INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Project-Team s4*

# *System Synthesis and Supervision, Scenarios*

*Rennes*

THEME COM

# Table of contents

# 1. Team

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

**Head of project-team**
Benoît Caillaud [ Research Associate (CR) ]

**Administrative assistant**
Laurence Dinh [ TR, part-time in S4 ]

**Research scientists**
Éric Badouel [ Research Associate (CR), HdR ]
Albert Benveniste [ Research Director (DR), part-time in S4, HdR ]
Philippe Darondeau [ Research Director (DR), HdR ]

**Faculty member (University of Rennes 1)**
Sophie Pinchinat [ Lecturer Rennes I, HdR ]

**Postdoctoral research scientist**
Guillaume Feuillade [ Teaching assistant at University of Rennes 1 until August 2006 ]

**Ph. D. students**
Benoît Delahaye [ ENS CACHAN, ANTENNE DE BRETAGNE, since September 2006 ]
Rodrigue Djeumen [ Funded by SCAC Yaoundé (Service de Coopération et d'Action Culturelle de l'Ambassade de France), supplemented by SARIMA, part time in France ]
Bernard Fotsing [ Funded by AUF (Agence universitaire de la Francophonie), part time in France ]
Jean-Baptiste Raclet [ Funded by the collaboration CO2 with France Telecom ]
Maurice Tchoupé [ Funded by SCAC Yaoundé (Service de Coopération et d'Action Culturelle de l'Ambassade de France), supplemented by SARIMA, part time in France ]

# 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* [20] (sequence diagrams and statecharts), the *High-Level Message Sequence Charts* [19] 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 platforms, 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.

**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.

Team S4 contributes methods, 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 five tracks:

**Petri net theory:** This track follows up the main research theme of the former Team PARAGRAPHE at INRIA Rennes. In addition to further developments on topics related to the theory of regions, an algebraic theory of nets has been developed.

**Scenario languages:** Current research work concentrates on the composition of system views expressed in scenario formalisms such as *High-Level Message Sequence Charts* (HMSC) [19]. This research is done in cooperation with France Telecom (section 6.1).

**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 a component-based language development environment, where attribute grammars are used to define executable specifications of software components.

**Discrete event system synthesis and supervisory control:** Many synthesis and supervisory control problems can be expressed in the *quantified mu-calculus* with full generality, 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* [20] or *Message Sequence Charts* [19]).

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* [19] 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 to contributing 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 are reducible 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.

Research on scenario languages, and in particular on compositions of *High-Level Message Sequence Charts*, is well suited to the specification and analysis of *services* in *intelligent* telecommunication networks. This work is funded by France Telecom (section 6.1).

Our work on heterogeneous reactive systems facilitates the mapping of pure synchronous designs to 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 recent Airbus planes. 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 7.1) 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

**Keywords:** *MV-algebra*, *Petri algebra*, *Petri net*, *modular synthesis of Petri nets*.

**Participants:** Éric Badouel, Philippe Darondeau.

A serious limitation of all known procedures for synthesizing Place/Transition-nets based on regions of graphs or languages is to require computing the full state space of the input transition system. An attempt towards *modular synthesis* has been made with Luca Bernardinello (University of Milano) and Laure Petrucci (University of Villetaneuse, Paris) in order to avoid the state space explosion. Laure Petrucci's modular automata stay in between indexed families of automata $(A_i)_{i=1}^n$ and their mixed products $\otimes_i A_i$. In a modular automaton $((A'_i)_{i=1}^n, \mathcal{S})$, the modules $A'_i$ are the residues left after all synchronized transitions have been exported to the synchronization graph $\mathcal{S}$. A modular synthesis procedure has been designed for *distributed P/T-nets*. It remains to analyze the performances of this procedure as a function of $n$ and the respective size of each component automaton.

