INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Pop Art

Programming and OPerating systems for Applications in Real-Time

Rhône-Alpes

THEME COM

Table of contents

1. Team

Project leader
Alain Girault [CR INRIA, HdR]

Project assistants
Stéphanie Berger [Secretary (SAR) INRIA]

Inria permanent researchers
Pascal Fradet [CR INRIA, HdR]
Gregor Goessler [CR INRIA]
Bertrand Jeannet [CR INRIA, since 03/2006]
Daniel Le Métayer [DR INRIA, since 06/2006, HdR]
Eric Rutten [CR INRIA, since 10/2006, HdR]
Daniel Simon [CR INRIA, until 11/2006, HdR]

PhD students
Gwenaël Delaval [MENRT grant]
David Robert [MENRT grant, until 10/2006]
Simplice Djoko Djoko [INRIA grant, AOSD NoE]
Mouaiad Alras [Region Rhône-Alpes grant, ISLE cluster, since 10/2006]
Gérald Vaisman [INRIA grant, DCN contract, since 10/2006]

Post-doctoral fellows
Massimo Tivoli [MENRT grant, until 10/2006]
Hamoudi Kalla [09/2006 to 12/2006]
Adrien Richard [MENRT grant, since 09/2006]

Interns
Abdul-Malik Khan [M2R Grenoble, until 07/2006, co-advised with the VASY team]
Rodrigo Guimaraes [Internship for International students, since 03/2006 until 07/2006]
Christophe Junke [M2P Grenoble, 04/2006 to 08/2006]

Visitor
Rémi Douence [École des Mines de Nantes, since 09/2006]

External Partners
Emil Dumitrescu [INSA Lyon]
Olivier Sename [LAG, INPG-ENSIEG, until 10/06]

2. Overall Objectives

2.1. Overall Objectives

We work on the problem of the safe design of real-time control systems. This area is related to control theory as well as computer science. Application domains are typically safety-critical systems, as in transportation (avionics, railways), production, medical or energy production systems. Both methods and formal models for the construction of correct systems, as well as their implementation in computer assisted design tools, targeted to specialists of the applications, are needed. We contribute to propose solutions all along the design flow, from the specification to the implementation: we develop techniques for the specification and automated generation of safe real-time executives for control systems, as well as static analysis techniques to check additional properties on the generated systems. Our special research themes are:

- implementations of synchronous reactive programs, generated automatically by compilation, particularly from the point of view of distribution (in relation with the LUSTRE [1] and ESTEREL [2] languages) and fault tolerance (in relation with the SYNDEX [3] environment);

[1] http://www-verimag.imag.fr/SYNCHRONE
[2] http://www.inria.fr/recherche/equipes/aoste.en.html

- control/scheduling co-design, with cross-interactions between techniques of servoing and real-time operating systems (RTOS), in order to obtain an adaptive scheduling, with regard to quality of service (in relation with the ORCCAD [4] environment);
- high-level design and programming methods, with support for automated code generation, including: the automated generation of correct controllers using discrete control synthesis (in relation with the Mode Automata [5] and SIGNAL [6] languages, and with the SIGALI synthesis tool); compositionality for the verification, and construction of correct systems; reactive programming, aspect-oriented programming.
- static analysis and abstract interpretation techniques, which are applied both to low-level synchronous models/programs and to more general imperative programs; this includes the verification of general safety properties and the absence of runtime errors.

Our applications are in embedded systems, typically in the robotics, automotive, and telecommunications domains with a special emphasis on dependability issues (*e.g.*, fault tolerance, availability). International and industrial relations feature:

- two IST European networks of excellence:
  - ARTIST II [7], about embedded real-time systems,
  - AOSD-Europe[8], about formal methods for Aspect-Oriented Programming,
- three ACIs ("Actions Concertées Incitatives"): ALIDECS (on large-scale critical embedded systems), DISPO (on security policies for software components), and APRON (numerical program analysis);
- one ARA ("Action de Recherche Amont"): SAFE_NECS on networked embedded control systems,
- the OPENTLM project of the MINALOGIC Pole of Competitiveness, dedicated to the design flow for next generation SoC and SystemC,
- industrial collaborations with DCN and POLYSPACE.

3. Scientific Foundations

3.1. Embedded systems and their safe design

Keywords: *Embedded systems*, *control*, *distribution*, *real-time*, *safety-criticality*.

3.1.1. The safe design of embedded real-time control systems.

The context of our work is the area of embedded real-time control systems, at the intersection between control theory and computer science. Our contribution consists of methods and tools for their safe design. The systems we consider are intrinsically safety-critical because of the interaction between the embedded, computerized controller, and a physical process having its own dynamics. What is important is to analyze and design the safe behavior of the whole system, which introduces an inherent complexity. This is even more crucial in the case of systems whose malfunction can have catastrophic consequences, for example in transport systems (avionics, trains), production, medical, or energy production systems.

[3] http://www-rocq.inria.fr/syndex
[4] http://sed.inrialpes.fr/Orccad
[5] http://www-verimag.imag.fr/PEOPLE/Florence.Maraninchi/MATOU
[6] http://www.irisa.fr/espresso
[7] http://www.artist-embedded.org/FP6/Overview/
[8] http://www.aosd-europe.net/

Therefore, there is a need for methods and tools for the design of safe systems. The definition of adequate mathematical models of the behavior of the systems allows the definition of formal calculi. They in turn form a basis for the construction of algorithms for the analysis, but also for the transformation of specifications towards an implementation. They can then be implemented in software environments made available to the users. A necessary complement is the setting-up of software engineering, programming, modeling, and validation methodologies. The motivation of these problems is at the origin of significant research activity, internationally and in particular, in the European IST network of excellence ARTIST II (Advanced Real-Time Systems)[9].

3.1.2. Models, methods and techniques.

The state of the art upon which we base our contributions, is twofold.

From the point of view of discrete control, there is a set of theoretical results and tools, in particular in the synchronous approach, often founded on labeled transition systems finite or infinite [42], [49]. During the past years, methodologies for the formal verification [75], [52], control synthesis [77] and compilation, and extensions to timed and hybrid systems [72], [43] have been developed. Asynchronous models consider the interleaving of events or messages, and are often applied in the field of telecommunications, in particular for the study of protocols. A well-known formalism for reactive systems is STATECHARTS [67], which can be encoded in a synchronous model [44].

From the point of view of verification, we use the methods and tools of symbolic model-checking and of abstract interpretation. From symbolic model-checking, we reuse BDD techniques [45] for manipulating Boolean functions and sets, and their MTBDD extension for more general functions. Abstract Interpretation [55] is used to formalize complex static analysis, in particular when one wants to analyze the possible values of variables and pointers of a program. Abstract Interpretation is a theory of approximate solving of fix-point equations applied to program analysis. Most program analysis problems, among others reachability analysis, come down to solving a fix-point equation on the state space of the program. The exact computation of such an equation is generally not possible for undecidability (or complexity) reasons. The fundamental principles of Abstract Interpretation are: (i) to substitute to the state-space of the program a simpler domain and to transpose the equation accordingly (static approximation); and (ii) to use extrapolation (widening) to force the convergence of the iterative computation of the fix-point in a finite number of steps (dynamic approximation). Examples of static analysis based on abstract interpretation are the Linear Relation Analysis [56] and Shape Analysis [51].

The synchronous approach [10] [65], [66] to reactive systems design gave birth to complete programming environments, with languages like ARGOS, LUSTRE [11], ESTEREL [12], SIGNAL/ POLYCHRONY [13], SYNDEX [14], LUCID SYNCHRONE [15] or Mode Automata [16]. This approach is characterized by the fact that it considers periodically sampled systems whose global steps can, by synchronous composition, encompass a set of events (known as simultaneous) on the resulting transition. Generally speaking, formal methods are often used for analysis and verification; they are much less often integrated in the compilation or generation of executives (in the sense of executables of tasks combined with the host real-time operating system). They are notoriously difficult to use by end-users, who are usually specialists in the application domain, not in formal techniques. This is why encapsulating formal techniques in an automated framework can dramatically improve their diffusion, acceptance, and hence impact. Our work is precisely oriented towards this direction.

From the point of view of the executables and execution platforms for the implementation of embedded systems, there are software or middleware approaches and hardware-based approaches. Concerning the

[9] http://www.systemes-critiques.org/ARTIST
[10] http://www.synalp.org
[11] http://www-verimag.imag.fr/SYNCHRONE
[12] http://www.inria.fr/recherche/equipes/aoste.en.html
[13] http://www.irisa.fr/espresso/Polychrony
[14] http://www-rocq.inria.fr/syndex
[15] http://www.lri.fr/~pouzet/lucid-synchrone/
[16] http://www-verimag.imag.fr/PEOPLE/Florence.Maraninchi/MATOU

quantitative aspects of the problem, one can find techniques for structuring the programs in multiple tasks, possibly preemptable, based on the real-time operating system. Their durations and periods, for example, are taken into account within the framework of scheduling according to various strategies. The analytical approach, with the determination of schedulability of a set of real-time tasks with constraints, is a very active field of research, primarily turned towards the respect of computer-centered constraints only: the task characteristics are derived from measurements of periods and execution time imposed by the environment. There has been, until recently, only little work formalizing the relation with discrete models and control. The techniques of real-time control usually take into account only criteria internal to the computer system, related to the resources of computation. In other words, they have an open loop character. However, the progress of the reflexive systems, providing sensors (of reconfiguration) and actuators (of dynamic control of the system) make it possible to close the loop [50], [71]; we contribute to this new approach by the development of methods for control/scheduling co-design.

