INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team s4

System Synthesis and Supervision, Scenarios

Rennes

THEME 1C

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 [CR, INRIA]

Administrative assistant

Huguette Béchu [TR, INRIA, part-time in S4]

Research scientists

Éric Badouel [CR, INRIA, since September 1, 2003]
Albert Benveniste [DR, INRIA, part-time in S4]
Philippe Darondeau [DR, INRIA]

Project technical staff

Pierre Le Maigat [INRIA (funded by IST Project COLUMBUS), from October 1, 2003, until November 28, 2003]

Faculty members (University of Rennes 1)

Gilles Lesventes [Lecturer]
Sophie Pinchinat [Lecturer]

Ph. D. students

Guillaume Feuillade [ENS Cachan until August 31, 2003, INRIA since September 1, 2003]
Stéphane Riedweg [INRIA (partially funded by Brittany Regional Authority) until September 30, 2003, teaching assistant at the University of Rennes 1 since October 1, 2003]

Post-doctoral fellow

Dumitru Potop-Butucaru [INRIA (funded by IST Project ARTIST), since January 6, 2003]

2. 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 telecommunication and embedded systems. The behavioral views of the *Unified Modeling Language* [40] (sequence diagrams and statecharts), the *High-Level Message Sequence Charts* [34] 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 development of real-time software to be deployed over distributed architectures, such as telecommunication systems, complex control systems (automotive or avionics), flexible production systems, work-flow, etc.

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 being 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.

A particular effort to develop and transfer software prototypes: Tools have been developed which demonstrate the results of our research on (i) Petri net synthesis and (ii) scenarios languages.

All these elements are detailed below:

2.1.1. A focus on a precise type of applications: The development of real-time software to be deployed over distributed architectures

2.1.1.1. 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).

2.1.1.2. Deployment on a distributed architecture.

Domain specific software platforms, known as *middleware*, are now part of the usual software design process in industry, especially in telecommunication [32][39][41][31][30]. They offer 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.

Our research is focused on several problems related to the context described above:

2.1.2. Service adaptation and control

The telecommunication industry is often facing the problem of the integration of new features in existing protocol stacks [35][33]. This is a most difficult problem which requires costly changes to the software, and later, even more costly testing of the end-to-end service and of the possibly unexpected interactions between features. As of today, integration of new features is done directly on the implementation and not on the requirements nor on the detailed specifications.

Our research contributes to the development of methods and tools which assist the adaptation and control of services, at the level of requirement or design specifications.

2.1.3. Deployment on specific distributed architectures

The correctness of the synthesized communication and control depends only on generic properties of the underlying middleware. This allows to cover large classes of middlewares, instead of one middleware, specific to one application domain. We take into account simple functional properties of the middleware (for instance, reliable or lossy channels). We are also taking into account very particular temporal properties of the service to be deployed and of the middleware (periodic communications, bounded transmission time, etc.).

2.1.4. Component based design, using heterogeneous specification formalisms

The *unified modeling language* (UML) [40] offers a large and heterogeneous set of specification formalisms: architectural (*class* and *deployment* diagrams) or, behavioral (*sequence* and *state* diagrams). Our ambition is to provide both a formal semantics to subsets of these formalisms, and effective and correct mappings between them.

Requirements of several kinds can be expressed in these formalisms: functionality (synchronization, conflict, communication), control (safety, reachability, liveness), architectural (mapping, segregation) and quantitative performances (response time, throughput). The main problem is to analyze and transform system specifications expressed in these formalisms.

2.1.5. Research tracks, scientific foundations

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 inconsistent ?) and mapped to lower level specifications (for instance, communicating automata, or Petri nets).

The scientific method 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 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 in 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 tracks:

2.1.6. Petri net synthesis

This track follows up the main research theme of the former Team PARAGRAPHE of INRIA Rennes. In addition to further developments of the theory, applications in several fields are being investigated (automated production systems, work-flow engineering, component based software engineering).

2.1.7. Scenario languages

Current research work concentrate on the composition of system views expressed in scenario formalisms such as *High-Level Message Sequence Charts* (HMSC) [34].

2.1.8. Weakly-synchronous systems

This track contributes to the extension, to distributed systems, of the well-established synchronous paradigm. 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.