The firing rule of Petri nets relies on a residuation operation for the commutative monoid of natural numbers. We have identified in [10] a class of residuated commutative monoids, called Petri algebras, for which one can mimic the token game of Petri nets to define the behavior of generalized Petri nets whose flow relation and place contents are valued in such algebraic structures. The sum and its associated residuation respectively capture how resources within places are produced and consumed through the firing of a transition. We have shown that Petri algebras coincide with the positive cones of lattice-ordered commutative groups and constitute the sub-variety of the (duals of) residuated lattices generated by the commutative monoid of natural numbers. We however have exhibited a Petri algebra whose corresponding class of nets is strictly more expressive than the class of Petri nets. More precisely, we have introduced a class of nets, termed lexicographic Petri nets, that are associated with the positive cones of the lexicographic powers of the additive group of real numbers. This class of nets has been proved universal in the sense that any net associated with some Petri algebra can be simulated by a lexicographic Petri net. All the classical decidable properties of Petri nets however (termination, covering, boundedness, structural boundedness, accessibility, deadlock, liveness, *etc*.) are undecidable on the class of lexicographic Petri nets. Finally we have turned our attention to bounded nets associated with Petri algebras, and have shown that their dynamic can be reformulated in term of MV-algebras.

## 5.2. Reactive components

**Keywords:** *attribute grammar*, *behavioral type*, *coherence of views*, *interface*.

**Participants:** Éric Badouel, Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Rodrigue Djeumen, Bernard Fotsing, Maurice Tchoupé, Jean-Baptiste Raclet.

Component-based design is acknowledged as an important approach to improving the productivity in the design of complex software systems, as it allows pre-designed components to be reused in larger systems. However, designing components for reuse calls for a system of program annotations rich enough to ensure that the components will interact in a coherent manner when connected together. The dynamic information about the interactions of the component with its environment combines expectations of the component about its environment with guarantees offered in return by the component to its environment.

When it comes to reactive systems, a natural extension of the usual functional type systems is to consider finite automata with input and output channels as alphabets. These automata can take care of the more involved protocols of communication that take place among components in interaction. For that purpose, L. De Alfaro and Th. Henzinger introduced *Interface Automata*, viewed as enriched type systems (the so-called *Behavioral Type Systems*), which capture the temporal aspects of software component interaction. A component refines another component if it imposes less constraint about the environment and offers more guarantee in return. We obtain in this way a compositional semantics due to the fact that a component can be replaced with a more refined version in any environment compatible with the original component. The refined version may offer more services but both are equivalent in restriction to the set of services of the original component. This situation is reminiscent to the sub-class polymorphism in object-oriented programming.

During Year 2006, the activity in this track of the S4 team can be summarized as follows:

- In 2005, we addressed the problem of assembling partial views of the behavior of a distributed system, and introduced for this purpose an operation of controlled shuffle on languages of (compositional) message sequence charts (MSC). The operation of controlled shuffle of two MSCs amounts, for each process, to interleave the send/receive events of these two components, and to amalgamate their synchronization events, without ever introducing circularity in the resulting flow. In this context, a subclass of MSC languages, called the existentially bounded MSCs, seems to be especially interesting, since no implementation can reasonably be envisaged for MSC languages that do not belong to this class. Answers have been provided for the main decision problems left open last year concerning controlled shuffle and existential boundedness. Whether the controlled shuffle of two existentially bounded languages of MSCs is existentially bounded is undecidable. However, this problem is decidable when the two component MSCs only synchronize on a single process. These results, presented in [18], have been obtained in cooperation with Blaise Genest and Loïc Hélouët

(DISTRIBCOM project) within action CO2 (collaboration between INRIA and France Telecom Research and Development).

