IN PARTNERSHIP WITH:
CNRS

Université Rennes 1

Activity Report 2012

Project-Team S4

System synthesis and supervision, scenarios

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Embedded and Real Time Systems

Table of contents

Project-Team S4

Keywords: Concurrency, Discrete Event Systems, Embedded Systems, Game Theory, Hybrid Systems, Synchronous Languages

Creation of the Project-Team: March 15, 2002 , *Updated into Team:* January 01, 2013 .

1. Members

Research Scientists

Eric Badouel [Junior Researcher, HdR]
Albert Benveniste [Senior Researcher, part-time in S4, HdR]
Benoît Caillaud [Team Leader, Junior Researcher, HdR]
Philippe Darondeau [Senior Researcher, HdR]

Faculty Member

Sophie Pinchinat [Professor, HdR]

PhD Students

Lamine Diouf [part-time in S4]
Stéphanie George [part-time in S4]
Bastien Maubert

Administrative Assistant

Angélique Jarnoux [TR, part-time in S4]

2. Overall Objectives

2.1. Overall Objectives

The objective of the project is the realization by algorithmic methods of reactive and distributed systems from partial and heterogeneous specifications. Methods, algorithms and tools are developed to synthesize reactive software from one or several incomplete descriptions of the system's expected behavior, regarding functionality (synchronization, conflicts, communication), control (safety, reachability, liveness), deployment architecture (mapping, partitioning, segregation), or even quantitative performances (response time, communication cost, throughput).

These techniques are better understood on fundamental models, such as automata, Petri nets, event structures and their timed extensions. The results obtained on these basic models are then adapted to those realistic but complex models commonly used to design embedded and telecommunication systems.

The behavioral views of the *Unified Modeling Language* (UML) (sequence diagrams and statecharts), the *High-Level Message Sequence Charts* (HMSC) and the synchronous reactive language Signal are the heart of the software prototypes being developed and the core of the technology transfer strategy of the project.

The scientific objectives of the project can be characterized by the following elements:

A focus on a precise type of applications: The design of real-time embedded software to be deployed over dedicated distributed architectures. Engineers in this field face two important challenges. The first one is related to system specification. Behavioral descriptions should be adaptable and composable. Specifications are expressed as requirements on the system to be designed. These requirements fall into four categories: (i) functional (synchronization, conflict, communication), (ii) control (safety, reachability, liveness), (iii) architectural (mapping, segregation) and (iv) quantitative (response time, communication cost, throughput, etc). The second challenge is the deployment of the design on a distributed architecture. Domain-specific software environments, known as *middleware* or *real-time operating systems* or *communication layers*, are now part of the usual software design process in industry. They provide a specialized and platform-independent distributed environment to higher-level software components. Deployment of software components and services should be done in a safe and efficient manner.

A specific methodology: The development of methods and tools which assist engineers since the very first design steps of reactive distributive software. The main difficulty is the adequacy of the proposed methods with standard design methods based on components and model engineering, which most often rely on heterogeneous formalisms and require correct-by-construction component assembly.

A set of scientific and technological foundations: Those models and methods which encompass (i) the distributed nature of the systems being considered, (ii) true concurrency, and (iii) real-time.

The contribution of the S4 Project-Team consists of algorithms and tools producing distributed reactive software from partial heterogeneous specifications of the system to be synthesized (functionality, control, architecture, quantitative performances). This means that several heterogeneous specifications (for instance, sequence diagrams and state machines) can be combined, analyzed (are the specifications consistent?) and mapped to lower-level specifications (for instance, communicating automata, or Petri nets).

The scientific approach of Team S4 begins with a rigorous modeling of problems and the development of sound theoretical foundations. This not only allows to prove the correctness (functionality and control) of the proposed transformations or analysis; but this can also guarantee the optimality of the quantitative performances of the systems produced with our methods (communication cost, response time).

Synthesis and verification methods are best studied within fundamental models, such as automata, Petri nets, event structures, synchronous transition systems. Then, results can be adapted to more realistic but complex formalisms, such as the UML. The research work of Team S4 is divided in four main tracks:

Petri net synthesis: Petri nets and apparented concurrency models have found applications in several fields, from telecommunications, to embedded systems, to process mining. Team S4 has been on the forefront of Petri net theory since its inception. In particular, the team has been very active on the theory of regions and Petri net synthesis algorithms, which has found applications in asynchronous hardware circuit design, communicating supervisory control synthesis and process mining applications.

Heterogeneous systems: This track contributes to the extension of the well-established synchronous paradigm to distributed systems. The aim is to provide a unified framework in which both synchronous systems, and particular asynchronous systems (so-called weakly-synchronous systems) can be expressed, combined, analyzed and transformed.

Reactive components: The design of reusable components calls for rich specification formalisms, with which the interactions of a component with its environment combines expectations with guarantees on its environment.We are investigating questions related to reactive component refinement and composition. We are also investigating the issues of coherence of views and modularity in complex specifications.

Discrete event system synthesis and supervisory control: Many synthesis and supervisory control problems can be expressed with full generality in the *quantified mu-calculus*, including the existence of optimal solutions to such problems. Algorithms computing winning strategies in parity games (associated with formulas in this logic) provide effective methods for solving such control problems. This framework offers means of classifying control problems, according to their decidability or undecidability, but also according to their algorithmic complexity.