2.1.9. Classification et resolution of control problems through the quantified mu-calculus

Many 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

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, defined extensively by a set of states and a set of transitions) or, implicit (symbolic transition rules, usually parametrized 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* [40] or *Message Sequence Charts* [34]).

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.

For more information on the numerous models of concurrency, the reader is referred to:

1. Jan van Leeuwen (ed.), *Handbook of Theoretical Computer Science - Volume B: Formal Models and Semantics*, Elsevier, 1990.
2. Wolfgang Reisig and Grzegorz Rozenberg (eds.), *Lectures on Petri nets: advances in Petri nets*, Lecture Notes in Computer Science, Vol. 1491, 1492, Springer, 1998.
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 uses 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 automata, languages, or even *High-Level Message Sequence Charts* [34] 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 contribute 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 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

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

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 7.1).

Our work on weakly-synchronous 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 reading or writing periodically (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. The objective of the IST European project COLUMBUS (Section 7.2) is to provide a theoretical and methodological framework in which the correct mapping of synchronous designs to such particular distributed architectures can be best understood.

Our work on Petri net synthesis (Section 6.1), and the PN synthesis tool SYNET (Section 5.1) have found applications is 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. Software

5.1. Synet: A general Petri net synthesis tool

Participant: Benoît Caillaud.

SYNET is a multivalent toolbox integrating general Petri net synthesis algorithms. The toolbox allows to synthesize Petri nets from finite automata and regular languages. Synthesis from *High-Level Message Sequence charts* (HMSC) has been implemented recently, yet it has not been integrated in the toolbox. The tool as been distributed to a limited group of academic users in several fields of applications: control and optimization of work-flow systems, control of automated production systems and automatic synthesis of interfaces for software components.

6. New Results

6.1. Petri nets: synthesis and control

Participants: Éric Badouel, Benoît Caillaud, Philippe Darondeau, Guillaume Feuillade.

Key words: *Petri net*, *marked graphs*, *path-automatic specifications*, *synthesis*, *supervisory control*.

Glossary

Marked graph A marked graph is an ordinary Petri net where each place has exactly one input transition and one output transition.

Path-automatic specifications Path-automatic specifications are rational presentations of sets of finite or infinite graphs, given by a regular set of paths and rational relations on this set. They cover for instance trace domains, modal transition systems, and pushdown automata.

Synthesis The Petri net synthesis problem consists in deciding, constructively, from a given labeled transition system, whether it is isomorphic to the reachable state graph of some initialized Petri net.

Region The regions of a labeled transition system are the morphisms that map this graph to the Cayley graph of the group of integers, restricted on the non-negative nodes. The regions of a graph are the places of the associated Petri net.

Supervisory control A supervisor is a master system that may prevent the occurrence of some controllable transitions in a slave system based on the record of observable transitions of the slave system.

The work started last year on the synthesis of Petri nets from automatic graphs has been pursued and extended. We consider now path-automatic specifications as follows. Given an alphabet of actions, a specification comprises: a regular subset W of path labels (words on this alphabet), two rational relations on W defining which pairs of paths may not, resp. must, be confluent, and for each action, two rational relations on W defining which occurrences of this action may, resp. must, be present in a model of the specification (models

are graphs). We were able to show a decision of the problem: does a given path-automatic specification have some Petri net model (*i.e.*, some model isomorphic to the reachable state graph of a Petri net)? This result opens a new perspective of Petri net synthesis, since it may now be applied to ambiguous specifications, halfway between transition systems and modal logic specifications. A paper co-authored by Éric Badouel and Philippe Darondeau has been submitted to a journal. Guillaume Feuillade is now trying to go further along this direction, by considering the synthesis of Petri nets from non-disjunctive modal formulas with only greatest fix-points.

We have solved an open problem on marked graphs due to W. Reisig. The problem was to prove constructively that for any bounded marked graph (or more generally, T-system), there exists a labeled one-safe marked graph (resp. T-system) with the same language. The construction which we propose starts with a decomposition of the marked graph into sequential processes, using ideas from *FIFO* nets after a suitable coloring of the tokens, proceeds by a finite unfolding of the cyclic processes based on least common multiples of their periods, and ends with imposing a fixed cyclic synchronization to all the resulting processes. A paper co-authored by Philippe Darondeau and Harro Wimmel (Univ. of Oldenburg) has been submitted to a journal.