3.2. Issues in design automation for complex systems

Keywords: *compilation*, *design automation*, *formal methods*, *real-time executives*, *scheduling*, *synthesis*, *verification*.

3.2.1. Hard problems

The design of safe real-time control systems is difficult due to various issues, among them their complexity in terms of the number of interacting components, their parallelism, the difference of the considered time scales (continuous or discrete), and the distance between the various theoretical concepts and results that allow the study of different aspects of their behaviors, and the design of controllers. The European IST network of excellence ARTIST II identifies three principal objectives: hard real-time for critical applications (which concerns the synchronous approach), component-based design, and adaptive real-time systems for quality of service management.

A currently very active research direction focuses on the models and techniques that allow the automatic use of formal methods. In the field of verification, this concerns in particular the technique of model checking; the verification takes place after the design phase, and requires, in case of problematic diagnostics, expensive backtracks on the specification. We want to provide a more constructive use of formal models, using them to derive correct executives by formal computation and synthesis, integrated in a compilation process. We therefore use models throughout the design flow from specification to implementation, in particular by automatic generation of embeddable executives.

3.2.2. Applicative needs

They initially come from the fields of safety-critical systems (avionics, energy) and complex systems (telecommunication), embedded in an environment with which they strongly interact (comprising aspects of computer science and control theory). Fields with less strong criticality, or which support variable degrees of quality of service, such as in the multi-media domain, can also take advantage of methodologies that improve the quality and reliability of software, and reduce the costs of test and correction in the design.

Industrial acceptance, the dissemination, and the deployment of the formal techniques inevitably depend on the usability of such techniques by specialists in the application domain — and not in formal techniques themselves —, and also on the integration in the whole design process, which concerns very different problems and techniques. The application domains are rather rare where the actors are ready to employ specialists in formal methods or advanced control theory. Even then, the methods of systematic application of these theoretical results are not ripe. In fields like industrial control, where the use of PLC (Programmable Logic Controller [46]) is dominant, this question can be decisive.

Essential elements in this direction are the proposal of realistic formal models, validated by experiments, of the usual entities in control theory, and functionalities (*i.e.*, algorithms) that correspond indeed to services useful for the designer. Take for example the compilation and optimization taking into account the platforms of execution, possible failures, or the interactions between the defined automatic control and its implementation. A notable example for the existence of an industrial need is the activity of the ATHYS company (now belonging to DASSAULT SYSTEME) concerning the development of a specialized programming environment, CELLCONTROL, which integrates synchronous tools for compilation and verification, tailored to the application domain. In these areas, there are functionalities that commercial tools do not have yet, and to which our results contribute.

3.2.3. Our approach

We are proposing effective trade-offs between, on the one hand, expressiveness and formal power, and on the other, usability and automation. We focus on the area of specification and construction of correct real-time executives for discrete and continuous control, while keeping an interest in tackling major open problems, relating to the deployment of formal techniques in computer science, especially at the border with control theory. Regarding the applications, we propose new automated functionalities, to be provided to the users in integrated design and programming environments.

3.3. Main Research Directions

Keywords: *aspect-oriented programming*, *compositionality*, *controller generation*, *dedicated languages*, *distribution*, *fault tolerance*.

3.3.1. Principles

We intend to exploit our knowledge of formal techniques and their use, and of control theory, according to aspects of the definition of fundamental tools, and applications.

The integration of formal methods in an automated process of generation/compilation is founded on the formal modeling of the considered mechanisms. This modeling is the base for the automation, which operates on models well-suited for their efficient exploitation, by analysis and synthesis techniques that are difficult to use by end-users.

The creation of easily usable models aims at giving the user the role rather of a pilot than of a mechanics *i.e.*, to offer her/him pre-defined functionalities which respond to concrete demands, for example in the generation of fault tolerant or distributed executives, by the intermediary use of dedicated environments and languages.

The proposal of validated models with respect to their faithful representation of the application domain is done through case studies in collaboration with our partners, where the typical multidisciplinarity of questions across control theory and computer science is exploited.

The overall consistency of our approach comes from the fact that the main research directions address, under different aspects, the specification and generation of safe real-time control executives based on *formal models*.

We explore this field by linking, on the one hand, the techniques we use, with on the other, the functionalities we want to offer. We are interested in questions related to:

- dedicated languages and models for automatic control that are the interface between the techniques we develop and the end-users on the one hand, and the designers of formal models on the other;
- compositional modeling and analysis that aim at deriving crucial system properties from component properties, without the need to actually build and check the global system;
- static analysis and abstract interpretation methods for checking functional properties on models and generated programs;
- Aspect-Oriented Programming (AOP) that allows to express safety concerns separately from the functional part and to enforce them on programs.

3.3.2. Implementations of synchronous programs

This issue can be tackled differently depending on the execution platform. Based on a formal model of the program to be implemented, our approach is to obtain by compilation (*i.e.*, automatically):

- the distribution on a multiprocessor architecture, with code partitioning according to directives, and insertion of the necessary communication actions to ensure the coherence of control; the distribution must be correct with respect to the original specification, and must be optimized;
- fault tolerance by replication of computations on a multiprocessor architecture, and scheduling of computations according to the faults to be tolerated; such a scheduling must be optimized *w.r.t.* its length and reliability.

3.3.3. Control/scheduling co-design

The interaction of the intrinsic nature of the control we consider, with its real-time implementation can be tackled in two ways:

- scheduling for regulation where the scheduling scheme and parameters are designed to capture the control system requirements and to improve the quality of the implemented controller;
- regulation for scheduling where the latter is made adaptive and is dynamically controlled by using techniques from control theory.

3.3.4. Automatic generation of correct controllers

We use techniques of discrete controller synthesis, especially the tools SIGALI [74] and Mode Automata [73] within an automated framework, for:

- multi-mode multi-tasking systems where the management of interactions (exclusions, optimization of cost or quality criteria, ...) is obtained by synthesis;
- a locally imperative, globally declarative language whose compilation comprises a phase of discrete controller synthesis;
- fault-tolerance management, by reconfiguration following objectives of consistent execution, functionality fulfillment, boundedness and optimality of response time.

4. Application Domains

4.1. Industrial applications.

Our applications are in embedded systems, typically: robotics, automotive, telecommunications, systems on chip (SoC). In some areas, safety is critical, and motivates the investment in formal methods and techniques for design. But even in less critical contexts, like telecommunications and multimedia, these techniques can be beneficial in improving the efficiency and quality of designs, as well as the design, production and test costs themselves.

Industrial acceptance of formal techniques, as well as their deployment, goes necessarily through their usability by specialists of the application domain, rather than of the formal techniques themselves. Hence our orientation towards the proposal of domain-specific (but generic) realistic models, validated through experience (*e.g.*, control tasks systems), based on formal techniques with a high degree of automation (*e.g.*, synchronous models), and tailored for concrete functionalities (*e.g.*, code generation).

4.2. Industrial design tools.

The commercially available design tools (such as UML with real-time extensions, MATLAB/ SIMULINK/ dSPACE [17]) and execution platforms (OS such as VXWORKS, QNX, real-time versions of LINUX ...) propose a collection of functionalities without accompanying it by design or verification methods. Some of them, founded on models of reactive systems, come close to tools with a formal basis, such as for example STATEMATE by iLOGIX.

Regarding the synchronous approach, commercial tools are available: SCADE (based on LUSTRE), ESTEREL [18], SILDEX [19] (based on SIGNAL), specialized environments like CELLCONTROL for industrial automatism (by the INRIA spin-off ATHYS). One can note that behind the variety of actors, there is a real consistency of the synchronous technology, which makes sure that the results of our work related to the synchronous approach are not restricted to some language due to compatibility issues.

The scheduling methods we propose, are of interest for the designers of embedded applications, who lack adequate design methods to effectively use the tools offered by the RTOS. The dissemination of these methods can be done via the success of applications (as in the former European project TELEDIMOS), or by distribution in the context of free software around the real-time/embedded versions of LINUX [20].

4.3. Current industrial cooperations.

Regarding applications and case studies with industrial end-users of our techniques, we cooperate with STMicroelectronics on compositional verification and abstract interpretation for the TLM-based System-on-Chip design flow, and with DCN on the multi-criteria real-time scheduling issues for action planning of their defense systems.

5. Software

5.1. Orccad

Participants: S. Arias, R. Pissard-Gibollet, C. Junke, H. Houichi, D. Simon [contact person].

ORCCAD [4][21] is a software environment that allows the design and implementation of the discrete and continuous control of complex robot systems. It also allows the specification and validation of missions to be realized by this system.

It is mainly intended for critical real-time applications in robotics, in which automatic control aspects (*servo loops*, control) have to interact narrowly with the handling of discrete events (*exception handling*). ORCCAD offers a complete and coherent vertical solution, ranging from the high level specification to real-time code generation.

