INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team s4

System Synthesis and Supervision, Scenarios

Rennes

THEME COM

Activity Report

2004

Table of contents

1. Team **1**

2. Overall Objectives **1**

2.1.1. A focus on a precise type of applications: The development of real-time software to be deployed over distributed architectures 2

2.1.1.1. System specification. 2

2.1.1.2. Deployment on a distributed architecture. 2

2.1.2. Service adaptation and control 2

2.1.3. Deployment on specific distributed architectures 2

2.1.4. Component based design, using heterogeneous specification formalisms 2

2.1.5. Research tracks, scientific foundations 2

2.1.6. Petri net synthesis 3

2.1.7. Scenario languages 3

2.1.8. Weakly-synchronous systems 3

2.1.9. Classification et resolution of control problems through the quantified mu-calculus 3

3. Scientific Foundations **3**

4. Application Domains **4**

5. Software **5**

5.1. Synet: A general Petri net synthesis tool 5

6. New Results **5**

6.1. Petri nets: synthesis and control 5

6.2. Heterogeneous reactive systems 7

6.3. Classification and resolution of control problems through the quantified mu-calculus 7

7. Contracts and Grants with Industry **8**

7.1. CO2: Composition of viewpoints based on scenario languages 8

7.2. Columbus: embedded software design for mission-critical systems 8

8. Other Grants and Activities **9**

8.1. ARTIST – Network of Excellence on Advanced Real-Time Systems 9

8.2. ARTIST2 – Network of Excellence on Embedded Systems Design 10

8.3. Catalysis: categorical and algebraic approaches to synthesis 10

9. Dissemination **10**

9.1. Participation to editorial boards and program committees 10

9.2. 68NQRT: Theory of computing seminar of Irisa 11

9.3. Teaching 11

10. Bibliography **11**

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]
Albert Benveniste [DR, INRIA, part-time in S4]
Philippe Darondeau [DR, INRIA]

Faculty member (University of Rennes 1)
Sophie Pinchinat [Lecturer]
Stéphane Riedweg [Teaching Assistant, until 31 August 2004]

Ph. D. students
Guillaume Feuillade [INRIA]
Jean-Baptiste Raclet [INRIA, from 2 November 2004, funded by the collaboration CO2 with France Telecom]

Post-doctoral fellow
Dumitru Potop-Butucaru [INRIA (funded by IST Project COLUMBUS), until 31 December 2004]

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* [30] (sequence diagrams and statecharts), the *High-Level Message Sequence Charts* [27] 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 [26][29][31][25][24]. 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.

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 [28][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 middleware, instead of one single middleware, specific to one single application domain. We take into account simple functional properties of the middleware (for instance, reliable or lossy channels). We are also considering very specific 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) [30] 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 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 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 concentrates on the composition of system views expressed in scenario formalisms such as *High-Level Message Sequence Charts* (HMSC) [27].

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* [30] or *Message Sequence Charts* [27]).

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. 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 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 automata, languages, or even *High-Level Message Sequence Charts* [27] 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 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. 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 Petri net 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 has 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

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

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

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 two years ago 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 exhibit a decision procedure of the following 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 published [11]. Guillaume Feuillade is now working along this direction, by considering the synthesis of Petri nets from non-disjunctive modal formulas with only greatest fix-points.

Badouel has pursued the development of an algebraic theory of generalized Petri nets [16][21]. In this theory, the firing rule of nets relies on a residuation operation for the commutative monoid of natural number. A class of residuated commutative monoids, called Petri algebras, has been identified. In these monoids, one can mimic the token game of Petri nets to define the behaviour of generalized Petri net whose flow relation and place contents are valued in such algebraic structures. The sum and its associated residuation capture respectively how resources within places are produced and consumed through the firing of a transition. Petri algebras have been shown to coincide with the positive cones of lattice-ordered commutative groups and constitute the subvariety of the (duals of) residuated lattices generated by the commutative monoid of natural number. However, We have exhibited a Petri algebra whose corresponding class of nets is strictly more expressive than the class of Petri nets. More precisely, we introduce 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 is universal in the sense that any net associated with some Petri algebras 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 ...) are undecidable on the class of lexicographic Petri nets. Finally we turn our attention to bounded nets associated with Petri algebras and show that their dynamic can be reformulated in term of MV-algebras.

Philippe Darondeau has published a survey paper on the synthesis of unbounded Petri nets [12]. The paper addresses the problem of deciding uniformly for graphs or languages of a given class whether they are generated by unlabelled Place-Transition nets whose sets of reachable markings may be infinite.