We have finally made some progress on elementary nets synthesis, by showing a universal embedding of *partial 2-structures* (or equivalently, finite labeled transition systems) into *full and forward closed set 2-structures* (or equivalently, elementary nets in which transitions form a partial group, where each transition has an inverse and the product of two transitions that may be fired consecutively is a transition). A paper co-authored by Andrzej Borzyszkowski (IPI PAN, Gdansk) and Philippe Darondeau has been written and will appear soon as a research report of the Polish Academy of Sciences.

6.2. Heterogeneous reactive systems

Participants: Albert Benveniste, Benoît Caillaud, Dumitru Potop-Butucaru.

Key words: *synchronous, Kahn's networks, distributed architectures, endochrony, isochrony, loosely synchronous architectures*.

In the framework of the COLUMBUS project (see Section 7.2) we have developed a systematic method to formally model heterogeneous reactive systems. This is joint work with Alberto Sangiovanni-Vincentelli and Luca Carloni (U.C. Berkeley and PARADES) and Paul Caspi (VERIMAG).

The motivation is twofold. On the one hand, heterogeneous models are encountered throughout the design flow for embedded systems: use of UML notations, of Simulink-Stateflow, of synchronous languages. On the other hand, execution architectures for deployment generally follow a *model of computation* that is different from that of the modeling tools. For example, whereas the Time-Triggered Architecture (TTA) by H. Kopetz [36] strictly obeys the synchronous model, this is no longer the case for other commonly used infrastructures (field buses, CAN, ARINC, etc.). In 2002, we analyzed the Loosely Time-Triggered Architecture (LTTA), that is in use at Airbus.

To address this issue of heterogeneity in a formal way, we started from the so-called *tag system* model originally due to Edward Lee and Alberto Sangiovanni-Vincentelli. We have simplified and restricted this model to our needs. The new version covers synchronous and asynchronous models, timed and untimed models, and their free combination. We have formally defined what it means to migrate from one model to another. We have formally defined what heterogeneous parallel composition means (*e.g.*, what $P||Q$ means, for P synchronous and Q asynchronous. We have formally defined what it means to preserve semantics, *e.g.*, when migrating from a synchronous to a *globally asynchronous, locally synchronous* design (GALS). We have characterized, by algebraic means, those designs that preserve semantics when deployed on an infrastructure which *model of computation* differs.

These results nicely complement the previous results from our group on desynchronization and endochrony/isochrony. They have been published in [15].

On another direction, Dumitru Potop-Butucaru and Benoît Caillaud have found an error [29] in our long *Information and Computation* paper on desynchronization [3]. Isochrony is not compositional, unlike claimed in this paper (other results are correct). In the process of correcting this, a totally new theory has emerged for

correct GALS deployment for more than two components. It is currently under study and its results will be reported next year.

6.3. Classification and resolution of control problems through the quantified mu-calculus

Participants: Sophie Pinchinat, Stéphane Riedweg.

Key words: *control, discrete event system, partial observation, communicating system, logics, mu-calculus, tree automata, winning strategy, parity game*.

The theory of control synthesis introduced by Ramadge and Wohnam is a generic method which can be described as follows: given a program and some expected behavior, known as *control objective*, the goal is to produce, by automated methods, a device (*e.g.*, another program) with two main properties. First, this device must fulfill some constraints (*e.g.*, it should belong to some particular class of programs), and second, it should be able to control the original program (*e.g.*, by synchronous composition) in order to achieve the required behavior.

We have developed a logical formalism as a general formal language for the specification of control problems. The proposed framework extends the Mu-Calculus, a extremely expressive modal logic with fix-points operators, introduced by Dexter Kozen: we allow for quantifications over atomic propositions, yielding to a second order logic. We have established that *checking for the existence of a solution to the control problem is equivalent to perform verification of formulas*. Verification of formulas is often called *Model-Checking*. However, the is logic undecidable, as the decentralized control problems under partial observation can be expressed therein. We have explored various fragments of the logic. The fragments reveal to be expressive enough to specify interesting control problems, but small enough to remain decidable. An accurate study of the complexity of the satisfiability and model-checking problems, in these logical fragments, has been carried out.