ORCCAD is supported by the *Support Expérimentations & Développement (SED)* service of INRIA-Rhône-Alpes. ORCCAD is used by the experimental robotics platforms of INRIA-Rhône-Alpes. New functionalities and updates are developed jointly by the *SED* service and the researchers of the Pop Art team.

The former V3 version allows for the automatic generation of real-time single-rate controllers running on top of VxWorks, Solaris and Linux and multi-rate controllers running on top of Linux and Xenomai.

Although it has been developed years ago, the basic concepts upon which the ORCCAD architecture relies still appear to be solid in the field of software development for robot control [30][27], and compares well with other tools dedicated for real-time control implementation [34].

[17] http://www.dspaceinc.com
[18] http://www.esterel-technologies.com
[19] http://www.tni-valiosys.com
[20] http://www.realtimelinuxfoundation.org/projects/projects.html
[21] http://www.inrialpes.fr/iramr/pub/Orccad

However the ORCCAD V3 software was designed with proprietary tools that moreover are now becoming obsolete. During the year 2006, ORCCAD has been deeply re-engineered to be compliant with open-source and free software tools (Java/Eclipse/XML) while being fully compliant with V3 based former projects.

5.2. NBac

Participant: B. Jeannet [contact person].

NBAC (Numerical and Boolean Automaton Checker)[22] is a verification/slicing tool for reactive systems containing combination of Boolean and numerical variables, and continuously interacting with an external environment. NBac can also handle the same class of hybrid systems as the HyTech tool. It aims at handling efficiently systems combining a non-trivial numerical behaviour with a complex logical (Boolean) behaviour.

NBAC is connected to 2 input languages: the synchronous dataflow language LUSTRE, and a symbolic automaton-based language, AUTOC/AUTO, where a system is defined by a set of symbolic hybrid automata communicating via valued channels. It can perform reachability analysis, co-reachability analysis, and combination of the above analyses. The result of an analysis is either a verdict to a verification problem, or a set of states together with a necessary condition to stay in this set during an execution. NBAC is founded on the theory of abstract interpretation: sets of states are approximated by abstract values belonging to an abstract domain, on which fix-point computations are performed.

It has been used for verification and debugging of LUSTRE programs [69] [59]. It is connected to the LUSTRE toolset [23] It has also been used for controller synthesis of infinite-state systems The fact that the analyses are approximated results simply in the obtention of a possibly non-optimal controller. In the context of conformance testing of reactive systems, it is used by the test generator STG [53] [70] for selecting test cases.

5.3. Prometheus

Participants: G. Goessler [contact person], A.M. Khan.

The BIP component model (Behavior, Interaction model, Priority) [64] [62] has been designed to support the construction of heterogeneous reactive systems involving different models of computation, communication, and execution, on different levels of abstraction. By separating the notions of behavior, interaction model, and execution model, it enables both heterogeneous modeling, and separation of concerns.

The verification and design tool PROMETHEUS implements the BIP component framework. PROMETHEUS is regularly updated to implement new developments in the framework and the algorithms for compositional verification of properties like deadlock-freedom, liveness, and reachability. It has allowed us to carry out several complex case studies from the system-on-chip and bioinformatics domains.

5.4. Implementations of synchronous programs

Participants: A. Girault [contact person], H. Kalla.

5.4.1. Code distribution

OCREP distributes automatically synchronous programs according to specifications given by the user. Concretely, starting from a centralized source synchronous program obtained either with the LUSTRE or the ESTEREL compiler, from a number of desired computing locations, and an indication of where each input and output of the source program must be computed, OCREP produces several programs, one for each location, each one computing only its assigned variables and outputs, and communicating harmoniously. Their combined behavior is equivalent to the behavior of the centralized source program and that there is no deadlock.

[22] http://pop-art.inrialpes.fr/people/bjeannet/nbac/
[23] http://www-verimag.imag.fr/SYNCHRONE/index.php?page=tools

Currently our software OCREP is distributed in the form of executable on the web[24]. It consists in 15000 lines of C++ code. In 2002, a contract for industrial transfer was drawn up with France Télécom R&D in order to integrate OCREP into their compiler SAXO-RT for ESTEREL programs.

5.4.2. Fault tolerance

We have been cooperating for several years with the INRIA team AOSTE (INRIA Sophia-Antipolis and Rocquencourt) on the subject of fault tolerance. In particular, we have implemented several new heuristics for fault tolerance and reliability within their software SYNDEX [25]. This has taken place within the framework of the European project EAST-EEA in which we participated together with AOSTE. In this context, we have developed several new scheduling heuristics that produce static multiprocessor schedules tolerant to a specified number of processor and communication link failures [38]. The basic principles upon which we rely to make the schedules fault tolerant is, on the one hand, the active replication of the operations [9], and on the other hand, the active replication of communications for point-to-point communication links, or their passive replication coupled with data fragmentation for multi-point communication media (*i.e.*, buses) [25].

5.5. Prototypes

5.5.1. Automatic Controller Generation

Participants: G. Delaval [contact person], E. Dumitrescu, A. Girault, E. Rutten.

We have developed a software tool chain to allow the specification of models, the controller synthesis, and the execution or simulation of the results. It is based on existing synchronous tools, and thus consists primarily in the use and integration of SIGALI [26] and of Mode Automata [27].

Useful component templates and relevant properties can be materialized, on one hand by libraries of task models, and, on the other hand, by properties and synthesis objectives. A prototype compiler has been developed to demonstrate a domain-specific language, named NEMO, for multi-task controllers (see Section 6.4).

5.5.2. Rapture

Keywords: *Markov Decision Processes*, *Probabilistic verification*.

Participant: B. Jeannet [contact].

RAPTURE [68] [57] is a verification tool that was developed jointly by BRICS (Denmark) and INRIA in years 2000–2002. The tool is designed to verify reachability properties on Markov Decision Processes (MDP), also known as Probabilistic Transition Systems. This model can be viewed both as an extension to classical (finite-state) transition systems extended with probability distributions on successor states, or as an extension of Markov Chains with non-determinism. We have developed a simple automata language that allows to describe a set of processes communicating over a set of channels *à la* CSP. Processes can also manipulate local and global variables of finite type. Probabilistic reachability properties are specified by defining two sets of initial and final states together with a probability bound. The originality of the tool is to provide two reduction techniques that limit the state space explosion problem: automatic abstraction and refinement algorithms, and the so-called essential states reduction.

5.5.3. Libraries for Abstract Interpretation

Participant: B. Jeannet [contact person].

[24] http://pop-art.inrialpes.fr/people/girault/Ocrep/
[25] http://www-rocq.inria.fr/syndex
[26] http://www.irisa.fr/vertecs/Logiciels/sigali.html
[27] http://www-verimag.imag.fr

We also develop and maintain libraries of general use for people working in the static analysis and abstract interpretation community.

APRON library [28]: implements a common API for numerical abstract domain. Three abstract domains has already been connected to this API: intervals, octagons, and convex polyhedra. Two bindings are available: C and OCAML. It has been developed with the partners of the ACI-SI APRON (see Section 8.2.3).

Analyzer : a generic fix-point engine written in OCAML. It allows to solve systems of fix-point equations on a lattice, using a parameterized strategy for the iteration order and the application of widening.

6. New Results

6.1. Distribution of higher-order synchronous dataflow programs

Participants: G. Delaval, A. Girault [contact person].

We are working in the context of the ALIDECS ACI[29] and the LUCID SYNCHRONE dataflow synchronous programming language [47] [48] [54]. LUCID SYNCHRONE is a higher-order programming language since streams can carry either scalar values (like classical flows in LUSTRE) or functions (hence streams of streams functions). Functions are therefore first class citizens. This feature makes it a programming language well suited for dynamically reconfigurable embedded systems such as software-defined radio. Our goal is to provide, by compilation of one synchronous program source with location annotations, an executable program for each physical location specified. The result of the parallel execution of these programs will be then a functionally distributed system, whose semantic, abstraction made of the computations' locations, will be the same as the program without the location annotations.

We have extended LUCID SYNCHRONE with new primitives allowing the programmer to specify a *distributed architecture* and express the *location* of flows. The first step to achieve the desired distributed program is to propagate the available location informations to the flows for which this was not specified. This is usually done on a fully inlined program, but unfortunately this does not work for higher-order programs, since such programs cannot be, in general, inlined in order to perform such a semantic computation. Furthermore, we are interested in modular compilation, in the framework of which such inlining cannot be fulfilled. Therefore, we have proposed a new type system to check the consistency of the location specifications w.r.t. the distributed architecture, to infer the location of the non located flows, and to insert automatically communication primitives at the right place. We are currently working on the implementation of this type system and on semantics preserving issues.

Gwenaël Delaval is doing his PhD on this topic, co-advised by Alain Girault and Marc Pouzet (LRI, Orsay).

6.2. Reliable distributed real-time embedded systems

Participants: A. Girault [contact person], H. Kalla, E. Saule, D. Trystram.

We have continued our work on the automatic generation of reliable and multiprocessor static schedules, with bi-criteria scheduling heuristics. The context of our work is to start from an algorithmic specification under the form of a DAG of operations (Directed Acyclic Graph), and an architecture specification under the form of a bipartite graph of processors and communication media.