3. Scientific Foundations

3.1. Scientific Foundations

The research work of the team is built on top of solid foundations, mainly, algebraic, combinatorial or logical theories of transition systems. These theories cover several sorts of systems which have been studied during the last thirty years: sequential, concurrent, synchronous or asynchronous. They aim at modeling the behavior of finite or infinite systems (usually by abstracting computations on data), with a particular focus on the control flow which rules state changes in these systems. Systems can be autonomous or reactive, that is, embedded in an environment with which the system interacts, both receiving an input flow, and emitting an output flow of events and data. System specifications can be explicit (for instance, when the system is specified by an automaton, extensively defined by a set of states and a set of transitions), or implicit (symbolic transition rules, usually parameterized by state or control variables; partially-synchronized products of finite transition systems; Petri nets; systems of equations constraining the transitions of synchronous reactive systems, according to their input flows; etc.). Specifications can be non-ambiguous, meaning that they fully define at most one system (this holds in the previous cases), or they can be ambiguous, in which case more than one system is conforming to the specification (for instance, when the system is described by logical formulas in the modal mu-calculus, or when the system is described by a set of scenario diagrams, such as *Sequence Diagrams* or *Message Sequence Charts*).

Systems can be described in two ways: either the state structure is described, or only the behavior is described. Both descriptions are often possible (this is the case for formal languages, automata, products of automata, or Petri nets), and moving from one representation to the other is achieved by folding/unfolding operations.

Another taxonomy criteria is the concurrency these models can encompass. Automata usually describe sequential systems. Concurrency in synchronous systems is usually not considered. In contrast, Petri nets or partially-synchronized products of automata are concurrent. When these models are transformed, concurrency can be either preserved, reflected or even, infused. An interesting case is whenever the target architecture requires distributing events among several processes. There, communication-efficient implementations require that concurrency is preserved as far as possible and that, at the same time, causality relations are also preserved. These notions of causality and independence are best studied in models such as concurrent automata, Petri nets or Mazurkiewicz trace languages.

Here are our sources of inspiration regarding formal mathematical tools:

1. Jan van Leeuwen (ed.), *Handbook of Theoretical Computer Science - Volume B: Formal Models and Semantics*, Elsevier, 1990.
2. Jörg Desel, Wolfgang Reisig and Grzegorz Rozenberg (eds.), *Lectures on Concurrency and Petri nets*, Lecture Notes in Computer Science, Vol. 3098, Springer, 2004.
3. Volker Diekert and Grzegorz Rozenberg (eds.), *The Book of Traces*, World Scientific, 1995.
4. André Arnold and Damian Niwinski, *Rudiments of Mu-Calculus*, North-Holland, 2001.
5. Gérard Berry, *Synchronous languages for hardware and software reactive systems - Hardware Description Languages and their Applications*, Chapman and Hall, 1997.

Our research exploits decidability or undecidability results on these models (for instance, inclusion of regular languages, bisimilarity on automata, reachability on Petri nets, validity of a formula in the mu-calculus, etc.) and also, representation theorems which provide effective translations from one model to another. For instance, Zielonka's theorem yields an algorithm which maps regular trace languages to partially-synchronized products of finite automata. Another example is the theory of regions, which provides methods for mapping finite or infinite automata, languages, or even *High-Level Message Sequence Charts* to Petri nets. A further example concerns the mu-calculus, in which algorithms computing winning strategies for parity games can be used to synthesize supervisory control of discrete event systems.

Our research aims at providing effective representation theorems, with a particular emphasis on algorithms and tools which, given an instance of one model, synthesize an instance of another model. In particular we have contributed a theory, several algorithms and a tool for synthesizing Petri nets from finite or infinite automata, regular languages, or languages of *High-Level Message Sequence Charts*. This also applies to our work on supervisory control of discrete event systems. In this framework, the problem is to compute a system (the controller) such that its partially-synchronized product with a given system (the plant) satisfies a given behavioral property (control objective, such as a regular language or satisfaction of a mu-calculus formula).

Software engineers often face problems similar to *service adaptation* or *component interfacing*, which in turn, often reduce to particular instances of system synthesis or supervisory control problems.

4. Application Domains

4.1. Modular design of embedded systems with interface theories

In 2006, with the oportunity of the SPEEDS European project on embedded system design, we decided to open a new research track on contract-/interface-based design. Our objective was to provide theory, methods and tools to support the design of embedded software in transport system industries. According to our understanding of industrial needs, gained during the SPEEDS European project, the following requirements apply to the notions of *contract* and *interface* and have been used to guide our research on this topic:

- Complex embedded and reactive systems are generally developed under a multi-layered OEM-supplier chain. Hence, a contract-based methodology should offer provision for formalizing the technical part of contractual relations. This should be achieved by formalizing, for a considered subsystem: 1/ its context of use (*assumptions*), and 2/ what is expected from the subsystem (*guarantees*). Assumptions and guarantees can be specified separately, or in a single automata-theoretic structure called interface.
- When developed under a contract-/interface-based methodology, subsystems or components should be designable in isolation, by including the needed information regarding possible future contexts of use. Subsystems or components should be substitutable to their specifications, meaning that their integration should raise no problem.
- Large systems are concurrently developed for their different *aspects* or *viewpoints* by different teams using different frameworks and tools. Examples of such aspects include the functional, reliability, timing, memory and power aspects. Each of these aspects requires specific frameworks and tools for their analysis and design. Yet, they are not totally independent but rather interact. The issue of dealing with multiple aspects or multiple viewpoints is thus essential. This implies that several contracts or interfaces are associated with a same system, sub-system, or component, namely at least one per viewpoint. These contracts/interfaces are to be interpreted in a conjunctive way and modular reasoning methods have to be developed to support large sets of contracts.
- The need for supporting conjunctive contracts/interfaces also follows from the current practice in which early requirement capture results in many elementary requirements. These requirements typically consist of English text, semi-formal languages whose sentences are translatable into predefined behavioral patterns, or even graphical scenario languages.
- It is highly desirable that designing by contracts and interfaces has the mildest possible impact on the design process, a key proprietory asset to all major companies.