In [19], we consider the fragment corresponding to the setting where the moves of the systems to be controlled are fully observable. These are the so called *control problems under total observation*. The logical setting offers a uniform way to describe, as parameters, the kind of system (closed or open), the control objective, the type of interaction between the controller and the system, optimality criteria (fairness, maximally permissive), etc. To our knowledge, none of the former approaches can capture such a wide range of concepts. Moreover, we have established that model-checking this fragment is decidable and that *the synthesis of controllers* can be obtained on the basis of the underlying model checking procedure.

In [26], the case of control requirements for systems under partial observation is studied. We have focused on a fragment expressive enough to specify the unobservable sets of events of (decentralized) controllers, and to allow for the joint unobservability and controllability of an event. We have identified the set of formulas representable by infinite tree automata. Technically, the automata constructions are borrowed from the work of André Arnold et al. [28]. For formulas expressing control requirements, any model of the associated automaton provides an adequate controller. For example, given any Mu-Calculus definable control objective, a maximal permissive controller in some class of controllers under partial observation can be specified by a formula and synthesized in time 3EXP in the size of the formula.

This logical framework brings a new vision of the field, and makes discrete event system control theory much clearer. In particular, it provides a rigorous classification of control problems. Our logical framework is also expected to be relevant to problems related to control theory, such as diagnosis or test generation. This will be the objective of this continuing research work.

7. Contracts and Grants with Industry

7.1. Ouate: Tools for the composition and analysis of timed HMSCs

Participant: Benoît Caillaud.

This collaboration with France Telecom Research and Development, in Lannion (OUATE, contract 101C04550031334061, 2001–2003), has allowed to develop techniques and tools for the analysis and composition of timed *High-Level Message Sequence Charts* (HMSC) [34]. A performance analysis tool has been developed in 2002 [37].

In 2003, we have contributed (in collaboration with Loïc Hélouët, Team TRISKELL) to the problem of analyzing the behavior of a system described by a set of local views, expressed by HMSCs. For this purpose, a categorical approach to HMSC composition has been developed, re-using the *pull-back* of *asynchronous transition systems* proposed in [14][22]. In this framework, interaction between two views (*i.e.*, HMSCs) is defined by a pair of morphisms from the two views to an *interface* view. The composition of two interacting views is the pullback (or fibered product) of the two views (with their interaction morphisms). The resulting view (HMSC) is a limit construction: it projects in the two views in a manner that is consistent with the interaction view. This research work will continue in 2004. We are currently negotiating, with France Telecom, a followup to this collaboration on that specific topic.

7.2. Columbus: embedded software design for mission-critical systems

Participants: Albert Benveniste, Benoît Caillaud, Dumitru Potop-Butucaru.

The COLUMBUS project (IST-2001-38314, 2002–2004) is a light-weight project, involving teams from both Europe and USA. In this project, our team mainly cooperates with the teams of Alberto Sangiovanni-Vincentelli (PARADES and U.C. Berkeley) and Janos Sztipanovits (Vanderbilt U., USA). The focus is on fundamental studies related to the overall design flow for embedded systems. Our research on heterogeneous modeling is part of that [15][17]. Janos Sztipanovit's team is currently working on a specific domain for the Signal reactive synchronous language, in their GME meta-modeling [38].

7.3. Artist: Embedded real-time systems

Participant: Albert Benveniste.

The ARTIST network of excellence (IST-2001-34820, 2002–2004) is a FP5 network in the area of embedded systems. It is headed by Joseph Sifakis from VERIMAG, Grenoble. ARTIST is composed of three actions: Hard-Real Time Systems (headed by Albert Benveniste), Component-based Design and Development (headed by Bengt Jonsson, Uppsala, Sweeden), and Adaptive Real-Time Systems for Quality of Service Management (headed by Giorgio Buttazzo, Pavia, Italy). This year, the main result of our team within ARTIST has been the ARTIST road-map on Hard Real-Time[20].