- We are also investigating a component-based language development environment, based on the formalism of attribute grammars. There an abstract syntax tree decorated with attributes that conforms to an attribute grammar is seen as an executable specification of a software component. Language-oriented programming is an approach to software composition based on domain specific languages (DSL) dedicated to specific aspects of an application domain. In order to combine several such languages, we embed them into a host language (Haskell, a strongly-typed, higher-order, lazy functional language). We have shown how an attribute grammar provides a specification of an embedded DSL, and thus appears as a formalism adapted to the prototyping of such languages. We have adopted an incremental approach consisting in growing such a DSL by cascaded composition of modular attribute grammars. An abstract attribute grammar consists of an algebraic type, whose operators are the constructors of abstract syntax trees, together with a set of semantic rules defining the value of attributes attached to each syntactic category. The evaluation of the attributes can be computed as the catamorphism associated with an algebra derivable from the semantic rules of the attribute grammar. By generalizing on Bekìc theorem we provide a modular decomposition of such catamorphisms from which we derive a modular class of (extended) attribute grammars associated with regular functors. We have shown how to grow a language using cascaded composition of modular attribute grammars. A dual result for the decomposition of anamorphisms has possible applications to the modular specification of editors and syntax-directed translators. A complex structured document is intentionally represented as a tree decorated with attributes. The set of legal structures are given by an abstract context-free grammar. That intentional representation may be asynchronously manipulated by a set of independent tools, each of which operates on a distinct partial view of the whole structure. In order to synchronize these various partial views, we face the problem of their coherence: can we decide whether there exists some global structure corresponding to a given set of partial views and in the affirmative, can we produce such a global structure? We solved this problem in the case where a view is given by a subset of grammatical symbols, those associated with the so-called visible syntactical categories [14]. The proposed algorithm, that strongly relies on the mechanism of lazy evaluation, produces an answer to this problem even though partial views may correspond to an infinite set of related global structures assuming however that the join data of the different views reduces the solution set to a finite set of global structures. We are investigating an extension of this algorithm in the context of attribute grammars.
- In current component platforms, reuse of a software component is characterized at the signature level. This ensures that two components can interact but not that they will not deadlock. We aim at characterizing reuse of a software at the behavioral level. We have proposed a quotient operation for component specifications. The rationale is the following: given a component $C_1$ model of a specification $S_1$ and a specification $S$ for the desired behavior of the overall system, $C_2$ is a model of the specification $S/S_1$ if and only if the composition of $C_1$ and $C_2$ is a system satisfying $S$. The part to be reused is regarded as a black box: its implementation C1 is unknown, we only make use of the specification S1. To describe the component specifications. We have studied two kind of graphical formalisms: the modal automata and the acceptance automata. Application of this quotient operation concerns the adaptor synthesis problem where a supervisor is synthesized to prevent the composition of two components from deadlocking.
- The SPEEDS European project (http://www.speeds.eu.com/), 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, allowing for the description of the expected behaviour of a component, on a per-viewpoint basis (functional, timing, reliability, resource usage, *etc.*), thanks to an assume/guarantee style of reasoning. This formalism enables scalable modular analysis methods capable of checking the realizability of a virtual system model at a early stage of design. Since the beginning of the project, we have contributed to the mathematical semantics of the Heterogeneous

Rich Component formalism [21].

## 5.3. Discrete event system synthesis and supervisory control

**Keywords:** *control*, *discrete event system*, *modal logics*, *model synthesis*, *mu-calculus*, *parity game*, *partial observation*, *regular languages*, *tree automata*, *winning strategy*.

**Participants:** Éric Badouel, Benoît Caillaud, Philippe Darondeau, Sophie Pinchinat.

Synthesis is a challenging problem: in general, *program synthesis* concerns investigations on automated methods to derive a program from a specification. When the synthesis is done according to semantically correct transformation rules, the resulting programs, if any, is a correct implementation of the specification. Therefore, synthesis techniques are generally preferred to *verification-based approaches*, as the latter can be qualified as "post-mortem examinations", where incorrect programs have to be modified.