New research on the Petri net synthesis problem for Mu-Calculus definable specifications has been undertaken: In the theory of the modal Mu-Calculus, hence interpreted over labeled transition systems, or Kripke structures, the *satisfiability problem* consists in the following: given a Mu-Calculus sentence, find a labeled transition systems which satisfies the sentence. This problem is decidable (in EXPTIME-complete) since it reduces to the emptiness problem of alternating parity tree automata; as a consequence, each satisfiable sentence has a finite model. However, in cases where concurrent and distributable models are required, the satisfiability problem of Mu-Calculus sentences becomes a lot more intricate since we cannot rely on the finite model property anymore. For the case of unlabeled Petri nets, we have established the undecidability of the satisfiability problem (by a reduction of the Minsky machines language emptiness problem). We next have designed a syntactical fragment of the Mu-Calculus, called the *conjunctive Nu-Calculus*, which sentences satisfiability can sometimes be automatized. The results strictly extend exiting works on Petri net synthesis from regular languages and are incomparable with existing extensions to automatic specifications. Ongoing work focuses on determining whether the satisfiability problem of the full conjunctive Nu-Calculus is decidable or not; we aim at extending further the results to get the widest class of temporal and modal logic specifications with an effective Petri nets synthesis.

6.2. Heterogeneous reactive systems

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

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

In the framework of the COLUMBUS project (see Section 7.2) and of the ARTIST network of excellence (see Sections 8.1 and 8.2), we have developed a systematic method to formally model heterogeneous reactive systems. This is joint work with Alberto Sangiovanni-Vincentelli, 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) of H. Kopetz [34] strictly obeys the synchronous model, this is no longer the case for other commonly used infrastructures (field buses, CAN, ARINC, etc.). In 2002-2003, 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 taylored this model to our needs [17]. The new version covers not only synchronous and asynchronous models, timed and untimed models, and their free combination, but also is able to model causality dependence relations, such as those induced by data- and control-flow dependencies. 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.

In another direction, Dumitru Potop-Butucaru and Benoît Caillaud have developped a totally new theory for the correct deployment of synchronous designs over globally asynchronous, locallly synchronous (GALS) architectures. We have introduced the notion of weak endochrony, which extends to a synchronous setting the classical theory of Mazurkiewicz traces. The notion is useful in the synthesis of correct-by-construction communication protocols for GALS systems. The independence between various computations can be exploited here to provide communication schemes that do not restrict concurrency while still guaranteeing correctness. Such communication schemes are then lighter and more flexible than existing latency-insensitive or endo/isochronous counterparts [22][19].

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

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

Participant: Sophie Pinchinat.

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 fixpoints 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 logic is 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 has been carried out, in these logical fragments [35]. This logical framework brings a new vision of the field, and makes discrete event system control theory much clearer.

Maximal permissiveness is a qualitative property of the controllers meaning a control which disallows only what needs being forbidden, but no more. This is a highly relevant issue, as for safety properties trivial solutions always suit. In 2004, we have obtained new results [20] on maximal permissive control. The maximal permissiveness of a controller can be seen as a kind of "pattern formula" independent of the system. This pattern formula has a fixed size, and only needs to be filled with the control objectives property (which actually is any Mu-Calculus formula). The significant contribution of the work is twofold: (1) a decision procedure is proposed to compute maximally permissive controllers, if any, even in frameworks where such solutions do not exist in general. This is opposed to e.g. the regular languages frameworks, and even ω-regular languages, where a unique supremal controllable sub-language always exists, whatever the system and the control objectives are. Whereas when branching-time logics, as the Mu-Calculus, are used to specify the control objectives, easy examples show that the maximal permissiveness is not granted. (2) we have established the strict expressiveness of our approach regarding other existing works in the literature

Non-deterministic systems can also be handled. This has been made possible by mapping non-deterministic systems to deterministic ones, with silent transitions, and by mapping formulas of the quantified mu-calculus interpreted on non-deterministic systems to formulas interpreted on deterministic systems [23].

7. Contracts and Grants with Industry

7.1. CO2: Composition of viewpoints based on scenario languages

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

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) [27].

In 2004, we have contributed (in collaboration with Loïc Hélouët, Team DISTRIBCOM) to the problem of analyzing the behavior of a system described by a set of local views, expressed by HMSCs [18]. For this purpose, a categorical approach to HMSC composition has been developed, re-using the *pull-back* of *asynchronous transition systems* proposed in [32]. 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.

