the challenging domain of embedded systems, provided the framework can take real-time aspects into account.

Towards this end, the extension of the algebraic framework of [44] to a timed setting has been investigated. Timed modal specifications (TMS) provide a logical formalism which combines modal and timed statements. They generalize both modal specifications and timed automata, just as timed automata generalize ordinary automata, and modal specifications generalize ordinary automata, respectively. We have studied the refinement and the consistency (the existence of a common model) of TMS. Decision methods for these two problems are achieved by bridging timed modal specifications and (untimed) modal specifications, via a region-based construction. These results are stepping stones to compositional reasoning on timed systems. This work have been recently submitted for publication.

We are now considering the issue of extending product and quotient of [44] to timed modal specifications. We strongly believe that this should be done in a region-based manner, borrowing know-hows from the untimed setting. We also plan to study the embedding of timed interfaces [49] into modal timed specifications, via a construction in the line of [39] for the untimed setting. The main difficulty would be to establish that the compatibility relation of [49] is hereditary with respect to the present notion of refinement preorder; as a consequence, two compatible components could be implemented independently.

5.3.2.3. Residuation of Processes

The semantics of system components and their specifications may often be defined as (partial) functions that map trajectories to properties reached at their endpoints. The properties of interest may be logical (e.g. they may be state predicates or they may be modal predicates that describe sets of continuations) or non-logical (e.g. they may measure the cost of the trajectories). The composition of components, and the conformance of

the components to their specification may then be rendered by process composition, resp. by comparison of processes. In this general framework we are particularly interested in computing residues as follows: given two processes r and q, compute a third process r/q (the residue of r by q) such that r is equivalent to the composition of q and r/q. This problem was addressed in J.B. Raclet's PhD thesis for two types of processes with synchronous composition, namely modal trees and acceptance trees. Our goal is to extend the constructions to arbitrary definitions of processes and process composition.

We see processes as partial maps $\mathcal{D} \to \mathcal{K}$ where $\mathcal{D}$ is a domain of trajectories and $\mathcal{K}$ is a lattice of properties. We assume that $\mathcal{D}$ comes equipped with a non-deterministic composition $\bowtie$: $\mathcal{D} \times \mathcal{D} \to \mathcal{P}(D)$, that combines the trajectories of two components into one trajectory, and that $\mathcal{K}$ is a quantale, whose tensor product $\otimes$ combines the properties of two components into one global property. We recall that a quantale is a complete lattice with arbitrary meets $(K, \bigvee)$ and simultaneously a monoid $(K, \otimes, 1)$, such that arbitrary meets distribute over tensor product. Then, the composition $p \bowtie q$ of two processes may be defined with $p \bowtie q(w) = \bigvee\{p(u) \otimes q(v) \,|\, w \in u \bowtie v\}$ (where u, v, w are trajectories in $\mathcal{D}$). In a quantale $\mathcal{K}$ with commutative tensor $\otimes$, $x \otimes y \leq z$ iff $x \leq (z/y)$ where the residue (z/y) is the property $\bigvee\{x \,|\, x \otimes y \leq z\}$. We have shown that the computation of such residues may be lifted from the quantale of properties $\mathcal{K}$ to the set of processes $\mathcal{D} \to \mathcal{K}$. More precisely, for any processes p, q and r, $p \bowtie q \leq r \; iff \; p \leq (r/q)$ where the residue (r/q) is the process defined with $(r/q)(u) = \bigwedge\{r(w)/q(v) \,|\, w \in u \bowtie v\}$.

It remains to recast J.B. Raclet's specific constructions in this framework, using for $\mathcal{K}$ the binary quantale of modalities $\{may, must\}$ or the quantale of acceptance sets over an alphabet, with expected simplifications due to the more algebraic treatment, and then to explore the application of the ideas to other types of properties, e.g. mixing logical aspects, durational aspects, costs and so on (this generality should come for free since quantales are closed under cartesian product). This work is done as part of the European project Combest.

5.3.3. Heterogeneous Rich Components Models

The Speeds European project (see Section 6.4), started on May 1st, 2006, is a concerted effort to define a new generation of end-to-end methodologies and supporting tools for safety-critical embedded system design. The technology developped in Speeds is based on the concept of heterogeneous rich components [25], allowing for the description of the expected behavior of a component, on a per-viewpoint basis (functional, timing, reliability, resource usage, etc.), thanks to an assume/guarantee style of reasoning [16]. This formalism enables scalable modular analysis methods capable of checking the realizability of a virtual system model at an early stage of design.