The general trend is to restrict synthesis problems to simplified situations, where only some aspects of a program are considered, rather than the whole program itself. We have investigated synthesis methods both on finite transition systems, unlabeled Petri nets, which naturally capture the concurrency of distributed systems, and modal logic specifications. Non-functional properties such as timing constraints are considered.

It is worthwhile noticing that synthesis problems encompass classic theoretical issues such as satisfiability of logical assertions, or control of discrete event systems.

As the computation of winning strategies in two-players games amount to synthesize controller for games, the study of the quantified mu-calculus, and its extensions, deserves being pursued in order to e.g. establish decidable classes of particular classes of strategies.

During year 2006, the activity in this track of the S4 team can be summarized as follows:

- We have 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. Given a discrete event system with regular behaviour $L \subseteq \Sigma^*$ and a subset of observable actions $\Sigma_o \subseteq \Sigma$, a *secret set* $S \subseteq L$ is defined as a regular subset of trajectories of the system. The secret is *opaque* if observing actions from $\Sigma_o$ does not allow to decide whether the trajectory of the system is in $S$. A *concurrent secret* is specified similarly with a n-tuple $\mathcal{S} = \{(\Sigma_1, S_1), ..., (\Sigma_n, S_n)\}$ where each $S_i$ specifies a secret set to be protected against an observer with the set of observable actions $\Sigma_i$. The concurrent secret is *concurrently opaque* if all sets $S_i$ are opaque. Checking concurrent opacity is rather easy. What is more problematic is to compute the largest subset $K$ of $L$ such that $\mathcal{S}$ is concurrently opaque in restriction to $K$. We give sufficient conditions under which this largest restriction $K$, which always exists, is regular and can be computed. These results, obtained in cooperation with Marek Bednarczyk and Andrzej Borzyszkowski (within the joint research action CATALYSIS) have been presented at WODES 2006 [13]
- We have explored a logical framework for the control theory of reactive systems modeled by discrete event systems. The considered logic, called the *Conjunctive Nu-calculus*, is an expressive fragment of the mu-calculus; it possesses an alternative presentation based on modal specifications generalizing finite state automata, while still having simple graphical representations. Modal specifications turn out to be very natural to express and solve centralized control problems of fully observable discrete-event systems. Moreover, they strictly increase the class of control objectives expressible with regular languages, and even by intervals of regular languages, while preserving the validity of the Maximal Model theorem.

  To our knowledge, no investigation has ever been pursued on how far one can go while guaranteeing maximally permissive solutions; this is decisive in the control theory of discrete event systems, and it demonstrates the relevance of the logical fragment in the control theory of reactive systems.

This contribution as been recently accepted for publication in the Journal of Discrete Event Dynamic Systems, and is to appear early 2007.

- Diagnosis problems concern the construction of a device, called a *diagnoser*, which, during the execution of a given partially observable system, can detect/identify the occurrence of particular (sequences of) events, called *patterns*. The diagnoser can thereby be used online to *e.g.* activate an alarm when an undesirable fault has occurred. Importantly, an occurrence of the pattern might not be detected immediately after this occurrence, as some additional observations of the system might be necessary to collect the information required to ascertains its occurrence. The *diagnosability* problem takes this degree of freedom to consideration: given a model of the system and a pattern, it addresses the question of whether an occurrence of the pattern can be detected in some fixed bounded delay. Diagnosability can often be checked effectively .

  Existing works exhibit fairly ad hoc methods to check diagnosability and to synthesize diagnosers – depending on the shape of the pattern. We have proposed in [16] an abstract notion of *supervision pattern* to separate the properties to be diagnosed from the model of the system (as opposed to most existing approaches), and which provides a natural formalism to specify diagnosis problems. In this framework, techniques based on standard operations on transition systems are easily derived both for the construction of a diagnoser and for the verification of diagnosability. These techniques are proved general enough to express and solve a broad class of problems, *e.g.*, diagnosing permanent faults, multiple faults, fault sequences and some problems of intermittent faults.