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) was a light-weight project, involving teams from both Europe and USA. In this project, our team cooperated 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 [17][22][19]. Documents and reports related to this collaboration can be retrieved from the Columbus web-page: http://www.columbus.gr/

8. Other Grants and Activities

8.1. ARTIST – Network of Excellence on Advanced Real-Time Systems

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

IST-2001-34820, Contract INRIA *— April 1st, 2002, October 1st, 2005*

The objective of ARTIST is to Coordinate the R&D effort in the area of Advanced Real-time Systems so as to :

- Improve awareness of academics and industry in the area, especially about existing innovative results and technologies, standards and regulations.
- Define innovative and relevant work directions, identify obstacles to scientific and technological progress and propose adequate strategies for circumventing them.

ARTIST is organized into three actions:

1. *Hard Real-Time Systems.* Consolidate and further improve a strong European competence and know-how that is strategic for safety or mission critical applications (Synchronous languages-TTA- Fixed priority scheduling).
2. *Component-based Design and Development.* Transfer, enhance interaction between teams working on compositionality/composability problems and software and systems engineering teams involved in the definition of standards e.g. UML, SDL.
3. *Adaptive Real-Time Systems for Quality of Service (QoS) Management.* Soft real-time approaches and technology for telecommunications, large open systems and networks Teams with expertise in real-time operating systems and middleware.

The work directions are :

- Establishing a roadmap on future directions in advanced real-time systems.
- Proposing curricula for Education and Training in advanced real-time systems.
- Dissemination and International Collaboration.
- Creating strong two-way ties with industry.

Albert Benveniste is leader of action 1 on Hard Real-Time Systems.

8.2. ARTIST2 – Network of Excellence on Embedded Systems Design

Participants: Albert Benveniste, Benoît Caillaud.

IST-004527 ARTIST2, Starts September 2004; duration 48 months

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.

ARTIST2 has a double core, consisting of leading-edge research in embedded systems design issues (described later in this document) in the Joint Programme of Research Activities (JPRA), and complementary activities around shared platforms and staff mobility in the Joint Programme of Integration Activities (JPIA).

The JPRA activities are pure research, and the JPIA are complementary efforts for integration. Both work towards deep integration between the participating research teams.

The JPRA and JPIA are structured into clusters - one for each of the selected topics in embedded systems design (in red). Teams may be involved in one or several clusters.

Around this double core is the Joint Programme of Activities for Spreading Excellence (JPASE). These are complementary activiites for disseminating excellence across all available channels, targetting industry, students, and other European and international research teams.

ARTIST2 is structured into 8 clusters:

Modelling and Components.

Hard Real-Time. This is one of the two main approaches to embedded systems design. It is applied to systems where temporal constraints are critical. This includes applications, which are often safety-critical, such as applications in space, automobile, rail transport, air traffic control, production control, etc. This is the oldest and most mature approach, and has led to significant research results in Europe, such as synchronous languages, fixed-priority scheduling, the time-triggered architectures, and these have been significantly transferred and developed in industry (for example, SCADE and Esterel to Airbus, and TTA to automotive).

Adaptive Real-time.

Control for Embedded Systems.

Execution Platforms.

Testing and Verification.

Compilers and Timing Analysis

Albert Benveniste is leader of the cluster on Hard Real-Time Systems.

8.3. Catalysis: categorical and algebraic approaches to synthesis

Participants: Éric Badouel, Benoît Caillaud, Philippe Darondeau.

CATALYSIS, which stands for *categorical and algebraic approaches to synthesis*. It 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.

9. Dissemination

9.1. Participation to editorial boards and program committees

Philippe Darondeau served as chairman of the program committee of the conference ACSD 2004 for the ICALP 2003 conference. With Sadatoshi Kumagai (U. of Osaka), he co-organized the Workshop on Discrete

Event Systems Control at the ATPN 2003 conference. Philippe Darondeau is serving as a program committee member for the STACS 2004 conference.

Albert Benveniste is associated editor at large (AEAL) for the journal *IEEE Trans. on Automatic Control* and member of the editorial board of the journal and «Proceedings of the IEEE». He has been in 2004 member of the Program Committee of the following conferences: MOVEP, EMSOFT, and has been invited to the for 2005 the Program Committee of HSCC and CONCUR. He has served as en expert for the Strong Research Environment programme of the Swedish Research Council and is member of the Strategic Advisory Council of the Institute for Systems Research, Univ. of Maryland, College Park, USA.