4.2. Opacity, Supervision, and Petri Nets

Our activities on components emerged from a larger basis of competences developed in the past of S4 on supervisory control and Petri net synthesis. Components and their interfaces are intimately tied to supervisory control, and Petri net synthesis is a possible approach to controller synthesis. In the last four years, we have carried on work on both themes, but refocussed our research on fresh topics. A major contribution has been to study supervisory control for secrecy objectives, with promising results. Another contribution has been to study supervisory control for finite abstractions of services. The fusion of both topics, that would increase the interest of the results for Web applications, is not yet done. A different topic that we continued to investigate is the synthesis of distributed controllers based on the synthesis of distributed Petri nets. Our progress on this difficult topic is limited, but we feel we should pursue the effort.

Opacity is an abstract property that includes non-interference and that can cover confidentiality, authenticity and many other specific security concepts. Our project-team has inaugurated research on supervisory control of discrete event systems for opacity, which became soon a theme of cooperation with project-team Vertecs and subsequently attracted concurrent researches at Wayne State U., Kyoto Inst. of Tech., and U. Illinois. We have some advance over these concurrent teams.

The rest of our work on supervision focusses on minimizing communication between decentralized controllers, on asynchronous and distributed control, and on the enforcement of modal specifications. Decreasing communication between decentralized controllers was studied at Michigan U. but we could further show that minimizing communication reduces to a classical optimization problem. As regards asynchronously communicating control, the only current attempts we are aware of are those of project-teams S4 and Vertecs. As regards supervisory control w.r.t. modal specifications, the closest work is Lohmann and Wolf's synthesis of communication partners for Web services.

The approach which we propose towards distributed control relies upon the synthesis of distributed Petri nets. We have been leaders for fifteen years on the synthesis of P/T nets, on a par with the Petrify team focussed on Elementary (or safe) net synthesis. The algorithms which we have defined have been reused or adapted by many other researchers in Europe, in the US, and in China, to respond to three types of problems: controller synthesis, process mining, and concise representation of services. We are currently writing a book covering all aspects of the theory and applications of Petri net synthesis. We also pursue research on structure theory of Petri nets, in cooperation with U. Oldenburg, with focus set recently on non-interference.

4.3. Hybrid Systems Modelers

This is an opportunistic objective, not part of the plans stated when the team was formed. It results from a series of events: in 2008, Benoît Caillaud was part of the *Synchronics* large scale initiative (see section 7.1.1), dedicated to "embedded systems programming in 2020". Hybrid Systems Modelers were part of the research program. Such tools are nowadays absolutely central in the development of Cyber Physical Systems (CPS), which are physical systems in closed loop with embedded control. Hybrid Systems Modelers support the modeling of physical systems (with Ordinary Differential Equations, ODE, and Differential Algebraic Equations, DAE): Matlab-Simulink and Modelica are the main players. Our vision was that these tools should deserve similar effort in theory as synchronous languages did for the programming of embedded systems. About one year after Synchronics started (focusing mostly on other topics), the PhD thesis of Simon Bliudze came to our knowledge. This thesis contained a long chapter on the use of *non-standard analysis* as a semantic framework for hybrid systems. The exposure relied on a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström. That attracted the attention of Albert Benveniste, so he joined the group of Synchronics working on hybrid systems. This was the beginning of a deeply novel and exciting research track.

The computer science community has devoted significant efforts to the analysis and verification of hybrid automata. The framework of hybrid automata is, however, much less flexible than what actual Hybrid Systems Modelers offer. The only ongoing effort towards modeling has been developed by Edward Lee and his team as part of the Ptolemy II project. This has led to the proposal of *super-dense time semantics,* in which cascades of successive instants can occur in zero time by using $R_+ \times N$ as a time index. It turns out that the set $T = \{n\partial \mid n \in N^*\}$, where ∂ is an *infinitesimal* and N^* is the set of *non-standard integers* is such that 1/ T is dense in R_+, making it "continuous", and 2/ every $t \in T$ has a predecessor in T and a successor in T, making it "discrete" (le beurre et l'argent du beurre, as we say in french). Although non-effective from the operational point of view, the *non-standard semantics* of hybrid systems provides a framework that is very familiar to the computer scientist (who is afraid of continuous time) and at the same time efficient as a symbolic abstraction. This makes it an excellent candidate for the development of compilation schemes.

5. Software

5.1. Mica: A Modal Interface Compositional Analysis Toolbox

Participant: Benoît Caillaud.

http://www.irisa.fr/s4/tools/mica/

Mica is an Ocaml library developed by Benoît Caillaud implementing the Modal Interface algebra published in [8]. The purpose of Modal Interfaces is to provide a formal support to contract based design methods in the field of system engineering. Modal Interfaces enable compositional reasoning methods on I/O reactive systems.

In Mica, systems and interfaces are represented by extension. However, a careful design of the state and event heap enables the definition, composition and analysis of reasonably large systems and interfaces. The heap stores states and events in a hash table and ensures structural equality (there is no duplication). Therefore complex data-structures for states and events induce a very low overhead, as checking equality is done in constant time.

Thanks to the Inter module and the mica interactive environment, users can define complex systems and interfaces using Ocaml syntax. It is even possible to define parameterized components as Ocaml functions.

Mica is available as an open-source distribution, under the CeCILL-C Free Software License Agreement (http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html).