- Decentralized control of discrete-event systems is a natural approach to decrease the computational complexity of synthesizing controllers for large-scale systems: the overall task of the synthesis is divided into synthesizing individual controllers, each of them reacting according to a partial observation of the system's moves. Most of the works provide only existential results, by exhibiting necessary and sufficient conditions under which a given specification language can be exactly achieved. There are very few results on the synthesis of decentralized controllers for specifications that do not fulfill these conditions.

  Alternatively, modular approaches consider systems that have a structure, and attempt to decompose the specification according to this structure. Under some hypothesis, an optimal solution can be computed efficiently.

  We have proposed in [17] a way to transfer decentralized control problems into modular ones, in order to benefit from the know-how of modular control theory: any decentralized control problem is associated to a natural modular control problem, which over-approximates it. We have studied under which hypothesis a solution of the latter problem delivers a solution of the former one.

# 6. Contracts and Grants with Industry

## 6.1. CO2: Composition of viewpoints based on scenario languages

**Participants:** Éric Badouel, Benoît Caillaud, Philippe Darondeau.

This collaboration with France Telecom Research and Development, in Lannion and Issy-les-Moulineaux is a followup of a previous collaboration with them, which has allowed to develop techniques and tools for the analysis and composition of timed *High-Level Message Sequence Charts* (HMSC) [19].

In 2005, we have addressed the problem of assembling partial views of the behavior of a distributed system. For this purpose, we have defined an operation of controlled shuffle on languages of message sequence charts. Controlled shuffle appears to be suitable for a concise description of protocols. In 2006 we have established that it is undecidable whether the controlled shuffle of two existentially bounded languages of MSCs is existentially bounded. However, this problem is decidable when the two component only synchronize on a single process. These results, presented in [22], have been obtained in cooperation with Blaise Genest and Loïc Hélouët (DISTRIBCOM project)

# 7. Other Grants and Activities

## 7.1. ARTIST I/II – Network of Excellence on Advanced Real-Time Systems

**Participants:** Albert Benveniste, Benoît Caillaud.

*IST-2001-34820 (April 2002, September 2005), http://www.artist-embedded.org/ and IST-004527 ARTIST2 (September 2004, September 2008), http://www.artist-embedded.org/FP6/*

The strategic objective of the ARTIST2 Network of Excellence is to strengthen European research in Embedded Systems Design, and promote the emergence of this new multi-disciplinary area. Until 2005, Albert Benveniste was leading the Hard Real-Time cluster. This year, this cluster has merged with Cluster Modeling and components, and the resulting cluster, called Real-Time Components, is still led by Albert Benveniste. This cluster has four activities, namely

1. Platform for component modelling and verification
2. Development of UML for Real-Time Embedded Systems
3. Forums with specific industrial sectors
4. Seeding new research directions

In the framework of the third activity, Albert Benveniste and Werner Damm (Univ. of Oldenburg and OFFIS, Germany) have co-organized a workshop Beyond AUTOSAR, held on March 23-24, 2006, in Innsbruck, see http://www.artist-embedded.org/artist/-ARTIST2-Workshop-Beyond-AutoSar.html. The workshop was co-located with the Modellierung 2006 conference http://conf.ifit.uni-klu.ac.at/mod2006/, an event organized by German car makers. AUTOSAR is the recent standard for the distributed architecture and runtime in automobile industries.

Also, the responsibility of INRIA as a partner in ARTIST2 has shifted, from Albert Benveniste (until may 2006), to Alain Girault from Team POPART at INRIA Rhône Alpes (starting june 2006).

## 7.2. SPEEDS: Speculative and Exploratory Design in Systems Engineering

**Participants:** Éric Badouel, Albert Benveniste, Benoît Caillaud, Benoît Delahaye, Jean-Baptiste Raclet.

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 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 behaviour can be expressed [21].

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 [23]. The HRC Eclipse plugin will have gateways with existing IDE tools, including SCADE (Esterel Tech.), Rapsody (Telelogic), RT-Builder (TNI) and MatLab/Simulink.