[28] http://apron.cri.ensmp.fr/library/
[29] http://www-verimag.imag.fr/SYNCHRONE/alidecs/

On the theoretical side, we have chosen a simplified reliability model where we assume that the communication media are reliable. In this context, we have designed a new method that dissociates, one the one hand the spatial allocation of the operations to the processors, and on the other the temporal allocation of the operations allocated to the same processor. According to our simplified reliability model, the reliability of the resulting schedule depends only on the spatial allocation. Hence, our method first optimizes the reliability of the schedule during the spatial allocation phase, then optimizes the makespan of the schedule during the temporal allocation phase. Regarding the bi-criteria aspect of this problem, we have chosen to transform the reliability criterion into a constraint: this means that we iteratively fix several reliability thresholds, and for each such threshold we solve our problem by minimizing the makespan of the schedule while remaining above the reliability threshold. As a result, we are able to obtain, for a given instance, a set of non-dominated solutions (in the Pareto sense), among which the user can choose the compromise that fits his requirements best. We are also able to compute the average compromise between the reliability gain and the makespan overhead. We have conducted extensive simulations of our scheduling algorithm and have compared it against the popular HEFT algorithm [80].

On the practical side, we are improving the cost function used inside our bi-criteria scheduling heuristic. This work uses a more general reliability model, where communication media have a rate of failure per time unit, just like the processors. Our bi-criteria cost function attempts to optimize both the reliability and the makespan of the resulting schedule. The difficulty arises from the fact that these two measures (the reliability and the makespan) have drastically different orders of magnitude and evolve in radically different ways during the incremental building of the schedule.

We have therefore proposed a new framework for the bi-criteria multiprocessor scheduling problem. Our first criteria remains the static schedule's length (crucial to assess the system's real-time property). For our second criteria, we consider the global failure rate of the system (GSFR) instead of the usual reliability, because it does not depend on the schedule length like the reliability does (due to its computation in the classical reliability model of Shatz). The GSFR of a static schedule $\mathcal{S}$ is the failure rate of $\mathcal{S}$ seen as if it were a single operation scheduled onto a single processor. Thanks to the GSFR, we control better the replication factor of each individual task of the dependency task graph given as an input specification, with respect to the desired failure rate. Like above, we solve this bi-criteria optimization problem by considering the failure rate as a constraint, and by minimizing the schedule length. We are therefore able to compute the average compromise between the GSFR and the makespan overhead.

6.3. Control and scheduling co-design

Participants: D. Robert, O. Sename, D. Simon [contact person].

The real-time community has usually considered that control tasks have fixed periods, hard deadlines and worst-case execution times. This assumption has served the separation of control and scheduling designs, but has led to under utilization of CPU resources. However current real-time design methods and associated analysis tools do not provide a model flexible enough to fit well with control systems engineering requirements.

We aim to provide an *integrated control and scheduling co-design* approach [19]. It is assumed that robust control focusing on timing uncertainties may provide a first level of fault tolerance. When the capabilities of feedback scheduling are exceeded, exception handling will be handled by a decision process working on a discrete events time scale. The proposed methodology will be assessed using realistic simulations and experiments.

6.3.1. Integrated control/scheduling co-design

In our framework the feedback scheduling is designed w.r.t a QoC (Quality of Control) measure. The QoC criterion captures the control performance requirements, and the problem can be stated as QoC optimization under constraint of available computing resources. However, preliminary studies suggest that a direct synthesis of the scheduling regulator as an optimal control problem leads, when it is tractable, to a solution too costly to be implemented in real-time [50]. Practical solutions will be found in the currently available control theory and tools or in enhancements and adaptation of current control theory. We propose in Figure 1 a hierarchical

control structure : besides the usual process control loops we add an outer control loop which goal is to manage the execution of the real-time application through the control of the scheduling parameters of the inner loops. Together with the outer loop (working on a periodic sampled time scale) we also need a scheduling manager working on a discrete events time scale to process exception handling and admission control.

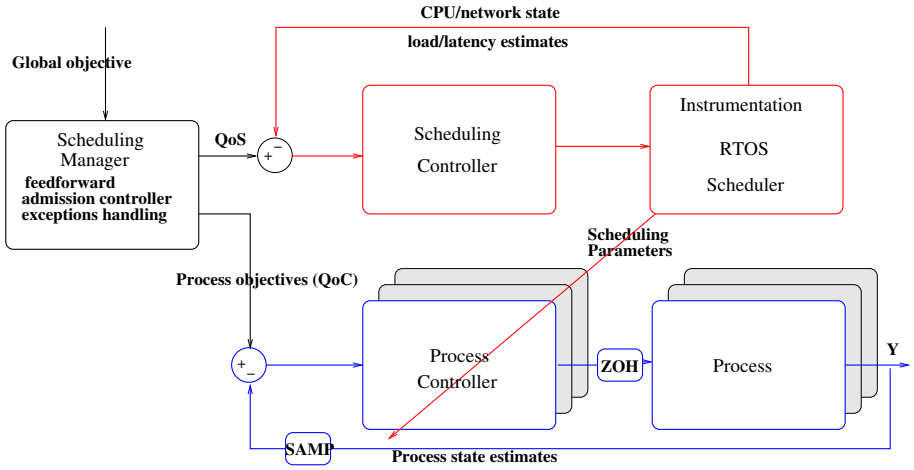

Figure 1. Hierarchical control structure.

The task periods directly affect the computing load, they have been chosen as actuators. They can be implemented through software variable clocks. As timing uncertainties cannot be avoided and are difficult to model or measure, we currently design robust control algorithms using the H_∞ control theory, which have been successfully simulated and experimentally validated [31].

This methodology is supported by ORCCAD where a runtime library for multi-rate multitasking has been developed and integrated. It will be further improved using a QoS-based management of the timing constraints to fully benefit from the intrinsic robustness of closed-loop controllers w.r.t. timing uncertainties. Further improvements are sketched at the end of the PhD of D. Robert [78], where the mean square of the tracking error during the scheduling window appears to be an effective measure of the quality of control of a inverted pendulum.

6.3.2. Synthesis of variable sampling control

As variable control periods are used as actuators in feedback schedulers it is necessary to ensure the stability of the control laws under variable sampling conditions. Indeed it is known that on-line switches between stable controlled systems sampled at different rates may lead to instability. The synthesis of control laws using variable sampling has been developed via new extensions of the gain scheduling and Linear Parameter Varying (LPV) design methods, considering here that the sampling period is the varying parameter [29][40].

The first point is the problem formulation such that it can be solved following the LPV design of [41]. We first propose a parametrised discretization of the continuous time plant and of the weighting functions, leading to a discrete-time sampling period dependent augmented plant leading to the discrete-time LPV system (1).

$$G_d : \left\{ \begin{array}{rcl} x_{k+1} & = & A_d(h)\, x_k + B_d(h)\, u_k \\ y_k & = & C_d(h)\, x_k + D_d(h)\, u_k \end{array} \right. \tag{1}$$

with h ranging in $[h_{min}; h_{max}]$.

To get a polytopic model (and then apply an LPV design), we approximate the exponential by a Taylor series of order N as:

$$e^{Mh} \approx \sum_{i=0}^{N} \frac{(Mh)^i}{i!}, \tag{2}$$

which leads, with $H = [h \;\; h^2 \;\; \ldots \;\; h^N]$, to:

$$\begin{array}{l} A_d(h) \approx I + \sum_{i=1}^{N} \frac{A^i}{i!} h^i := A_d(H) \\ B_d(h) \approx \sum_{i=1}^{N} \frac{A^{i-1}B}{i!} h^i := B_d(H) \end{array} \tag{3}$$

As the self-scheduled controller will be a convex combination of 2^N "vertex" controllers, the choice of the series order N gives a trade-off between the approximation accuracy and the controller complexity. However to reduce the complexity (and the conservatism of the corresponding control design as well), a reduction of the polytope is proposed using the dependency between the parameters, which actually are the successive powers of the sampling period h : as illustrated in figure 2 the number of vertices to be considered for the convex combination of the global controller becomes $N + 1$ rather than 2^N [78], [79].