5.2. Synet: A General Petri-Net Sythesis Toolbox

Participant: Benoît Caillaud.

http://www.irisa.fr/s4/tools/synet/

Synet is a software tool for the synthesis of bounded and unbounded Petri-nets, based on the theory of regions [33]. It can synthesize Petri-nets from automata or regular expressions and can be configured by command-line options to synthesize nets modulo graph isomorphism or language equality. Petri nets computed by Synet can be displayed using the GraphViz 2D graph layout software, or saved to a file for further transformation and analysis.

The tool actually implements two linear-algebraic synthesis methods: a first method uses the simplex algorithm and the second one is based on the computation of extremal rays of polyhedral cones, using Chernikova's algorithm [35]. Both methods imply that the input graphs are given by extension. Nevertheless, Synet yields good performances on many practical use-cases and is the only tool supporting unbounded net synthesis.

The main application of Synet is the synthesis of communicating distributed protocols and controllers [32]. Synthesis is constrained to produce so-called distributables nets [34], a class of nets that can be turned into networks of communicating automata by automated methods. This allows to divide the synthesis problem in two steps: Given the specification of a protocol as a finite automaton, (i) synthesize (if it exists) a distributable net, and then (ii) derive a network of communicating automata from the distributable net. While the second step is automatic and straightforward, the first step is in essence a computer assisted design task, where the distributed Petri-net synthesis algorithm helps the designer to refine the protocol specification into a graph isomorphic to the marking graph of a distributable net.

6. New Results

6.1. Petri Nets and their Synthesis

Participants: Eric Badouel, Philippe Darondeau.

6.1.1. Deciding Selective Declassification of Petri Nets

In [20], we consider declassification, as effected by downgrading actions D, in the context of intransitive non-interference encountered in systems that consist of high-level (secret) actions H and low-level (public) actions L. In a previous work, we had shown the decidability of a strong form of declassification, by which D contains only a single action d declassifying all H actions at once. We continue this study by considering selective declassification, where each transition d in D can declassify a subset $H(d)$ of H. The decidability of this more flexible, application-prone declassification framework is proved in the context of (possibly unbounded) Petri nets with possibly infinite state spaces.

6.1.2. Petri Net Distributability

A Petri net is distributed if, given an allocation of transitions to (geographical) locations, no two transitions at different locations share a common input place. A system is distributable if there is some distributed Petri net implementing it. We address in [21] the question of which systems can be distributed, while respecting a given allocation. We state the problem formally and discuss several examples illuminating — to the best of our knowledge — the current status of this work.

6.1.3. Petri Net Reachability Graphs: Decidability Status of First Order Prioperties

We investigated in [13] the decidability and complexity status of model-checking problems on unlabelled reachability graphs of Petri nets by considering first-order, modal and pattern-based languages without labels on transitions or atomic propositions on markings. We have considered several parameters to separate decidable problems from undecidable ones. Not only were we able to provide precise borders and a systematic analysis, but we also demonstrated the robustness of our proof techniques.

6.1.4. α-reconstructibility of Workflow Nets

The α-algorithm is a process mining algorithm, introduced by van der Aalst et al, that constructs a workflow net from an event log. A class of nets, the structured workflow nets, was recognized to be reconstructible by algorithm α from their language (or a representative subset of it). In [14] we assessed more precisely the α-algorithm we provided a characterization of the class of the workflow nets that are discovered by α.

6.2. Hybrid Modeling

Participants: Albert Benveniste, Benoît Caillaud.

Hybrid system modelers have become a corner stone of complex embedded system development. Embedded systems include not only control components and software, but also physical devices. In this area, Simulink is a de facto standard design framework, and Modelica a new player. However, such tools raise several issues related to the lack of reproducibility of simulations (sensitivity to simulation parameters and to the choice of a simulation engine). In [10] we propose using techniques from non-standard analysis to define a semantic domain for hybrid systems. Non-standard analysis is an extension of classical analysis in which infinitesimal (the ϵ and η in the celebrated generic sentence $\forall\epsilon\exists\eta$... of college maths) can be manipulated as first class citizens. This approach allows us to define both a denotational semantics, a constructive semantics, and a Kahn Process Network semantics for hybrid systems, thus establishing simulation engines on a sound but flexible mathematical foundation. These semantics offer a clear distinction between the concerns of the numerical analyst (solving differential equations) and those of the computer scientist (generating execution schemes). We also discuss a number of practical and fundamental issues in hybrid system modelers that give rise to non-reproducibility of results, non-determinism, and undesirable side effects. Of particular importance are cascaded mode changes (also called "zero-crossings" in the context of hybrid systems modelers). This work has taken place in the framework of the Synchronics large scale initiative (see section 7.1.1).

6.3. Component-Based Design

Participants: Albert Benveniste, Benoît Caillaud, Sophie Pinchinat.

6.3.1. Application of Interface Theories to the Separate Compilation of Synchronous Programs

We study in [15], [26] the problem of separate compilation, i.e., the generation of modular code, for the discrete time part of block-diagrams formalisms such as Simulink, Modelica, or Scade. Code is modular in that it is generated for a given composite block independently from context (i.e., without knowing in which diagrams the block is to be used) and using minimal information about the internals of the block. Just using off-the-shelf C code generation (e.g., as available in Simulink) does not provide modular code. Separate compilation was solved by Lublinerman et al. for the special case of single clocked diagrams, in which all signals are updated at a same unique clock. For the same case, Pouzet and Raymond proposed algorithms that scale-up properly to real-size applications. The technique of Lublinerman et al. was extended to some classes of multi-clocked and timed diagrams. We study this problem in its full generality and we show that it can be cast to a special class of controller synthesis problems by relying on recently proposed modal interface theories.