8. Other Grants and Activities

8.1. Catalysis: categorical and algebraic approaches to synthesis

Participants: Éric Badouel, Benoît Caillaud, Philippe Darondeau.

CATALYSIS, which stands for categorical and algebraic approaches to synthesis, is 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, Benoît Caillaud and Philippe Darondeau for Team S4 and Marek Bednarczyk, Andrzej Borzyszkowski and Wieslaw Pawlowski for IPI PAN. A two-week visit in Gdansk of two members of the S4 project (in December) and a two-week visit in Rennes of two members of IPI PAN (in May) are planned yearly. This collaboration has enabled the publication of one conference paper [14] and two research reports [23][22].

9. Dissemination

9.1. Participation to editorial boards and program committees

Philippe Darondeau served as a program committee member for the conference ICALP 2003. He co-organized with Sadatoshi Kumagai (U. of Osaka) the Workshop on Discrete Event Systems Control at the conference ATPN 2003. Philippe Darondeau is serving as a program committee member for the conference STACS 2004.

Albert Benveniste is Associate Editor at Large (AEAL) for the *IEEE Trans. on Automatic Control*, and member of the editorial boards of *Proceedings of the IEEE* and *Discrete Event Dynamic Systems: Theory and Applications.* This year, he has been member of the program committee of TACAS, MOVEP, MSR. He has been a plenary speaker at CONCUR. He has been invited speaker at FMCO. He has been invited to become a member of the ISR Strategic Advisory Council (ISR is a center of excellence of the University of Maryland, USA, headed by Eyad Abed).

Benoît Caillaud is serving as program committee member for the SLAP 2004 workshop on synchronous programming languages.

9.2. 68NQRT: Theory of computing seminar of Irisa

Sophie Pinchinat organizes the 68NQRT seminar session of IRISA. Each session consists in scientific talks given by local or invited speakers, in the following research areas: software engineering, theoretical computer science, discrete mathematics, artificial intelligence. This year, up to 18 talks have been given, of which 11 by invited speakers.

9.3. 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*.
  - First year: Sophie Pinchinat is in charge of a student project course on *reactive systems design*.
- Second year undergraduate: Sophie Pinchinat is teaching a student project course on the *design of mobile robot systems in a virtual environment and object-oriented programming*. This course is in connection with the ARTIST Education Project — See 7.3.

9.4. Other elements of scientific life

Philippe Darondeau presented his work with Éric Badouel on path-automatic specifications at a meeting of the IFIP-WG2.2 in Amsterdam (May 2003).

10. Bibliography

Major publications by the team in recent years

[1] E. BADOUEL, P. DARONDEAU. *Theory of regions.* in « Lectures on Petri Nets I: Basic Models », series Lecture Notes in Computer Science, volume 1491, Springer, 1999, pages 529-586.

[2] A. BENVENISTE. *Some Synchronization Issues When Designing Embedded Systems from Components.* in « Embedded Software, First International Workshop, EMSOFT 2001 », series Lecture Notes in Computer

Science, volume 2211, Springer, T. HENZINGER, C. M. KIRSCH, editors, pages 32-49, Tahoe City, CA, USA, October, 2001.

[3] A. BENVENISTE, B. CAILLAUD, P. LE GUERNIC. *Compositionality in dataflow synchronous languages: specification and distributed code generation.* in « Information and Computation », volume 163, 2000, pages 125-171.

[4] A. BENVENISTE, C. JARD, S. GAUBERT. *Algebraic techniques for timed systems.* in « CONCUR'98, Concurrency Theory, 9th International Conference », series Lecture Notes in Computer Science, volume 1466, Springer, D. SANGIORGI, R. DE SIMONE, editors, pages 373-388, Nice, France, September, 1998.

[5] B. CAILLAUD, P. DARONDEAU, L. HÉLOUËT, G. LESVENTES. *HMSCs as specifications... with PN as completions.* F. CASSEZ, C. JARD, B. ROZOY, M. DERMOT, editors, in « Modeling and Verification of Parallel Processes », series Lecture Notes in Computer Science, volume 2067, Springer, 2001, pages 125-152, http://link.springer.de/link/service/series/0558/bibs/2067/20670125.htm.