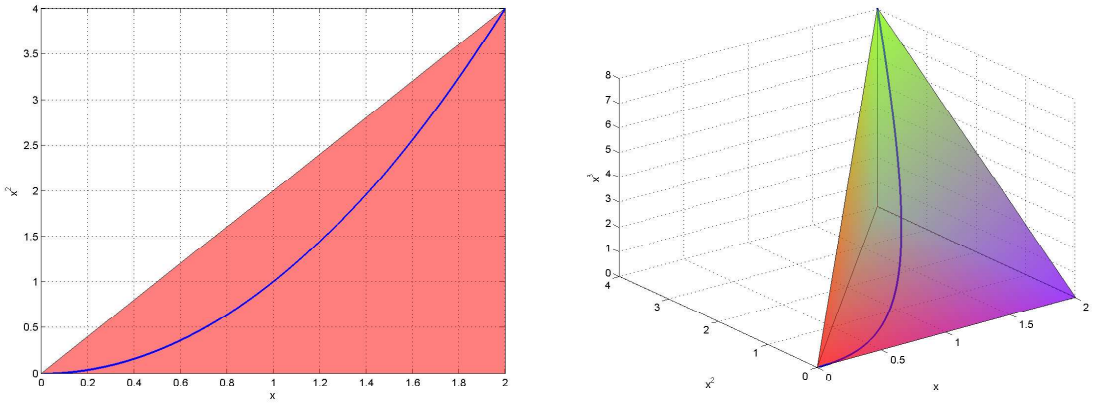

Figure 2. Polytope reduction for N=2 and N=3

In the H_∞ framework, the general control configuration of figure 3 is considered, where W_i and W_o are weighting functions specifying closed-loop performances. The objective is here to find a controller K such internal stability is achieved and $\|\tilde{z}\|_2 < \gamma \|\tilde{w}\|_2$, where γ represents the H_∞ attenuation level. Classical control design assumes constant performance objectives and produces a controller with an unique sampling period. This sampling period is chosen according to the controller bandwidth, the noise sensibility and the availability of computation resources.

When the sampling period varies the usable controller bandwidth also varies and the closed-loop objectives should logically be adapted. Thus the performance templates W_i and W_o are split into two parts : a constant part with constant poles and zeros to compensate for oscillations or flexible modes independent of the sampling period, and a variable part contains poles and zeros whose pulsations are expressed as an affine function of the frequency $f = 1/h$. The interconnection of figure 3a between the discrete-time polytopic model of the plant H and the weighting functions W_i and W_o leads to the discrete-time LPV augmented plant $P(\theta)$ mapping

exogenous inputs w and control inputs u to controlled outputs z and measured outputs y, with $x \in \mathbb{R}^n$, be given by the polytopic model

$$\left\{ \begin{array}{l} x_{k+1} = \overline{A}(\theta)x_k + \overline{B}_1(\theta)w + \overline{B}_2(\theta)u \\ z = \overline{C}_1(\theta)x_k + \overline{D}_{11}(\theta)w + \overline{D}_{12}(\theta)u \\ y = \overline{C}_2(\theta)x_k + \overline{D}_{21}(\theta)w + \overline{D}_{22}(\theta)u \end{array} \right. \tag{4}$$

where the dependence of $A(\theta)$, $B(\theta)$, $C(\theta)$ and $D(\theta)$ on θ is affine and the parameter vector θ, ranges over a fixed polytope Θ.

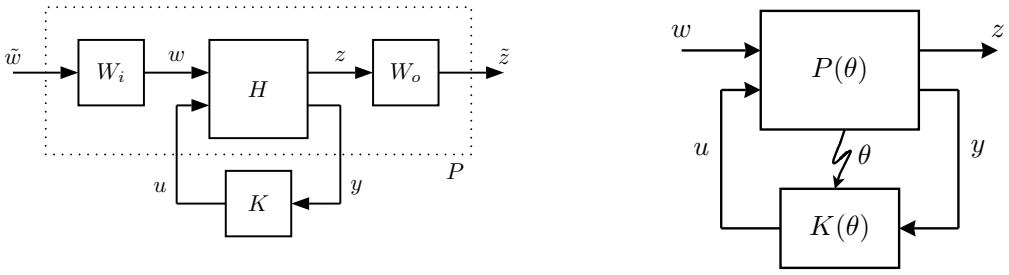

Figure 3. a: Focused interconnection - b: Closed-loop of the LPV system

The self-scheduled controller $K(\theta)$ is the convex combination of the elementary controllers synthesized at the vertex of the polytope.

$$\begin{array}{rcl} K(\theta) : \left(\begin{array}{cc} A_K(\theta) & B_K(\theta) \\ C_K(\theta) & D_K(\theta) \end{array} \right) & = & \sum_{i=1}^{r} \alpha_i \left(\begin{array}{cc} A_{K_i} & B_{K_i} \\ C_{K_i} & D_{K_i} \end{array} \right) \\ \text{with } \alpha_i \text{ such that } \theta & = & \sum_{i=1}^{r} \alpha_i \omega_i \end{array} \tag{5}$$

Under mild conditions this controller ensures the quadratic stability of the closed-loop system and the limitation of the input/output transfer $\mathcal{L}_2$-induced norm whatever are the variations of the sampling period h in the specified range.

6.3.3. Simulations and experiments

The LPV design of variable sampling controllers has been experimentally validated using a "T" inverted pendulum available at LAG (figure 4).

As such a T pendulum system is difficult to be controlled, our main objective is to get a closed-loop stable system, to emphasise the practical feasibility of the proposed methodology for real-time control.

The sampling period range has been set in the interval $[1, 3]\ ms$. The performance objectives are represented by weighting functions and may be given by the usual transfer functions:

$$\begin{array}{rcll} W_e(p, f) & = & \frac{p\, M_S + \omega_S(f)}{p + \omega_S\, \epsilon_S} & \omega_S(f) = h_{min}\, \omega_{S_{max}}\, f \\ W_u(p, f) & = & \frac{1}{M_U} & \end{array}$$

where $f = 1/h$, $\omega_{S_{max}}$ = 1,5 rad/s, M_S = 2, ϵ_S = 0.01 and M_U = 5.

The plant is controlled through Matlab/Simulink using the Real-time Workshop and xPC Target.

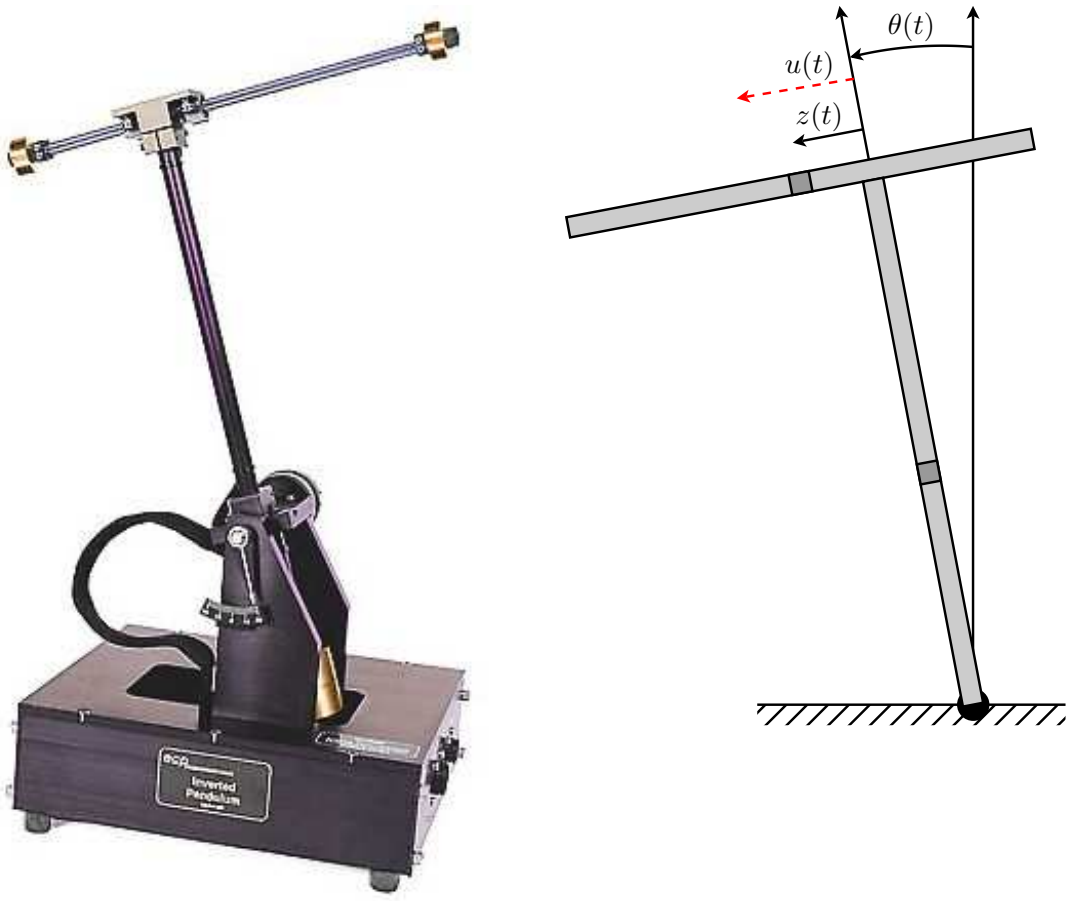

Figure 4. the T inverted pendulum

Some results are given in figures 5. The pendulum is requested to follow a square motion while the sampling period varies following a sinusoidal or square profile. As expected the settling time is minimal when the sampling period is maximal, and vice versa. In the same way, there is no abrupt changes in the control input even when the sampling period suddenly varies from 1 to 3 ms as in figure 5b.