6.3.2. Contracts for System Design

Systems design has become a key challenge and differentiating factor over the last decades for system companies. Aircrafts, trains, cars, plants, distributed telecommunication military or health care systems, and more, involve systems design as a critical step. Complexity has caused system design times and costs to go severely over budget so as to threaten the health of entire industrial sectors. Heuristic methods and standard practices do not seem to scale with complexity so that novel design methods and tools based on a strong theoretical foundation are sorely needed. Model-based design as well as other methodologies such as layered and compositional design have been used recently but a unified intellectual framework with a complete design flow supported by formal tools is still lacking albeit some attempts at this framework such as Platform-based Design have been successfully deployed. Recently an “orthogonal” approach has been proposed that can be applied to all methodologies proposed thus far to provide a rigorous scaffolding for verification, analysis and abstraction/refinement: contract-based design. Several results have been obtained in this domain but a unified treatment of the topic that can help in putting contract-based design in perspective is still missing. In [25], we intend to provide such treatment where contracts are precisely defined and characterized so that they can be used in design methodologies such as the ones mentioned above with no ambiguity. In addition, the paper provides an important link between interfaces and contracts to show similarities and correspondences. Examples of the use of contracts in system design are provided, including one based on Modal Interfaces, using the Mica tool (see section 5.1).

6.3.3. Ensuring Reachability by Design

In [18], [28], we study the independent implementability of reachability properties, which are in general not compositional. We consider modal specifications, which are widely acknowledged as suitable for abstracting implementation details of components while exposing to the environment relevant information about cross-component interactions. In order to obtain the required expressivity, we extend them with marked states to model states to be reached. We then develop an algebra with both logical and structural composition operators ensuring reachability properties by construction.

6.3.4. Modal event-clock specifications for timed component-based design

Modal specifications are classic, convenient, and expressive mathematical objects to represent interfaces of component-based systems. However, time is a crucial aspect of systems for practical applications, e.g. in the area of embedded systems. And yet, only few results exist on the design of timed component-based systems. In [11], we propose a timed extension of modal specifications, together with fundamental operations (conjunction, product, and quotient) that enable reasoning in a compositional way about timed system. The specifications are given as modal event-clock automata, where clock resets are easy to handle. We develop an entire theory that promotes efficient incremental design techniques.

6.4. Automata, Games and Logics for Supervisory Control and System Synthesis

Participants: Philippe Darondeau, Bastien Maubert, Sophie Pinchinat.

6.4.1. Distributed Control of Discrete Event Systems: A First Step

To initiate a discussion on the modeling requirements for distributed control of discrete-event systems, a partially-automated region-based methodology is presented in [23]. The methodology is illustrated via a well-known example from distributed computing: the dining philosophers.

6.4.2. Enforcing Opacity of Regular Predicates on Modal Transition Systems

In [22] we considered the following problem: Given a labelled transition system LTS partially observed by an attacker, and a regular predicate Sec over the runs of LTS, enforcing opacity of the secret Sec in LTS means computing a supervisory controller K such that an attacker who observes a run of K/LTS cannot ascertain that the trace of this run belongs to Sec based on the knowledge of LTS and K. We then lifted the problem from a single labelled transition system LTS to the class of all labelled transition systems specified by a modal transition system MTS. The lifted problem is to compute the maximally permissive controller K such that Sec is opaque in K/LTS for every labelled transition system LTS which is a model of MTS. The situations of the attacker and of the controller are dissymmetric: at run time, the attacker may fully know LTS and K whereas the controller knows only MTS and the sequence of actions executed so far by the unknown LTS. We addressed the problem in two cases. Let Σ_a denote the set of actions that can be observed by the attacker, and let Σ_c and Σ_o denote the sets of actions that can be controlled and observed by the controller, respectively. We provided optimal and regular controllers that enforce the opacity of regular secrets when $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_a = \Sigma$. We also provided optimal and regular controllers that enforce the opacity of regular upper-closed secrets ($Sec = Sec.\Sigma^*$) when $\Sigma_a \subseteq \Sigma_c \subseteq \Sigma_o = \Sigma$.

6.4.3. Analysis of partially observed recursive tile systems

The analysis of discrete event systems under partial observation is an important topic, with major applications such as the detection of information flow and the diagnosis of faulty behaviors. In [19], we consider recursive tile systems, which are infinite systems generated by a finite collection of finite *tiles*, a simplified variant of deterministic graph grammars. Recursive tile systems are expressive enough to capture classical models of recursive systems, such as the pushdown systems and the recursive state machines. They are infinite-state in general and therefore standard powerset constructions for monitoring do not always apply. We exhibit computable conditions on recursive tile systems and present non-trivial constructions that yield effective computation of the monitors. We apply these results to the classic problems of opacity and diagnosability.

6.4.4. Uniform Strategies

In [29], we consider turn-based game arenas for which we investigate uniformity properties of strategies. These properties involve bundles of plays, that arise from some semantical motive. Typically, we can represent constraints on allowed strategies, such as being observation-based. We propose a formal language to specify uniformity properties and demonstrate its relevance by rephrasing various known problems from the literature. Note that the ability to correlate different plays cannot be achieved by any branching-time logic if not equipped with an additional modality, so-called R in this contribution. We also study an automated procedure to synthesize strategies subject to a uniformity property, which strictly extends exitsting results based on, say standard temporal logics. We exhibit a generic solution for the synthesis problem provided the bundles of plays rely on any binary relation definable by a finite state transducer. This solution yields a non-elementary procedure.

6.4.5. Emptiness Of Alternating Parity Tree Automata Using Games With Imperfect Information

In [30], we focus on the emptiness problem for alternating parity tree automata. The usual technique to tackle this problem first removes alternation, going to non-determinism, and then checks emptiness by reduction to a two-player perfect-information parity game. In this note, we give an alternative roadmap to this problem by providing a direct reduction to the emptiness problem to solving an imperfect-information two-player parity game.