## 7.3. Odas: categorical and algebraic approaches to synthesis

**Participants:** Éric Badouel, Philippe Darondeau, Jean-Baptiste Raclet.

ODAS stands for *Open Dynamic Agent Systems*. It is a follow up of CATALYSIS (*categorical and algebraic approaches to synthesis*), a collaboration initiated in 1999 between Team S4 and the Institute for Computer Science (IPI PAN) in Gdansk, Poland. This collaboration is part of the scientific cooperation framework between CNRS and the Polish Academy of Science. Participant to that collaboration are Éric Badouel, Philippe Darondeau, and Jean-Baptiste Raclet for Team S4, and Marek Bednarczyk, Andrzej Borzyszkowski and Wieslaw Pawlowski for IPI PAN. Visits in Gdansk of members of the S4 project and visits in Rennes of members of IPI PAN (in May) are planned yearly.

# 8. Dissemination

## 8.1. Participation to editorial boards and program committees

Éric Badouel has served in the program committee of the CARI 2006 conference and is the secretary of the steering committee of this conference. He is a member of the editorial board of the ARIMA Journal. He has also participated in the program committee of the 3rd International Conference on Theoretical Aspect of Computing (ICTAC'06).

Albert Benveniste is an associated editor for the journal *IEEE Trans. on Automatic Control*, and a member of the editorial board of the journal and *Proceedings of the* IEEE. He has been in 2006 member of the Program Committee of the following conferences: WODES, EMSOFT. He has been plenary speaker at WODES'2006. He is a member of the Strategic Advisory Council of the Institute for Systems Research, Univ. of Maryland, College Park, USA. He is in charge of managing the *Inria* side of the Alcatel external Research Programme (ARP).

Benoît Caillaud has served in the program and steering committees of the ACSD 2006 conference.

Philippe Darondeau has participated to the program committees of the Workshop on Discrete Event Systems (WODES 2006), the International Conference on the Applications and Theory of Petri Nets (ICATPN 2006), and the International Conference on Formal Engineering Methods (ICFEM 2006). He is the secretary of the IFIP WG2.2 working group.

## 8.2. Demo development

We have pursued, in collaboration with H. Marchand from the INRIA project VerTeCs at IRISA, the development of the open platform, named CTRL-S, dedicated to (1) the simulation of synchronous products of finite state machines, and (2) the integration of toolboxes that compute their controllers. This development has started in 2005 as a demo for the 30th Birthday of IRISA. Programming tasks have been assigned to Zahi Karam, an MSc. student from "École Supérieure d'ingénieurs de Beyrouth" (Liban), and was supported by an INRIA INTERSHIP.

Currently, the platform CTRL-S hosts three toolboxes – UMDES and DESUMA (http://www.eecs.umich.edu/umdes/toolboxes.htm Supremica (http://www.supremica.org/), and Syntool (developped by VerTeCs) –, and is under testing.

## 8.3. 68NQRT: Theory of computing seminar of Irisa

Sophie Pinchinat takes part to the organization of the 68NQRT seminar session of Irisa, dedicated to the following research areas: software engineering, theoretical computer science, discrete mathematics, artificial intelligence.

## 8.4. Teaching

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

- Master of Computer Science, University of Rennes 1, second year: Benoît Caillaud and Sophie Pinchinat are teaching a course on *formal methods for the verification of reactive systems*.
- Eric Badouel is teaching an advanced course on functional programming in the Second year of the Master of Research in Computer Science, University of Yaoundé 1, Cameroon.

# 9. Bibliography

## Major publications by the team in recent years

[1] 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.

[2] 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.

[3] E. BADOUEL, P. DARONDEAU. *Theory of regions*, Lecture Notes in Computer Science, vol. 1491, Springer, 1999, p. 529–586.

[4] 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.