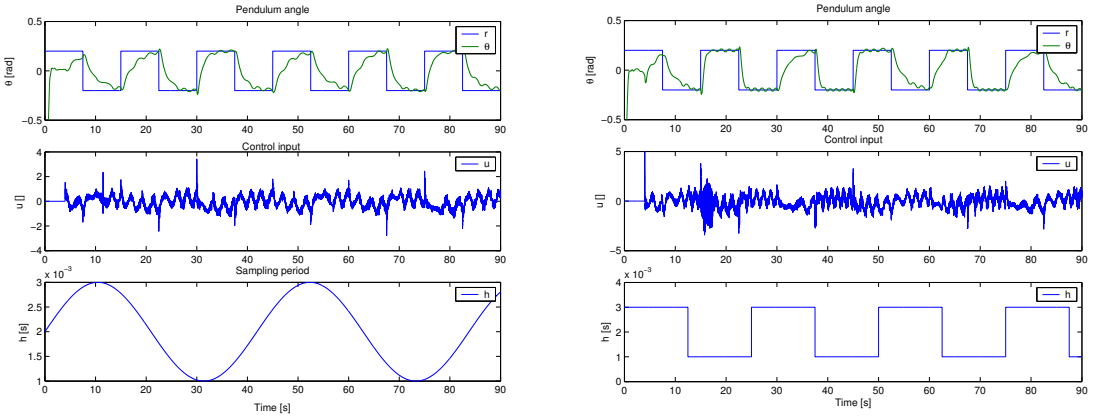

Figure 5. Experimental motion of the T pendulum: a) sinusoidal period - b: square period

6.4. Automatic generation of correct controllers

Participants: G. Delaval [contact person], E. Dumitrescu, A. Girault.

We address the difficulty of safely designing complex system controllers by proposing a method applying formal design techniques to the domain of embedded control systems. Such techniques are considered difficult to use, amongst other things because of the required theoretical competence. A general notion of *hidden formal methods* advocates for fully automated techniques, integrated into a design process and tool. The formal technique we aim to encapsulate into a tool chain is *discrete controller synthesis* [76].

6.4.1. Domain-specific language for application of discrete controller synthesis

We propose a simple programming language, called NEMO [23], specific to the domain of multi-task real-time control systems, such as in robotics, automotive or avionics systems. The notion of task is related to the one used in the ORCCAD tool [4]. It can be used to specify a set of resources with usage constraints, a set of tasks that consume them according to various modes, and applications sequencing the tasks. We obtain automatically an application-specific task handler that correctly manages the constraints (if any), through a compilation-like process including a phase of discrete controller synthesis. We use synchronous languages, modeling techniques and tools, particularly the Mode Automata language [73] and the SIGALI synthesis tool [74].

We are considering to confront NEMO with case-studies in manufacturing systems. We are also considering extensions with models of the environment, which can have a decisive influence on the existence of solutions for the synthesis, as well as a more general language, less domain-specific, where controller synthesis is integrated in the compilation.

6.4.2. Fault tolerant systems

In order to automatically obtain fault tolerant real-time systems, we investigate a new solution based on the application of discrete controller synthesis (DCS). The real-time systems we consider consist of a set of tasks and a set of distributed, heterogeneous processors. The latter are fail-silent, and an environment model can detail actual fault patterns. We apply DCS with objectives w.r.t. consistent execution, functionality

fulfillment, and some optimizations. We construct a manager that ensures fault tolerance by migrating the tasks automatically, upon occurrence of a failure, according to the policy given by the objectives. In this context, we have addressed different kinds of failures (crash, value, or Byzantine) affecting different kinds of hardware components (processors, communication links, actuators, or sensors) [60], [58][26].

We have new results concerning optimal synthesis along paths, and its application to the control of sequences of reconfigurations. Tasks that are interrupted by a fault can be restarted at their last checkpoint, and the control of the configuration restarts the tasks by placing them on processors chosen w.r.t. an objective on the shortest total execution time of the application. We therefore combine, on the one hand, guarantees on the safety of the execution by tolerating faults, and on the other hand, guarantees on the worst case execution time of the resulting dynamically reconfiguring fault tolerant system.

This work is conducted in collaboration with H. Marchand (VERTECS team from INRIA Rennes) and E. Dumitrescu (INSA Lyon).

6.5. Static Analysis and Abstract Interpretation

6.5.1. Design and Implementation of a common API for numerical abstract domains

Participant: B. Jeannet [contact person].

This new result corresponds to the software described in section 5.5.3, in the context of the ACI-SI APRON (see 8.2.3). The different teams were using different libraries and different interfaces for the analysis of numerical variables of programs. One of the goal of the ACI-SI APRON was to design a common API in order to gain the following benefits:

1. The ability to choose very easily the numerical abstract domain in use, and to compare the precision and efficiency of different abstract domains and/or implementations;
2. The factorization of higher-level layers built on the basic common interface. This offers a higher-level interface and simplifies the design and implementation of analysis tools.

The design required several meetings in 2005, as it had to remain simple yet to satisfy at least all the members of the project, which are interested in different kind of analysis. The year 2006 was devoted to the implementation, which required a significant effort (13000 LOC). This was done in collaboration with A. Miné (ENS Paris). The resulting library is distributed since 07/2006 under LGPL license [30] and is already used or in evaluation by several teams in France (LANDE and POP ART projects, ENS Paris, VERIMAG, CEA-Saclay).

6.5.2. Verification of Communication Protocols Using Abstract Interpretation of FIFO queues

Participants: T. Le Gall, B. Jeannet [contact person].

The verification of communication protocols or distributed systems that can be modeled by set of sequential machines communicating via unbounded FIFO channels is the topic of the PhD of Tristan Le Gall. The main challenge of its PhD is the verification of such systems in the case where

- the communicating machines are themselves infinite-state processes;
- the values sent to FIFO channels belong to unbounded datatypes.

The approach we follow is based on the theory of Abstract Interpretation. The applications of such verification techniques are the analysis of communicating protocols, which may contain subtle bugs, the automatic synthesis of controllers for distributed systems in order to ensure a correct global behavior, and possibly the diagnosis of distributed systems.

[30] http://apron.cri.ensmp.fr/library/

We focused first on the case of Communicating Finite-State Machines (CFSM) [28], a model where the values sent into FIFO queues belongs to bounded datatypes. Unlike recent related works based on acceleration techniques, we applied an Abstract Interpretation approach to such systems, which consists in using approximated representations of sets of configurations. We show that the use of regular languages together with an extrapolation operator provides a simple and elegant method for the analysis of CFSM, which is moreover often as accurate as acceleration techniques, and in some cases more expressive. Last, when the system has several FIFO queues, our method can be implemented either as an attribute-independent analysis or as a more precise (but also more costly) attribute-dependent analysis. We implemented both analyzes and provided experimental evidence of their efficiency and precision.

We have also designed and implemented an abstract domain for the a general model of communicating system where both processes and values contained in FIFO queues are unbounded. The implementation should be soon connected to the NBAC tool (see Section 5.2). A research report is being written.

6.5.3. *Supervisory Control of Symbolic and Hybrid Transition Systems*

Participants: T. Le Gall, B. Jeannet [contact person], H. Marchand.

We have been interested in solving the safety controller synthesis problem for various models (from finite transition systems to hybrid systems). Within this framework, we have been mainly interested in an intermediate model: symbolic transition systems. Due to the infiniteness of the alphabet, we have chosen to redefine the concept of controllability by introducing the notion of dynamic uncontrollable transitions (the controllability status is carried on by the symbolic transitions by means of guards, instead of the events). We focus on *safety requirements*, modeled by observers that encode the negation of a safety property. We then defined synthesis algorithms based on abstract interpretation techniques so that we can ensure convergence of fix-point computations in a finite number of steps [17].

6.6. Component-based Construction

Participants: G. Goessler [contact person], P. Fradet, R. Guimaraes, A. Girault, M. Tivoli.

The work on component-based construction of correct embedded systems is a cornerstone of our activity. Component-based construction techniques are crucial to overcome the complexity of embedded systems design. However, two major obstacles need to be addressed: the heterogeneous nature of the models, and the lack of results to guarantee correction of the composed system. The heterogeneity of embedded systems comes from the need to integrate components using different models of computation, communication, and execution, on different levels of abstraction and different time scales. The component framework and verification and construction algorithms have to support this heterogeneous nature of the components.

6.6.1. *Efficient Verification Techniques*

The BIP (Behavior, Interaction model, Priority) component model presented in [64] [62] has been designed to support the construction of heterogeneous reactive systems. By separating the notions of behavior, interaction model, and execution model, it enables both heterogeneous modeling, and separation of concerns.

We have shown how the framework can be used to discuss properties of systems including local and global deadlock, reachability, progress of subsystems, fairness, liveness, and robustness. In most cases, direct testing of the properties relies on an exploration of the global state space and hence cannot be performed efficiently. We have established a condition that can be tested in polynomial time and guarantees liveness of a component, a set of components, or an interaction. Part of these results have been implemented in the PROMETHEUS tool [63].