6.4.6. *On timed alternating simulation for concurrent timed games*

We address in [12] the problem of alternating simulation refinement for concurrent timed games (*TG*). We show that checking timed alternating simulation between *TG* is EXPTIME-complete, and provide a logical characterization of this preorder in terms of a meaningful fragment of a new logic, *TAMTL**. *TAMTL** is an action-based timed extension of standard alternating-time temporal logic *ATL**, which allows to quantify over strategies where the designated coalition of players is not responsible for blocking time. While for full *TAMTL**, model-checking *TG* is undecidable, we show that for its fragment *TAMTL*, corresponding to the timed version of *ATL*, the problem is instead decidable and in EXPTIME.

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. *Synchronics: Language Platform for Embedded System Design*

Participants: Albert Benveniste, Benoît Caillaud.

Large scale initiative funded by INRIA. *http://synchronics.inria.fr/*

This project, started Jan 1st 2008, is supported by INRIA. It capitalizes on recent extensions of data-flow synchronous languages (mode automata, Lucid Synchrone, Signal, Lustre, ReactiveML, relaxed forms of synchronous composition or compilation techniques for various platforms). We aim to address the main challenges of embedded system design, starting from a single, semantically well founded programming language.

Our contributions to Synchronics in 2012 are:

- A journal paper [10] presenting the non-standard semantics for hybrid systems and its applications to the semantics and compilation of hybrid modeling languages. Details can be found in Section 6.2.
- Inputs to the latest evolution of the Modelica language, related to state machines and a clock calculus.
- A study of modular code generation techniques for reactive synchronous programming languages, based on an interface theoretic approach [15], [26]. See 6.3 for further details.

7.2. European Initiatives

7.2.1. *Collaborations in European Programs, except FP7*

Program: ITEA 2

Project acronym: MODRIO

Project title: Model driven Physical Systems Operation

Duration: Sep 2012 - Aug 2015

Coordinator: EDF (France)

Other partners:ABB (Sweden and Germany), AIT (Austria), Ampère - INSA Lyon and CNRS (France), Bielefeld university (Germany), Dassault Aviation (France), DLR (Germany), DPS (France), Dassault Systèmes (France), EADS (France), Enicon (Austria), Equa Simulation (Sweden), IFPEN (France), Ilmenau university (Germany), ITI (Germany), KUL (Belgium), Knorr-Bremse (Germany), Linköping university (Sweden), LMS Imagine (France and Belgium), MathCore Engineering (Sweden), Modelon AB (Sweden), Pöyry Finland Oy (Finland), QTronic (Germany), Scania (Sweden), Semantum Oy (Finland), Sherpa Engineering (France), Siemens AG (Germany), Siemens Industrial Turbomachinery AB (Sweden), Simpack AG (Germany), Supméca (France), Triphase (Belgium), University of Calabria (Italy), Vattenfall (Sweden), VTT (Finland), Wapice Ltd. (Finland).

Abstract: MODRIO seeks solutions to support adoption of model-based systems engineering in the design of mechatronic systems. The project covers all phases of the development cycle - from early concept design, over detailed system design, to verification and validation - and operational use including diagnostics during the entire system's life cycle.

7.3. International Initiatives

7.3.1. Participation In International Programs

Eric Badouel is contributing to the ALOCO research project of the LIRIMA, on component-based software architectures (http://www.lirima.uninet.cm/index.php/component/content/article?id=2).

7.4. International Research Visitors

7.4.1. Internships

Hela GOMRI (from Mar 2012 until Sep 2012)

Subject: Systèmes collaboratifs à l'aide de documents actifs.

Institution: Ecole Nationale d'Ingénieurs de Tunis (Tunisia)

8. Dissemination

8.1. Scientific Animation

Eric Badouel is the secretary of the steering committee of CARI, the African Conference on Research on Computer Science and Applied Mathematics. He took part in the programme committee and in the organizing committee of CARI 2012. He is a member of the editorial board of the ARIMA Journal.

Albert Benveniste is associated editor at large (AEAL) for the journal *IEEE Trans. on Automatic Control*. He is member of the Strategic Advisory Council of the Institute for Systems Research, Univ. of Maryland, College Park, USA. He belongs to the Scientific Advisory Board of INRIA, where he is in charge of the area of Embedded Systems. [1]

Benoît Caillaud has served in the steering and program committees of the International Conference on Application of Concurrency to System Design (ACSD 2012). He is serving on the Evaluation Committee of INRIA.

Philippe Darondeau has served on the program committees of the 23rd International Conference on Concurrency Theory (CONCUR 2012) and 11th International Workshop on Discrete Event Systems (WODES 2012). He is a member of the IFIP Working Group WG2.2 and has served as secretery of this working group until September 2012.

Sophie Pinchinat is serving in the editorial board of the Journal on Discrete Event Dynamic Systems, Theory and Applications. She has served on the program committee of the 10th Conference on Logic and the Foundations of Game and Decision Theory (LOFT 2012).

[1] Only facts related to the activities of Team S4 are mentioned. Other roles or duties concern the DistribCom team, to which A. Benveniste also belongs.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

- Benoit Caillaud has contributed to the training programme for the computer-science option of the *agregation* in mathematics, at ENS Cachan-Ker Lann.
- Sophie Pinchinat has contributed to three courses on verification and game theory in the computer science master degree of the University of Rennes 1 and at Supelec. She has taught algorithmics at undergraduate level at the university of Rennes 1. She has taught embedded systems design in the engineering degree of the university of Rennes 1.