5.4. Discrete event system synthesis and supervisory control

Keywords: *concurrent secrets*, *control*, *discrete event system*, *modal logics*, *mu-calculus*, *opacity*, *parity game*, *partial observation*, *regular languages*, *tree automata*, *winning strategy*.
Participants: Benoît Caillaud, Philippe Darondeau, Sophie Pinchinat, Laurie Ricker.

5.4.1. Supervisory Control and Security

Two years ago, we started a new topic of research by considering the definition and computation of finite and optimal control (of discrete event systems) in the perspective of computer security [32]. This year, we focussed on security enforcing control in the presence of uncontrollable actions. The context is as follows. Given a finite DES with language L, a subset S of L called the secret, two subaphabets Σ_a and Σ_o containing the actions that may be observed by the adversary and by the controller, respectively, and a subalphabet Σ_c containing the actions that may be disabled by supervisory control, one wants to compute a maximal permissive controller K such that, for any word $w \in S \cap K$, there exists some corresponding word $w' \in (L \setminus S) \cap K$ with an identical projection on Σ_a^*. Such a controller K is said to enforce the opacity of S. When the problem has a solution, it has always an optimal solution. However, we have shown in [22] that the optimal controller K can generally not be computed by Ramadge and Wonham's theory and algorithms. We proposed three algorithms for computing the optimal controller K in the respective cases where $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_a$, or $\Sigma_a \subseteq \Sigma_c \subseteq \Sigma_o$, or $\Sigma_c \subseteq \Sigma_a \subseteq \Sigma_o$. The first two algorithms may be seen as adaptations of the classical supervisory control theory for safety properties, while the third algorithm is new. These results have been obtained in a cooperation with Jeremy Dubreil and Hervé Marchand (VERTECS team-project).

5.4.2. Quasi Static Scheduling

Good scheduling policies are required in distributed embedded applications for meeting hard real time constraints and for optimizing the use of computational resources. We have dealt in [20] with the yet unsolved problem known as quasi static scheduling. We set the problem for a network of sequential processes that communicate by point-to-point buffers. The network waits for a request from the environment, executes a fixed task, and returns to the waiting state. The difficulty comes from the uncontrollable choices made by each process, representing e.g races between input messages or branchings dependent upon the contents of the messages received. Quasi static scheduling should keep fixed bounds on resource usage (including the use of memory by the scheduler), and it should not harm the successful termination of the cyclic task. We prove that it is undecidable whether some quasi static scheduling can be found for a given system. However, we show that the same problem is decidable if all branchings within a process are internal to this process, i.e. they do not depend on the contents of the communication buffers at the time of the choice. This decidability result, which is not trivial to establish, exploits ideas derived from the Karp and Miller coverability tree as well as the concept of existential boundedness of message sequence charts. This work was done in the framework of the CASDS associated team between the INRIA project-teams S4 and DISTRIBCOM and the National University of Singapore.

5.4.3. Asymptotic Minimal Communication for Decentralized Discrete-Event Control

The foremost motivation for introducing communication into the decentralized discrete-event control problem is to allow controllers to distinguish between behaviors of a system that remain within a pre-defined set of so-called legal behaviors and those that lead to behaviors outside this set. After communication, updates to a controllers's partial view of the system should reflect that such behaviors are now distinguishable whenever either the legal or illegal behavior occurs.

Algorithms for finding minimal communication policies for decentralized discrete-event control use the following notion for optimality: remove any one of the elements from the communication set and either the control problem can no longer be solved correctly or a condition of observational equivalence is violeted. In contrast, a strategy for finding a globally optimal communication policy, based not on structural properties of the system, but rather, in a behavioral sense, is presented in [24], The problem of finding a minimal communication set is reduced to an optimization problem for a set of Markov chains.

In state-based models, updates occur only at information states that occur immediately after a communication, resulting in inconsistent information states in the rest of the model. In a paper submitted for publication, a previously-defined state-based structure [33] is expanded to include the notion of system-wide information state updates in response to expected communications.