[6] P. DARONDEAU. *On the Petri net realization of context-free graphs.* in « Theoretical Computer Science », number 1-2, volume 258, 2001, pages 573-598.

[7] P. LE MAIGAT, L. HÉLOUËT. *A (max,+) approach for time in message sequence charts.* in « Proceedings of the 5th Workshop on Discrete Event Systems », Kluwer Academic Publishers, R. BOEL, G. STREMERSCH, editors, pages 83-92, Ghent, Belgium, 2000.

[8] H. MARCHAND, S. PINCHINAT. *Supervisory Control Problem using Symbolic Bisimulation Techniques.* in « 2000 American Control Conference », pages 4067-4071, Chicago, Illinois, USA, June, 2000.

Books and Monographs

[9] *Proceedings of the ATPN-Workshop on Discrete Event Systems Control.* P. DARONDEAU, S. KUMAGAI, editors, Tech. Univ. Eindhoven, Netherlands, series BETA-RR, June, 2003.

Doctoral dissertations and "Habilitation" theses

[10] S. PINCHINAT. *Contributions à l'Analyse et au Contrôle des Systèmes Réactifs.* Thèse d'habilitation à diriger les recherches, Université de Renne 1, école doctorale MATISSE, December, 2003.

[11] S. RIEDWEG. *Logiques pour le contrôle d'automatismes discrets.* Thèse de doctorat, Université de Renne 1, école doctorale MATISSE, December, 2003.

Articles in referred journals and book chapters

[12] 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 », number 1, volume 91, 2003, pages 64–83, Special issue on modeling and design of embedded software.

[13] P. DARONDEAU, X. XIE. *Linear Control of Live Marked Graphs.* in « Automatica », number 3, volume 39, 2003, pages 429–440, http://dx.doi.org/10.1016/S0005-1098(02)00266-2.

Publications in Conferences and Workshops

[14] M. A. BEDNARCZYK, L. BERNARDINELLO, B. CAILLAUD, W. PAWLOWSKI, L. POMELLO. *Modular system development with pullbacks.* in « Applications and Theory of Petri Nets 2003 », series Lecture Notes in Computer Science, volume 2679, Springer, pages 140–160, June, 2003, http://link.springer.de/link/service/series/0558/bibs/2679/26790140.htm.

[15] 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 », series Lecture notes in computer science, volume 2855, Springer, R. ALUR, I. LEE, editors, pages 35–50, October, 2003.

[16] G. FEUILLADE, T. GENET. *Reachability in conditional term rewriting systems.* in « FTP 2003, International Workshop on First-Order Theorem Proving », Valencia, Spain, June, 2003.

[17] D. POTOP-BUTUCARU. *The Kahn Principle for Networks of Synchronous Endochronous Programs.* in « Proceedings of the 1st International Workshop on Formal Methods for Globally Asynchronous Locally Synchronous Architectures (FMGALS 2003) », pages 123–132, Pisa, Italy, September, 2003, http://www.DumitruPotop.net/Doc/potopFMGALS2003.pdf.

[18] D. POTOP-BUTUCARU, R. DE SIMONE. *Optimizations for Faster Execution of Esterel Programs.* in « Proceedings of the 1st ACM/IEEE International Conference on Formal Methods and Models for Co-Design (MEMOCODE 2003) », pages 227–236, Mont-Saint-Michel, France, June, 2003.

[19] S. RIEDWEG, S. PINCHINAT. *Quantified mu-calculus for control synthesis.* in « MFCS 2003, 28th International Symposium on Mathematical Foundations of Computer Science », series Lecture notes in computer science, volume 2747, Springer, pages 642–651, August, 2003.

Internal Reports

[20] ARTIST. *Hard Real-Time Development Environments.* Roadmap report, number W1.A1.N1.Y1, Artist, June, 2003, http://www.systemes-critiques.org/ARTIST/Roadmaps/A1-roadmap.pdf.

[21] E. BADOUEL, P. DARONDEAU. *The Petri net synthesis problem for automatic graphs.* Research report, number 4661, INRIA Rennes, December, 2002, http://www.inria.fr/rrrt/rr-4661.html.