8.2.2. Supervision

Several PhDs are in progress:

- Lamine Diouf, *Artefact opacity in workflow systems*, started in 2010, supervised by Eric Badouel.
- Bastien Maubert, *Logical Foundations of Imperfect Information Games*, started September 2010, co-supervised by Sophie Pinchinat and Guillaume Aucher.
- Stéphanie Georges, *Formal approaches to physical security analysis: Attack tree synthesis*, started December 2011, co-supervised by Sophie Pinchinat and Achour Mostefaoui.

8.2.3. Juries

Sophie Pinchinat has served on the PhD thesis jury of Loïg Jézeéquel (project-team DISTRIBCOM).

9. Bibliography

Major publications by the team in recent years

[1] E. BADOUEL, M. BEDNARCZYK, A. BORZYSZKOWSKI, B. CAILLAUD, P. DARONDEAU. *Concurrent Secrets*, in "Discrete Event Dynamic Systems", December 2007, vol. 17, n^{o} 4, p. 425-446, http://dx.doi.org/10.1007/s10626-007-0020-5.

[2] E. BADOUEL, B. CAILLAUD, P. DARONDEAU. *Distributing Finite Automata through Petri Net Synthesis*, in "Journal on Formal Aspects of Computing", 2002, vol. 13, p. 447–470, http://dx.doi.org/10.1007/s001650200022.

[3] E. BADOUEL, P. DARONDEAU. *Theory of regions*, in "Lectures on Petri Nets I: Basic Models", Lecture Notes in Computer Science, Springer, 1999, vol. 1491, p. 529–586.

[4] A. BENVENISTE, B. CAILLAUD, L. CARLONI, P. CASPI, A. SANGIOVANNI-VINCENTELLI. *Composing heterogeneous reactive systems*, in "ACM Trans. Embedded Comput. Syst.", 2008, vol. 7, n^{o} 4, http://doi.acm.org/10.1145/1376804.1376811.

[5] A. BENVENISTE, B. CAILLAUD, P. LE GUERNIC. *Compositionality in dataflow synchronous languages: specification and distributed code generation*, in "Information and Computation", 2000, vol. 163, p. 125-171.

[6] G. FEUILLADE, S. PINCHINAT. *Modal Specifications for the Control Theory of Discrete-Event Systems*, in "Discrete Event Dynamic Systems", 2007, vol. 17, n^{o} 2, p. 211–232, http://dx.doi.org/10.1007/s10626-006-0008-6.

[7] D. POTOP-BUTUCARU, B. CAILLAUD. *Correct-by-Construction Asynchronous Implementation of Modular Synchronous Specifications*, in "Fundamenta Informaticae", 2007, vol. 78, n^{o} 1, p. 131–159.

[8] J.-B. RACLET, E. BADOUEL, A. BENVENISTE, B. CAILLAUD, A. LEGAY, R. PASSERONE. *A Modal Interface Theory for Component-based Design*, in "Fundamenta Informaticae", 2011, vol. 108, n^{o} 1-2, p. 119-149, http://dx.doi.org/10.3233/FI-2011-416.

[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, Springer, aug 2003, vol. 2747, p. 642–651, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2747&spage=642.

Publications of the year

Articles in International Peer-Reviewed Journals

[10] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *Non-standard semantics of hybrid systems modelers*, in "Journal of Computer and System Sciences", 2012, vol. 78, n^{o} 3, p. 877-910, This work was supported by the SYNCHRONICS large scale initiative of Inria [*DOI :* 10.1016/J.JCSS.2011.08.009], http://hal.inria.fr/hal-00766726.

[11] N. BERTRAND, A. LEGAY, S. PINCHINAT, J.-B. RACLET. *Modal event-clock specifications for timed component-based design*, in "Science of Computer Programming", 2012, n^{o} 77, p. 1212-1234 [*DOI :* 10.1016/J.SCICO.2011.01.007], http://hal.inria.fr/hal-00752449.

[12] L. BOZZELLI, A. LEGAY, S. PINCHINAT. *On timed alternating simulation for concurrent timed games*, in "Acta Informatica", 2012, vol. 49, n^{o} 4, p. 249-279, http://dx.doi.org/10.1007/s00236-012-0158-y.

[13] P. DARONDEAU, S. DEMRI, R. MEYER, C. MORVAN. *Petri Net Reachability Graphs: Decidability Status of First Order Properties*, in "Logical Methods in Computer Science", October 2012, vol. 8, n^{o} 4:9, p. 1-28 [*DOI :* 10.2168/LMCS-8(4:9)2012], http://hal.inria.fr/hal-00743935.

International Conferences with Proceedings

[14] E. BADOUEL. *On the alpha-Reconstructibility of Workflow Nets*, in "Application and Theory of Petri Nets - 33rd International Conference, PETRI NETS 2012", Hamburg, Allemagne, S. HADDAD, L. POMELLO (editors), Lecture Notes in Computer Science, Springer, June 25-29 2012, vol. 7347, p. 128-147 [*DOI :* 10.1007/978-3-642-31131-4], http://hal.inria.fr/hal-00748230.

[15] A. BENVENISTE, B. CAILLAUD, J.-B. RACLET. *Application of Interface Theories to the Separate Compilation of Synchronous Programs*, in "51st IEEE Conference on Decision and Control", Maui, Hawaii, United States, M. E. VALCHER (editor), Jay A. Farrell, December 2012, http://hal.inria.fr/hal-00766793.

[16] L. BOZZELLI, S. PINCHINAT. *Verification of Gap-Order Constraint Abstractions of Counter Systems*, in "Verification, Model Checking, and Abstract Interpretation - 13th International Conference, VMCAI 2012", Philadelphia, PA, USA, V. KUNCAK, A. RYBALCHENKO (editors), Lecture Notes in Computer Science, Springer, January 22-24 2012, vol. 7148, p. 88-103, http://dx.doi.org/10.1007/978-3-642-27940-9_7.