5.4.4. Systems with Imperfect Information

Co-observability in discrete-event dynamical systems. The broad-ranging topics of diagnosis, control and games with imperfect information are united by one concept: the notion of *observability*. In these problems, existence of an optimal solution (e.g., a diagnoser, a control policy, a winning strategy) is contingent upon the observability of the goal (e.g., diagnosis pattern [45], control objective [40], winning conditions of the game). When there is a single observer of the system, observability problems can be solved using a variety of domain-dependant approaches (e.g., twin plant algorithm for diagnosis, language theoretic techniques in control). In the event of multiple observers, most observability problems are undecidable [47]. However, it is unclear how to proceed towards the identification of classes where decidability results *can* be obtained. We worked towards a better understanding of why observability problems are undecidable in general.

More precisely, we explored formalizations of observability problems in an infinitary setting in a way that extends the existing finitary settings. In this approach we revisited observability problems in terms of saturation problems which can be solved using formal language techniques (e.g., rational relations, transducers [34], [46], [42]).

This is initially joint work with Laurie Ricker (Mathematics and Computer Science, Mount Allison University) and Nicolas Bitouzé (Msc student, ENS Cachan) and will be submitted for publication.

Intended objectives in games with imperfect information. We consider a framework where decision makers are consulted to allow a move, and their decision is persistent between two consecutive consultations while the game goes on. An intended objective is given as a set of plays.

We give a necessary and sufficient condition for the existence of strategies to attain the objective. When this condition does not hold, we calculate the least attainable objective that extends the intended one. We examine some theoretical and computational aspects of this solution.

This is ongoing work, and has been presented at the annual meeting of the national working group GT Jeux (LaBRI, Bordeaux, 4-5 June 2008).

Diagnosis of infinite-state discrete-event systems. 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 higher-order pushdown systems and investigate two basic variants of diagnosis problems: 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.

We establish that the diagnosability problem is decidable for arbitrary sub-classes of higher-order visibly pushdown systems provided unobservable events leave the stacks unchanged. For this case, we present an effective algorithm. Otherwise, we show that diagnosability becomes undecidable already for first-order visibly pushdown automata. Furthermore, we establish that the bounded-latency problem for higher-order pushdown systems is as hard as deciding finiteness of a higher-order pushdown language. This is in contrast with the case of finite-state systems where the problem reduces to diagnosability.

This is joint work with Christophe Morvan (Institut Gaspard-Monge, Université de Marne-la-Vallée) available as a technical report [27], and submitted for publication.

5.4.5. Logics for Games

We have considered two basic approaches towards formal reasoning about games: the Propositional Logic of Games introduced by Parikh [41] in 1983, which is the first formalism to incorporate games into a logic of computation, and the framework of Alternating-Time Temporal Logics of Alur, Henzinger, and Kupferman [31] introduced 15 years later, which is arguably the most influential game-based formalism in Computer-Science applications by today.

We analyse 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.

This is joint work with Dietmar Berwanger (LSV, ENS Cachan) and it is submitted for publication.

6. Other Grants and Activities

6.1. Synchronics: Language Platform for Embedded System Design

Participants: Benoît Caillaud, Mateus Oliveira.

Synchronics is an INRIA Large 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/.

The Synchronics Large Initiative Action, funded by INRIA, 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.

In 2008, Benoît Caillaud and Mateus Oliveira have investigated techniques allowing efficient but sound coupling of simulation codes, covering both continuous and discrete evolutions. For the continuous parts, waveform relaxation techniques [48] have proved to be effective means of connecting simulation codes with dissimilar sampling rates. For particular classes of of dynamic systems defined by ordinary differential equations satisfying the Peano-Picard-Lindelof conditions, fast convergence has been proved, therefore ensuring fast simulation. In the case of synchronous programs, simulation amounts to compute fix-points of systems of boolean equations. Synchronous language compilers implement such fix-point computations. However, these algorithms can not be used for simulation, as fix-points have to be computed online, during the simulation. In general, computing all fix-points requires back-tracking. This makes coupling of simulation codes impractical because of poor performance. However, there are classes of synchronous designs for which there exists a least fix-point, and it can be computed online, without back-tracking, with very simple self-stabilizing distributed algorithms. Delay insensitive programs fall into this category [35]. More generally, designs that can be safely mapped to globally asynchronous locally synchronous designs also allow efficient simulation [43].