[22] M. A. BEDNARCZYK, L. BERNARDINELLO, B. CAILLAUD, W. PAWLOWSKI, L. POMELLO. *Modular system development with pullbacks.* Research report, number 4828, INRIA Rennes, May, 2003, http://www.inria.fr/rrrt/rr-4828.html.

[23] A. BORZYSZKOWSKI, P. DARONDEAU. *Transition Systems without Transitions.* Manuscript, number 964, IPI PAN, October, 2003, http://www.ipipan.gda.pl/~andrzej/papers/s2s-ipi.ps.gz.

[24] P. DARONDEAU, H. WIMMEL. *From bounded T-systems to 1-safe T-systems up to language equivalence.* Research report, number 4708, INRIA Rennes, January, 2003, http://www.inria.fr/rrrt/rr-4708.html.

[25] G. FEUILLADE, T. GENET, V. VIET TRIEM TONG. *Reachability Analysis over Term Rewriting Systems.* Research report, number 4970, INRIA Rennes, October, 2003, http://www.inria.fr/rrrt/rr-4970.html.

[26] S. RIEDWEG, S. PINCHINAT. *Quantified Loop-mu-calculus for Control under Partial Observation.* Research report, number 4949, INRIA Rennes, September, 2003, http://www.inria.fr/rrrt/rr-4949.html.

[27] S. RIEDWEG, S. PINCHINAT. *Quantified Mu-calculus for Control Synthesis.* Research report, number 4793, INRIA Rennes, April, 2003, http://www.inria.fr/rrrt/rr-4793.html.

Bibliography in notes

[28] A. ARNOLD, A. VINCENT, I. WALUKIEWICZ. *Games for Synthesis of Controllers with Partial Observation.* in « Theoretical Computer Science », number 303, volume 1, June, 2003, pages 7–34.

[29] B. CAILLAUD, D. POTOP-BUTUCARU, A. BENVENISTE. *Erratum to: A. Benveniste, B. Caillaud, P. Le Guernic. Compositionality in Dataflow Synchronous Languages, Specification and Distributed Code Generation. Information and Computation 163, 125-171 (2000).* September, 2003, http://www.irisa.fr/prive/Benoit.Caillaud/erratum-ic-2003.pdf.

[30] *CORBA components.* Object Management Group, March, 1999, http://www.omg.org/.

[31] *The common object request broker: architecture and specification.* Object Management Group, March, 1999, http://www.omg.org/.

[32] *Proceedings of Middleware'98, IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing.* N. DAVIES, K. RAYMOND, J. SEITZ, editors, Springer, Lake District National Park, UK, September, 1998.

[33] C. E., N. GRIFFETH, Y.-J. LIN, M. NILSON, W. SCHNURE. *A Feature Interaction Benchmark for IN and Beyond.* in « Feature Interactions in Telecommunications Systems », IOS press, pages 1–23, 1994.

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

[35] *FIW'98 Feature Interactions in Telecommunications and Software Systems, V.* K. KIMBLER, W. BOUMA, editors, IOS Press, Lund, Sweden, September, 1998.

[36] H. KOPETZ. *Real-Time Systems, Design Principles for Distributed Embedded Applications.* Kluwer Academic Publishers, 1997.

[37] P. LE MAIGAT. *Techniques algébriques max-plus pour l'analyse des performances temporelles de systèmes concurrents.* thèse de doctorat, université de Rennes 1, école doctorale Matisse, septembre, 2002, ftp://ftp.irisa.fr/techreports/theses/2002/lemaigat.pdf.

[38] A. LEDECZI, M. MAROTI, A. BAKAY, G. KARSAI, J. GARRETT, C. THOMASON, G. NORDSTROM, J. SPRINKLE, P. VOLGYESI. *The Generic Modeling Environment.* in « Workshop on Intelligent Signal

Processing », Budapest, May, 2001, http://www.isis.vanderbilt.edu/Projects/gme/GME2000Overview.pdf.

[39] *Middleware 2000 · IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing.* J. SVENTEK, G. COULSON, editors, series LNCS, volume 1795, Springer, New York, NY, USA, April, 2000.

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

[41] *ODP Reference model: foundations and architecture.* ITU-T Recommendations X.902 and X.903, ISO/IEC 10746, International Telecommunication Union, November, 1995, http://www.itu.int/home/index.html.