[17] L. BOZZELLI, H. P. VAN DITMARSCH, S. PINCHINAT. *The Complexity of One-Agent Refinement Modal Logic*, in "Logics in Artificial Intelligence - 13th European Conference, JELIA 2012", Toulouse, France, L.

F. DEL CERRO, A. HERZIG, J. MENGIN (editors), Lecture Notes in Computer Science, Springer, September 26-28 2012, vol. 7519, p. 120-133, http://dx.doi.org/10.1007/978-3-642-33353-8_10.

[18] B. CAILLAUD, J.-B. RACLET. *Ensuring Reachability by Design*, in "Theoretical Aspects of Computing - ICTAC 2012 - 9th International Colloquium", Bangalore, India, A. ROYCHOUDHURY, M. D'SOUZA (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7521, p. 213-227 [*DOI :* 10.1007/978-3-642-32943-2_17], http://hal.inria.fr/hal-00766735.

[19] S. CHÉDOR, C. MORVAN, S. PINCHINAT, H. MARCHAND. *Analysis of partially observed recursive tile systems*, in "11th Int. Workshop on Discrete Event Systems", Guadalajara, Mexico, October 2012, p. 265-271, http://hal.inria.fr/hal-00743196.

[20] P. DARONDEAU, E. BEST. *Deciding Selective Declassification of Petri Nets*, in "Principles of Security and Trust (POST)", Tallinn, Estonia, P. DEGANO, J. D. GUTTMAN (editors), LNCS, Springer, 2012, vol. 7215, p. 290-308 [*DOI :* 10.1007/978-3-642-28641-4], http://hal.inria.fr/hal-00745298.

[21] P. DARONDEAU, E. BEST. *Petri Net Distributability*, in "Perspectives of Systems Informatics - 8th International Andrei Ershov Memorial Conference, PSI 2011", Novosibirsk, Russian Federation, E. M. CLARKE, I. VIRBITSKAITE, A. VORONKOV (editors), LNCS, Springer, 2012, vol. 7162, p. 1-18 [*DOI :* 10.1007/978-3-642-29709-0_1], http://hal.inria.fr/hal-00745311.

[22] P. DARONDEAU. *Enforcing Opacity of Regular Predicates on Modal Transition Systems*, in "11th Int. Workshop on Discrete Event Systems", Guadalajara, Mexico, October 2012, p. 331-336.

Scientific Books (or Scientific Book chapters)

[23] P. DARONDEAU, L. RICKER. *Distributed Control of Discrete Event Systems: A First Step*, in "Transactions on Petri Nets and Other Models of Concurrency", 2012, vol. 6, p. 24–45, http://dx.doi.org/10.1007/978-3-642-35179-2_2.

Research Reports

[24] G. AUCHER, B. MAUBERT, F. SCHWARZENTRUBER. *Generalized DEL-sequents*, Inria, July 2012, n^{o} RR-8012, 23, http://hal.inria.fr/hal-00716074.

[25] A. BENVENISTE, B. CAILLAUD, D. NICKOVIC, R. PASSERONE, J.-B. RACLET, P. REINKEMEIER, A. SANGIOVANNI-VINCENTELLI, W. DAMM, T. HENZINGER, K. G. LARSEN. *Contracts for System Design*, Inria, November 2012, n^{o} RR-8147, 65, http://hal.inria.fr/hal-00757488.

[26] A. BENVENISTE, B. CAILLAUD, J.-B. RACLET. *Application of Interface Theories to the Separate Compilation of Synchronous Programs*, Inria, July 2012, n^{o} RR-8030, http://hal.inria.fr/hal-00721049.

[27] L. BOZZELLI, H. P. VAN DITMARSCH, T. FRENCH, J. HALES, S. PINCHINAT. *Refinement Modal Logic*, abs/1202.3538, 2012, http://arxiv.org/abs/1202.3538.

[28] B. CAILLAUD, J.-B. RACLET. *Ensuring Reachability by Design*, Inria, May 2012, n^{o} RR-7928, 20, http://hal.inria.fr/hal-00696151.

[29] B. MAUBERT, S. PINCHINAT. *Uniform Strategies*, Inria, December 2012, n^{o} RR-8144, http://hal.inria.fr/hal-00760370.

[30] S. PINCHINAT, O. SERRE. *Emptiness Of Alternating Parity Tree Automata Using Games With Imperfect Information*, IRISA, May 2012, n^{o} PI-1992, 6, http://hal.inria.fr/hal-00700334.

Other Publications

[31] M. GHESMOUNE. *Anonymisation de réseaux sociaux*, Université Rennes 1, June 2012, http://dumas.ccsd.cnrs.fr/dumas-00725254.

References in notes

[32] E. BADOUEL, B. CAILLAUD, P. DARONDEAU. *Distributing Finite Automata through Petri Net Synthesis*, in "Journal on Formal Aspects of Computing", 2002, vol. 13, p. 447–470.

[33] E. BADOUEL, P. DARONDEAU. *Theory of regions*, in "Lectures on Petri Nets I: Basic Models", Lecture Notes in Computer Science, Springer, 1999, vol. 1491, p. 529–586.

[34] R. P. HOPKINS. *Distributable nets*, in "Advances in Petri Nets 1991, Papers from the 11th International Conference on Applications and Theory of Petri Nets", G. ROZENBERG (editor), Lecture Notes in Computer Science, Springer, 1991, vol. 524, p. 161–187.

[35] A. SCHRIJVER. *Theory of linear and integer programming*, Wiley, April 1998.