6.2. 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.

6.3. Artist2 and ArtistDesign – Networks of Excellence on the Design of Advanced Real-Time Systems

Participants: Albert Benveniste, Benoît Caillaud.

IST-004527 Artist2 (September 2004, September 2008) and IST-214373 ArtistDesign (January 2008, December 2011), http://www.artist-embedded.org/artist/

In 2008, our contribution to the Artist2 and ArtistDesign networks of excellence has consisted in collaborations with Paul Caspi and Alberto Sangiovanni-Vincentelli, on the development of the Loosely Time-Triggered Architecture (LTTA) theory [11], [18] — See section 5.2.1 for more details.

6.4. Speeds: Speculative and Exploratory Design in Systems Engineering

Participants: Éric Badouel, Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Mateus Oliveira.

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), 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.

A component in the HRC is described as a set of assume/guarantee contracts [16] which explicates assumptions about its environment and state corresponding guarantees on the service offered by the component to its environment. Contracts fall in several categories, in which both functional and non-functional (timing, reliability, resource consumption, etc.) properties of the component's behavior can be expressed [25].

The HRC formalism is built on top of existing standard (UML and SysML of the OMG) and will be implemented as an Eclipse plugin, using state-of-the-art meta-modeling technology [29]. The HRC Eclipse plugin will have gateways with existing IDE tools, including SCADE (Esterel Tech.), Rapsody (Telelogic), RT-Builder (TNI) and MatLab/Simulink.

In 2008, the development of a Proof Obligation Generator (POG for short) has been initiated. The POG handles contracts contained in HRC models and reduces the verification contract refinement and satisfaction relations to classical reachability verification problems, to be solved by a model-checker. Feasibility experiments have been carried out with the NuSMV model-checker. This software development will continue in 2009. The integration of the POG in the Speeds Bus Integration Platform is expected by June 2009, for deployment and use by Speeds industrial users in pilot projects.

6.5. Combest: Component-Based Embedded Systems Design Techniques

Participants: Éric Badouel, Albert Benveniste, Benoît Caillaud, Philippe Darondeau, Benoît Delahaye, Sophie Pinchinat, Maurice Tchoupé.

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 6.4, both regarding heterogeneous rich components and compositional analysis methods.

In 2008, 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 T. Henzinger at EPFL, Lausanne, Switzerland (see section 5.3), probabilistic models of contracts (see section 5.3.1) and the residuation of modal specifications (see section 5.3.2).

6.6. 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.

6.7. 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.

7. Dissemination

7.1. Participation to editorial boards and program committees

Éric 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'08 (Rabat, November 2008). 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 is serving on the steering committee of the International Conference on Application of Concurrency to System Design (ACSD). He has also served on the program committee of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008).

Philippe Darondeau Philippe Darondeau has served in the programme committees for the Workshop on Discrete Event Systems (Wodes 2008, Götheborg) and for the China workshop (affiliated to ICATPN 2008, Xi'an). Philippe Darondeau is the secretary of the IFIP WG2.2 working group.

Sophie Pinchinat has served in the program committees of SLA++P 08 (Model-driven High-level Programming of Embedded Systems) and of WODES 2008 (Workshop on Discrete Event Systems). She takes part of the national working Groups : GT Jeux of GDR IM Complexity and Logics. She is serving as public relations associate and science policy advisor on the Advisory Board of the Marie Curie Fellows Association .

7.2. 68NQRT: Theory of computing seminar of Irisa

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.

7.3. 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.
- 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: Elements of Game Theory (4h), Automata, Logic, and Games(10h), Automata and languages (30h), Computability and Complexity (30h), Advanced Algorithmics (20h), Formal Verification (20h).

8. 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*, Lecture Notes in Computer Science, vol. 1491, Springer, 1999, p. 529–586.

[5] 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.

[6] A. BENVENISTE, P. CASPI, S. EDWARDS, N. HALBWACHS, P. LE GUERNIC, R. DE SIMONE. *The Synchronous Languages Twelve Years Later*, in "Proceedings of the IEEE", Special issue on modeling and design of embedded software, vol. 91, n^{o} 1, 2003, p. 64–83, http://www.irisa.fr/s4/download/papers/Benveniste-proc-ieee-2003.pdf.