[5] A. BENVENISTE, L. CARLONI, P. CASPI, A. SANGIOVANNI-VINCENTELLI. *Heterogeneous reactive systems modeling and correct-by-construction deployment*, in "Embedded software, third international conference, EMSOFT 2003", R. ALUR, I. LEE (editors). , Lecture notes in computer science, vol. 2855, Springer, oct 2003, p. 35–50, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2855&spage=35.

[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] H. MARCHAND, S. PINCHINAT. *Supervisory Control Problem using Symbolic Bisimulation Techniques*, in "2000 American Control Conference, Chicago, Illinois, USA", jun 2000, p. 4067–4071.

[9] 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 refereed journals and book chapters

[10] E. BADOUEL, J. CHENOU, G. GUILLOU. *An axiomatization of the token game based on Petri algebras*, in "Fundamenta Informaticae", vol. 76, 2006, p. 1-30.

[11] D. POTOP-BUTUCARU, B. CAILLAUD, A. BENVENISTE. *Concurrency in Synchronous Systems*, in "Formal Methods in System Design", vol. 28, n° 2, March 2006, http://dx.doi.org/10.1007/s10703-006-7844-8.

[12] P. POTOP-BUTUCARU, B. CAILLAUD. *Correct-by-Construction Asynchronous Implementation of Modular Synchronous Specifications*, in "Fundamenta Informaticae", vol. 77, 2006.

## Publications in Conferences and Workshops

[13] E. BADOUEL, M. BEDNARCZYK, A. BORZYSZKOWSKI, B. CAILLAUD, P. DARONDEAU. *Concurrent Secrets*, in "8th Workshop on Discrete Event Systems, WODES'06, Ann Arbor, Michigan, USA", S. LAFORTUNE, F. LIN, D. TILBURY (editors). , July 2006, http://www.irisa.fr/s4/download/papers/bbbcd-wodes-06.pdf.

[14] E. BADOUEL, M. TCHOUPÉ. *Cohérence de vues dans les grammaires algébriques*, in "Actes du CARI 2006", K. ASSOGBA, E. BADOUEL, Y. SLIMANI (editors). , 2006, p. 115-122, http://www.cari-info.org.

[15] A. BENVENISTE, B. CAILLAUD, L. CARLONI, P. CASPI, A. SANGIOVANNI-VINCENTELLI. *Communication by Sampling in Time-Sensitive Distributed Systems*, in "Proceedings of the Sixth Annual ACM Conference on Embedded Software, EMSOFT'06", ACM Press, 2006.

[16] T. JERON, H. MARCHAND, S. PINCHINAT, M.-O. CORDIER. *Supervision Patterns in Discrete Event Systems Diagnosis*, in "8th Workshop on Discrete Event Systems, WODES'06, Ann Arbor, Michigan, USA", July 2006.

[17] J. KOMENDA, H. MARCHAND, S. PINCHINAT. *A Constructive and Modular Approach to Decentralized Supervisory Control Problems*, in "3rd IFAC Workshop on Discrete-Event System Design, DESDes'06, Rydzyna Castle, Poland", September 2006.

## Miscellaneous

[18] P. DARONDEAU, B. GENEST, L. HÉLOUËT. *Mixed Product of Message Sequence Charts*, September 2006, Deliverable to the CRE CO2 collaboration.

# References in notes

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

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

[21] E. BADOUEL, A. BENVENISTE, B. CAILLAUD, R. PASSERONE. *SPEEDS SP2-WP2.1 : Heterogeneous Rich Component Definition, MAthematical Semantics*, SPEEDS working document, October 2006.

[22] P. DARONDEAU, B. GENEST, L. HÉLOUËT. *Mixed Product of Message Sequence Charts*, September 2006, Deliverable to the CRE CO2 collaboration.

[23] THE SPEEDS CONSORTIUM. *D.2.1.a HRC Definition (Basic Syntax)*, SPEEDS deliverable, September 2006.