In collaboration with Frédéric Lang (VASY team), we have developed a method to translate BIP models into the CADP [31] framework, so as to apply different verification techniques on the same model, and study combination of complementary compositional verification algorithms. We have implemented this method in a module within the PROMETHEUS tool. The three layers of the BIP component model are translated separately:

[31] http://www.inrialpes.fr/vasy/cadp

- The behavior model of each basic component is translated into a LOTOS process.
- The interaction model is converted into a set of EXP.OPEN synchronization vectors.
- The execution model, consisting of constraints on the state space and on the enabling conditions of interactions, is translated as follows: (1) special actions called observers, are added in the LOTOS model of each sequential process to enable observation of local variables that occur in state invariants or in interaction constraints, (2) vectors modeling synchronizations between observers are added in the EXP.OPEN model to identify the states at which the invariants or interaction constraints are violated, and (3) priorities are added in the EXP.OPEN model to cut the transitions which violate the interaction constraints or whose source state violate some invariant.

Additionally, an SVL script is generated to orchestrate the generation of intermediate files. The translation was experimented on a case study in the domain of systems-on-chip. This work led to a MSc thesis [39].

6.6.2. Adapter Synthesis for Synchronous Components

In the context of the ACI ALIDECS (see section 8.2.2), we have an ongoing research project on the definition of a language and framework for the construction of safe embedded systems based on synchronous components.

Building a real-time system from existing components introduces several problems, mainly related to compatibility, communication, and QoS issues. We have proposed an approach to automatically synthesize adapters in order to solve black-box integration incompatibilities within a lightweight component model. Adapter synthesis allows the developer to automatically build *correct-by-construction* systems from third-party components, hence, reducing time-to-market and improving reusability.

A component interface includes a formal description of the *interaction protocol* of the component with its expected environment. The interface language is expressive enough to specify QoS constraints such as *latency*, *duration*, and *controllability* of the component actions (ports), as well as the component's *clock*, *i.e.*, its activation frequency. Based on results from Petri net and supervisory control theory, we have developed an algorithm which automatically synthesizes correct-by-construction and bounded-memory adapter components from the interface specification of the components. The generated adapters coordinate the interaction behavior of the components and buffer their communications, in order to avoid deadlocks. The algorithm has been implemented in the *SynthesisRT* tool, and a research report has been written (accepted at TACAS'07).

6.6.3. Component Fusion

Given a system of concurrent components communicating through FIFO queues in a Kahn process network-style, the technique of *component fusion* [8] allows to obtain a sequential implementation thus getting rid of context switching and improving efficiency. In this work, we extend the component language with non-determinism features (*e.g.*, testing the size of a queue). We have been working on extending the fusion algorithm to fulfill the following requirements: (1) preserve functional (non-confluent) non-determinism, so as to observe the same non-deterministic behavior as in the original component network; (2) eliminate confluent non-determinism as far as possible to improve performance; (3) guarantee fairness. This work is still in progress.

6.7. Aspect-oriented programming

Participants: S. Djoko Djoko, R. Douence, P. Fradet [contact person], A. Girault.

The goal of Aspect-Oriented Programming (AOP) is to isolate aspects (such as security, synchronization, or error handling) which cross-cut the program basic functionality and whose implementation usually yields tangled code. In AOP, such aspects are specified separately and integrated into the program by an automatic transformation process called *weaving*.

Although this new paradigm has great practical potential, it still lacks formalization and undisciplined uses make reasoning on programs very difficult. Our work on AOP addresses these issues by studying foundational issues (semantics, analysis, verification) and by considering domain-specific aspects (availability or fault tolerance aspects) as formal properties.

6.7.1. Semantics and analysis of AOP

Existing semantics are typically based on a specific programming paradigm (*e.g.*, object oriented, functional, process based) and model definite aspect languages. We have defined a common aspect semantics base (CASB) as a small step semantics that allows the modular introduction of formal semantic descriptions of different aspect mechanisms [37]. The semantics relies on minimal requirements on the base language semantics and can therefore be instantiated to arbitrary base language paradigms. We have shown how to define general aspect mechanisms from different aspect languages such as AspectJ, Caesar or Composition Filters. As an illustration of our technique, we have described the semantics of an AspectJ-like core aspect language for a core Java language.

This work is a first step towards the design of static tools to analyze the semantic impact of weaving on programs. Our mid-term goal is to statically check whether the weaving of an aspect respects a property P or ensures a property P. Properties of interest can be invariant state properties (*e.g.*, $x > 0$), temporal properties (*e.g.*, *eventually* $x > 0$, *always* $x \neq 0$) or even non functional properties (*e.g.*, the worst case execution time of method m is less than 10 times units). Recently, Shmuel Katz has characterized informally classes of aspects whose weaving respects specific classes of properties (*e.g.*, invariant, liveness, etc.). We formalize these classes of aspects using the CASB to describe their semantics. The classes of properties (*e.g.*, invariant, liveness, etc.) are formalized as subsets of temporal logics. We have started to prove formally that the weaving of some specific classes of aspects respected some specific classes of properties.

The verification and analysis of the properties of aspect-oriented programs is the central topic of Simplice Djoko Djoko's PhD thesis. This work is conducted within the Formal Methods Lab of the network of excellence AOSD-Europe (see section 8.3.2). It is done in collaboration with Rémi Douence from the OBASCO project team at École des Mines de Nantes.

6.7.2. Resource management and aspects of availability

We have studied the use of aspect-oriented programming for resource management with the aim of enforcing availability properties [15]. Our technique permits to keep the construction of systems separate from resource management and availability issues. We have focused on denials of service caused by resource management (starvations, deadlocks). Our availability aspects specify time limits in the allocation of resources. They can be seen as formal temporal properties on execution traces that specify availability policies. The different components, services and aspects are abstracted/translated into timed automata. This allows us to specify weaving as an automata product and to use model-checking tools (such as UPPAAL) to verify that aspects enforce the required availability properties.

This research, related to the DISPO project (see section 8.2.1), is part of Stéphane Hong Tuan Ha's PhD thesis from the LANDE project team at IRISA/INRIA-Rennes.

6.7.3. Fault tolerance aspects for real-time software

Here, our objective is to design an aspect language for specifying fault tolerance as well as efficient techniques based on static analysis, program transformation and/or instrumentation to weave them into real-time programs.

As a first step, we have studied the implementation of specific fault tolerance techniques in real-time embedded systems using program transformation [32][20]. The fault-intolerant initial system consists of a set of independent periodic tasks scheduled onto a set of fail-silent processors. The tasks are automatically transformed such that, assuming the availability of an additional spare processor, the resulting system tolerates one failure at a time. Failure detection is implemented using heartbeating, and failure masking using checkpointing and roll-back. These techniques are described and implemented by automatic program transformations of the tasks' source programs. The proposed formal approach to fault tolerance by program transformation highlights the benefits of separation of concerns.

The second step, is to design an aspect language allowing users to specify and tune a wider range of fault tolerance techniques. For example, the user may want to use checkpointing, code or data replication at different places of the same program. For checkpointing, the user may also want to specify the subset of variables which must be saved. The definition of an aspect language to specify such choices is under completion.

This line of research is related to the ALIDECS project (see section 8.2.2).

6.8. Other results

6.8.1. Programming models and calculi

Participant: P. Fradet.

Gamma is a formalism in which programs are expressed in terms of multiset rewriting. It is often referred to as the Chemical Reaction Model. In this formalism, the execution of a program can be seen as a solution (a multiset) of molecules reacting until the solution becomes inert.

We have proposed the language HOCL (*Higher Order Chemical Language*), a higher-order extension of Gamma that can also manipulate multisets with negative or infinite cardinalities [21][13]. The higher-order extension makes it possible to consider a chemical program as a member of a multiset, thus eligible for reactions as any other element. This facilitates the description of notions such as code mobility, distribution, adaptation, etc. The extensions of the multiset data structure, combined with the higher-order properties, provide a powerful tool for expressing general (and original) coordination schemes.

We have been working on the application of HOCL to the programming of distributed applications, in particular to autonomic systems [14] and Grid programming [22]. In a first step, applications are programmed in an abstract manner describing essentially the chemical coordination between not necessarily chemical software components. In a second step, chemical service programs are specifically provided to the run-time system in order to obtain from the resources the expected quality of service in terms of efficiency, reliability, security, etc.

An introductory abstract of this line of research has been published in the special issue of *Ercim News* on "Emergent Computing" [35]. A position paper presenting some fundamental questions about non-classical programming languages has been published in the *International Journal of Unconventional Computing* [18].

This work is conducted in collaboration with Jean-Pierre Banâtre and Yann Radenac from the PARIS project team at IRISA.

6.8.2. Modeling and compositional analysis of genetic networks

Participants: G. Goessler [contact person], A. Richard.

Proteins fulfill a huge number of functions in any living organism. Any protein is encoded by a gene. In order to produce the protein, the corresponding gene has to be *transcribed* into messenger RNA, which is then *translated* to obtain the protein. This production mechanism is regulated by the concentration of proteins, which can *promote* or *inhibit* the production, *e.g.*, by binding to the gene and disabling transcription. The dynamics of the protein concentrations are thus defined by a regulatory network which usually encompasses a multitude of complex feedback loops. Being able to model and analyze its behavior is crucial for understanding the interactions between the proteins, and their functions.

Genetic regulatory networks have been modeled as discrete transition systems by many approaches, benefiting from a large number of formal verification algorithms available. However, most of these approaches face the problem of state space explosion, as even models of modest size (from a biological point of view) usually lead to large transition systems. In practice, non-compositional approaches for the analysis of genetic regulatory networks do not scale up well.