[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

Articles in International Peer-Reviewed Journal

[11] 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.

Articles in National Peer-Reviewed Journal

[12] 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/.

Invited Conferences

[13] P. DARONDEAU. *On the Synthesis of Zero-Safe Nets*, in "Concurrency, Graphs and Models. Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday", Lecture Notes in Computer Science, vol. 5065, Springer-Verlag, 2008, http://dx.doi.org/10.1007/978-3-540-68679-8_25.

International Peer-Reviewed Conference/Proceedings

[14] E. BADOUEL, B. FOTSING, R. TCHOUGONG. *Attribute grammars as recursion schemes over cyclic representations of zippers*, in "Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science (CMCS 2008)", Electronic Notes in Theoretical Computer Science, Elsevier, 2008, p. 37–54.

[15] 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.

[16] 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, vol. 5382, Springer, October 2008.

[17] E. BEST, P. DARONDEAU. *Decomposition Theorems for Bounded Persistent Petri Nets*, in "Applications and Theory of Petri Nets", Lecture Notes in Computer Science, vol. 5062, Springer-Verlag, 2008, p. 33–51, http://dx.doi.org/10.1007/978-3-540-68746-7_7.

[18] P. CASPI, A. BENVENISTE. *Time-Robust discrete control over networked Loosely Time-Triggered Architectures*, in "IEEE Control and Decision Conference", dec 2008, https://css.paperplaza.net/conferences/scripts/abstract.pl?ConfID=32&Number=87.

[19] P. DARONDEAU, B. GENEST, L. HÉLOUËT. *Products of Message Sequence Charts*, in "Foundations of Software Science and Computational Structures", Lecture Notes in Computer Science, vol. 4962, Springer-Verlag, 2008, p. 458–473.

[20] P. DARONDEAU, B. GENEST, P. S. THIAGARAJAN, S. YANG. *Quasi-Static Scheduling of Communicating Tasks*, in "CONCUR 2008 - Concurrency Theory, Berlin/Heidelberg", Lecture Notes in Computer Science, vol. 5201, Springer-Verlag, 2008, p. 310–324, http://dx.doi.org/10.1007/3-540-48068-4_7.

[21] P. DARONDEAU, M. KOUTNY, M. PIETKIEWICZ-KOUTNY, A. YAKOVLEV. *Synthesis of Nets with Step Firing Policies*, in "Applications and Theory of Petri Nets", Lecture Notes in Computer Science, vol. 5062, Springer-Verlag, 2008, p. 112–131, http://dx.doi.org/10.1007/978-3-540-68746-7_11.

[22] J. DUBREIL, P. DARONDEAU, H. MARCHAND. *Opacity Enforcing Control Synthesis*, in "Proceedings of the 9th International Workshop on Discrete Event Systems (WODES'08), Göteborg, Sweden", B. LENNARTSON, M. FABIAN, K. AKESSON, A. GIUA, R. KUMAR (editors), IEEE, May 2008, p. 28–35.

[23] B. FOTSING. *Interactive editing of tree-structured data*, in "Proceedings of the 9th African Conference on Research in Computer Science and Applied Mathematics (CARI'08)", E. BADOUEL, A. SBIHI, M. K. ASSOGBA (editors), October 2008, p. 711–718.

[24] L. RICKER. *Asymptotic Minimal Communication for Decentralized Discrete-Event Control*, in "Proceedings of the 9th International Workshop on Discrete Event Systems (WODES'08)", B. LENNARTSON, M. FABIAN, K. AKESSON, A. GIUA, R. KUMAR (editors), May 2008, p. 486–491.

Scientific Books (or Scientific Book chapters)

[25] A. BENVENISTE, B. CAILLAUD, R. PASSERONE. *Multi-Viewpoint State Machines for Rich Component Models*, in "Model-Based Design of Heterogeneous Embedded Systems", P. MOSTERMAN, G. NICOLESCU (editors), CRC Press, 2008.

Research Reports

[26] B. DELAHAYE, B. CAILLAUD. *A model for Probabilistic Reasoning on Assume/Guarantee Contracts*, Research Report, n^{o} 6719, INRIA, 2008, http://hal.inria.fr/inria-00337538.