Benoît Caillaud has served as program committee member for the SLAP 2004 workshop on synchronous programming languages. He is general chair of ACSD 2005: The Fifth International Conference on Application of Concurrency to System Design, to be held in St Malo in June 2005.

9.2. 68NQRT: Theory of computing seminar of Irisa

Sophie Pinchinat organizes the 68NQRT seminar session of IRISA. Each week, scientific talks are given by local or invited speakers, in the following research areas: software engineering, theoretical computer science, discrete mathematics, artificial intelligence.

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 8.1.

10. 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. JUHÃ¡S, 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. HÃ©louÃ«t, 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.

Books and Monographs

[10] M. Kishinevsky, P. Darondeau (editors)., IEEE Computer Society, 2004.

Articles in referred journals and book chapters

[11] E. Badouel, P. Darondeau. *The synthesis of Petri nets from path-automatic specifications*, in "Information and Computation", vol. 193, 2004, p. 117–135.

[12] P. Darondeau. *Unbounded Petri Net Synthesis*, J. Desel, W. Reisig, G. Rozenberg (editors)., Lecture Notes in Computer Science, vol. 3098, Springer, 2004, p. 413–438.

[13] G. Feuillade, T. Genet, V. Viet Triem Tong. *Reachability Analysis over Term Rewriting Systems*, in "Journal of Automated Reasoning", To appear., 2004.

[14] D. Potop-Butucaru, R. de Simone. *Formal Methods and Models for System Design*, R. Gupta, P. Le Guernic, S. Shukla, J.-P. Talpin (editors)., Kluwer, 2004, http://www.wkap.nl/.

[15] D. Potop-Butucaru, R. de Simone, J.-P. Talpin. *The Synchronous Hypothesis and Synchronous Languages*, R. Zurawski (editor)., To appear, CRC Press, 2004.

Publications in Conferences and Workshops

[16] E. BADOUEL, J. CHENOU. *Les réseaux de Petri commutatifs*, in "7th African Conference on Research in Computer Science, CARI'04", 2004.

[17] A. BENVENISTE, B. CAILLAUD, L. CARLONI, P. CASPI, A. SANGIOVANNI-VINCENTELLI. *Heterogeneous Reactive Systems Modeling: Capturing Causality and the Correctness of Loosely Time-Triggered Architectures (LTTA)*, in "Proceedings of the Fourth ACM International Conference on Embedded Software, EMSOFT'04", Lecture Notes in Computer Science, vol. September, Springer, September 2004, http://www.irisa.fr/s4/download/papers/Benveniste-emsoft2004.pdf.

[18] J. KLEIN, B. CAILLAUD, L. HÉLOUËT. *Merging Scenarios*, in "Proceedings of the Ninth International Workshop on Formal Methods for Industrial Critical Systems, FMICS'04, Linz, Austria", Electronic Notes in Theoretical Computer Science, to appear, September 2004, http://www.sciencedirect.com/science/journal/15710661.

[19] D. POTOP-BUTUCARU, B. CAILLAUD, A. BENVENISTE. *Concurrency in Synchronous Systems*, in "Proceedings of the International Conference on Application of Concurrency to System Design, ACSD 2004", 2004, http://www.irisa.fr/s4/download/papers/Potop_acsd2004.ps.gz.

[20] S. RIEDWEG, S. PINCHINAT. *Maximally Permissive Controllers in All Contexts*, in "WODES'04, 7th IFAC Workshop on Discrete Event Systems", September 2004, p. 283–288.

Internal Reports

[21] E. BADOUEL, J. CHENOU, G. GUILLOU. *Petri Algebras*, Research Report, n° 5355, INRIA, nov 2004, http://www.inria.fr/rrrt/rr-5355.html.

[22] D. POTOP-BUTUCARU, B. CAILLAUD, A. BENVENISTE. *Concurrency in Synchronous Systems*, Also published as IRISA Internal Publication PI-1605, Research Report, n° 5110, INRIA, feb 2004, http://www.inria.fr/rrrt/rr-5110.html.

[23] J.-B. RACLET, S. PINCHINAT. *The Control of Non-deterministic Systems*, PI, n° 1648, Irisa, October 2004.

Bibliography in notes

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

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

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

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

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

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

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

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

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

[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, 1994, p. 1–23.

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

[35] 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, August 2003, p. 642–651.