[27] C. MORVAN, S. PINCHINAT. *Diagnosis of Pushdown Systems*, PI, n^{o} 1904, Irisa, nov 2008, http://hal.inria.fr/inria-00337614/en/.

References in notes

[28] *ITU-TS Recommendation Z.120: Message Sequence Chart (MSC)*, International Telecommunication Union, Geneva, 1993, http://www.itu.int/home/index.html.

[29] *D.2.1.b SPEEDS Meta-model Syntax and Static Semantics*, 2007, SPEEDS project deliverable.

[30] *OMG Unified Modeling Language, version 2.0*, 2003, http://www.omg.org/uml/, Draft specification.

[31] R. ALUR, T. A. HENZINGER, O. KUPFERMAN. *Alternating-time temporal logic*, in "Journal of the ACM", vol. 49, 2002, AlurHenKup02.

[32] 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.

[33] G. BARRETT, S. LAFORTUNE. *Decentralized Spervisory Control with Communicating Controllers*, in "IEEE Transactions on Automatic Control", vol. 45, n^{o} 9, Sept. 2000, p. 1620–1638.

[34] J. BERSTEL. *Transductions and Context-Free Languages*, Teubner Studienbücher, Stuttgart, 1979.

[35] L. P. CARLONI, K. L. MCMILLAN, A. L. SANGIOVANNI-VINCENTELLI. *Theory of latency-insensitive design*, in "IEEE Transactions on CAD", vol. 20, n^{o} 9, September 2001.

[36] J. CORTADELLA, M. KISHINEVSKY. *Synchronous Elastic Circuits with Early Evaluation and Token Counterflow*, in "DAC", 2007, p. 416-419.

[37] J. CORTADELLA, M. KISHINEVSKY, B. GRUNDMANN. *Synthesis of synchronous elastic architectures*, in "DAC", 2006, p. 657-662.

[38] H. GIMBERT. *Pure Stationary Optimal Strategies in Markov Decision Processes*, in "STACS", W. THOMAS, P. WEIL (editors), Lecture Notes in Computer Science, vol. 4393, Springer, 2007, p. 200–211, http://dx.doi.org/10.1007/978-3-540-70918-3_18.

[39] KIM G. LARSEN, U. NYMAN, A. WASOWSKI. *Modal I/O Automata for Interfaces and Product Line Theories*, in "Proceedings of ESOP 2007", Lecture Notes in Computer Science, vol. 4421, Springer Verlag, 2007, p. 64-79.

[40] F. LIN, W. M. WONHAM. *On observability of discrete-event systems*, in "Information Sciences", vol. 44, n^{o} 3, 1988, p. 173–198.

[41] R. PARIKH. *Propositional Game Logic*, in "IEEE Symposium on Foundations of Computer Science", IEEE (editor), 1983, p. 195–200.

[42] D. PERRIN, J.-E. PIN. *Infinite words, automata, semigroups, logic and games*, Elsevier, 2004.

[43] P. 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.

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

[45] M. SAMPATH, R. SENGUPTA, S. LAFORTUNE, K. SINAAMOHIDEEN, D. TENEKETZIS. *Failure Diagnosis Using Discrete Event Models*, in "IEEE Transactions on Control Systems Technology", vol. 4, n^{o} 2, March 1996, p. 105-124.

[46] L. STAIGER. *ω-Languages*, in "Handbook of formal languages, vol. 3: beyond words, New York, NY, USA", G. ROZENBERG, A. SALOMAA (editors), chap. 10, Springer-Verlag New York, Inc., 1997, p. 339–388.

[47] S. TRIPAKIS. *Undecidable problems of decentralized observation and control on regular languages*, in "Inf. Process. Lett.", vol. 90, n^{o} 1, 2004, p. 21-28.

[48] J. K. WHITE, A. L. SANGIOVANNI-VINCENTELLI. *Relaxation Techniques for the Simulation of VLSI Circuit*, Kluwer, 1987.

[49] L. DE ALFARO, T. A. HENZINGER, M. STOELINGA. *Timed Inteface*, in "Second Workshop on Emedded System (EMSOFT)", Lecture Notes in Computer Science, vol. 2491, Springer Verlag, 2002, p. 108-122.