We have explored the use of compositionality for the analysis of genetic regulatory networks. A precondition for compositional algorithms to be applicable, is that the model be structured. We have therefore defined a modeling framework for genetic regulatory networks, based on our BIP component framework, in which the different components of the system (that is, proteins or sets of proteins) and the way they constrain each other, are modeled separately and modularly. Our approach is based on a conservative approximation of the mathematically well-founded formalism of qualitative simulation [81]. This ongoing work benefits from the interaction with Hidde de Jong (HELIX project).

The task of Adrien Richard, post-doc in the POP ART and HELIX projects since October 2006, will be to define a modular but exact representation of the network behavior as defined by qualitative simulation, and study how to decompose networks into modules.

6.8.3. Interactions Between Law and Information and Communication Sciences

Participant: D. Le Métayer [contact person].

Daniel Le Métayer, who joined the POP ART team in July, is working on a new activity concerning the interactions between law and the information and communication sciences ("STIC" in French). In particular, he is studying the impact of legal aspects on the software design flow. This activity shall become an independent project team in the short term.

7. Contracts and Grants with Industry

7.1. Pôle de compétitivité Minalogic/EMSOC

In the context of the *pôle de compétitivité* EMSOC, we participate in the new four-year project OPENTLM on verification of systems-on-chip is modeled at the transaction level in SystemC [61]. We intend to develop methods for abstraction, interprocedural analysis, and compositional verification of SystemC models.

7.2. DCN

With the INRIA project team MOAIS and the ProBayes start-up, we have signed a contract with DCN. DCN is a French company based in Toulon that builds warships. We will work on a R&D project aimed at improving the defense embedded software of their next generation warships.

7.3. PolySpace

A contract with POLYSPACE TECHNOLOGIES started at the end of the year. The collaboration concerns their main product, POLYSPACE, a static analyser for detecting possible run-time errors in C/C++/ADA programs, which is based on abstract interpretation techniques.

8. Other Grants and Activities

8.1. Regional actions

8.1.1. Local Arc C^3O

C^3O (Conception Conjointe Commande Ordonnancement) is a locally funded (by INRIA-Rhône-Alpes) cooperation with LAG about control/scheduling co-design. It supports research on feedback scheduling together with the development of dedicated software tools.

8.2. National actions

8.2.1. ACI "Sécurité et informatique" Dispo: disponibility of software

Participant: P. Fradet.

The DISPO project[32] is concerned with specifying, verifying and enforcing security policies governing the availability of services offered by software components. The consortium includes École des Mines de Nantes, INRIA (Rennes and Rhône-Alpes), IRIT (Toulouse) and ENST-Bretagne. We are interested in weaving-like techniques for enforcing availability properties on software components. The project has ended in October 2006.

8.2.2. ACI "Sécurité & Informatique" Alidecs: integrated development environment for safe embedded components

Participants: P. Fradet, A. Girault, G. Goessler, M. Tivoli.

The objective of the ALIDECS project[33] is to study an integrated development environment for the construction and use of safe embedded components. The consortium includes LRI (Orsay), INRIA (Rhône-Alpes and Sophia Antipolis), VERIMAG (Grenoble) and LAMI (Evry). We have proposed weaving-like techniques for enforcing fault tolerance properties to reactive systems. We have also studied an approach to automatically synthesize adapters in order to assemble off-the-shelf real-time components.

8.2.3. ACI "Sécurité et informatique" Apron: analysis of numerical programs

Participant: B. Jeannet.

The APRON (Analyse de PROgrammes Numériques) project (http://www.cri.ensmp.fr/apron/) [2004-2006] involves ENSMP, LIENS-ENS, LIX-Polytechnique, VERIMAG and VERTECS-IRISA.

The goal of the project is the static analysis of large specifications (*e.g.*, à la LUSTRE) and corresponding C programs, involving a lot of numerical floating-point computations, as well as boolean and counter-based control in order to verify critical properties (including the detection of possible runtime errors), and to help in automatically locating the origin of critical property potential violation.

An example of such critical properties, as found in control/command programs, is of the form "under a condition holding on boolean and numerical variables for some time, the program must imperatively establish a given boolean and/or numerical property, in given bounded delay".

POP ART contributes to the following topics within the APRON project:

- The design and implementation of a common interface to several abstraction libraries (intervals, linear equalities, octagons, polyhedra, ...and their combination).
- The verification of LUSTRE specifications with adaptive techniques, using the NBAC tool as an experimental platform.

In 2006, most of the effort of VERTECS was spent on the implementation of the common interface.

8.2.4. CNRS RTP 21: fault tolerance

We are collaborating to this RTP entitled *Sûreté de fonctionnement des systèmes informatiques complexes ouverts*[34].

8.2.5. CNRS RTP 55: Network controlled systems

NECS (NEtworked Control Systems)[35] is a research project funded by the CNRS (STICS department) in the framework of multi-labs projects. It intends to address problems and treat topics where control and communication theory interacts with information theory, such as control systems distributed over the nodes of a fieldbus. It currently gathers people from LAG, INRIA and LIS (Laboratoire des Images et Signaux).

[32] http://www.irisa.fr/lande/jensen/dispo.html
[33] http://www-verimag.imag.fr/SYNCHRONE/alidecs/
[34] http://www.laas.fr/RTP21-SdF
[35] http://www-lag.ensieg.inpg.fr/canudas/necs.htm

8.2.6. ARA-SSIA Safe_NECS

SAFE_NECS is an « Action de Recherche Amont - Sécurité, Systèmes embarqués et Intelligence Ambiante » funded for three years by the ANR and started in January 2006 http://safe-necs.cran.uhp-nancy.fr/. The research topic is fault tolerant control of distributed process and the project focuses on both diagnosis and robust control under execution resources constraints. It gathers teams from CRAN and LORIA (Nancy), LAAS (Toulouse), and LAG and POP ART (Grenoble).

8.2.7. Collaborations inside Inria

- The SED service at INRIA-Rhône-Alpes is maintaining ORCCAD and provides support for experiments within the C^3O ARC.
- AOSTE at INRIA-Rocquencourt is working with us on fault tolerant heuristics for their software SYNDEX.
- VERTECS at IRISA/INRIA-Rennes is working with us on applications of discrete controller synthesis, and in particular on the tool SIGALI [36].
- P. Fradet cooperates with S. Hong Tuan Ha (LANDE, IRISA/INRIA-Rennes), with J.-P. Banâtre and Y. Radenac (PARIS, IRISA/INRIA-Rennes) and with R. Douence and M. Südholt (OBASCO, Ecole des Mines de Nantes).
- A. Girault cooperates with the MOAIS project (UR Rhône-Alpes) on multi-criteria scheduling. In particular, we have a common industrial contract with DCN. A. Girault cooperates also with the VERIMAG lab on model-based design and a compilation tool chain from SIMULINK to distributed platforms, and with the DEMON team of LRI (Orsay) on the distribution of higher-order synchronous data-flow programs.
- G. Goessler cooperates with H. de Jong (HELIX project, UR Rhône-Alpes) and F. Lang (VASY project, UR Rhône-Alpes).
- B. Jeannet cooperates with T. Le Gall (VERTECS, IRISA/INRIA-Rennes) on the analysis of communicating systems, and with C. Constant, T. Jéron and F. Ployette (VERTECS, IRISA/INRIA-Rennes) on test generation.
- E. Rutten is working with the DART project at UR Futurs in Lille, on the synchronous modelling of massively parallel application, and the introduction of control and mode automata in the GASPARD framework.

8.2.8. Cooperations with other laboratories

- A. Girault cooperates with X. Nicollin (VERIMAG), M. Pouzet (LRI, University of Paris VI), D. Trystram and É. Saule from (ID-IMAG), and C. Dima (Université of Paris XII).
- G. Goessler cooperates with J. Sifakis and S. Graf (VERIMAG) and M. Majster-Cederbaum (University of Mannheim, Germany).
- B. Jeannet cooperates with N. Halbwachs and L. Gonnord (VERIMAG) on the static analysis of numerical variables.
- D. Simon cooperates with O. Sename (LAG).
- E. Rutten cooperates with H. Alla (LAG).

8.3. European actions

8.3.1. Artist II European IST network of Excellence

ARTIST II is a European Network of Excellence on embedded system design[36]. Its goal is to establish Embedded Systems Design as a discipline, combining expertises from electrical engineering, computer science, applied mathematics, and control theory. We collaborate as a core partner within the Real Time Components cluster, led by A. Benveniste (INRIA Rennes) and B. Jonsson (Uppsala University). A. Girault is the administrator of ARTIST II for INRIA.

[36] http://www.artist-embedded.org/FP6

8.3.2. AOSD European IST network of Excellence

AOSD-Europe is the European network of excellence on Aspect-Oriented Software Development. It lasts 4 years (September 2004-August 2008) and includes nine major academic institutions and two major industrial partners from UK, Germany, The Netherlands, France, Belgium, Ireland, Spain and Israel. We collaborate in the formal methods lab with OBASCO-INRIA, Technion (Israel), and Twente (The Netherlands).

9. Dissemination

9.1. Scientific community

- P. Fradet has participated in the program committee of FOAL'06 (*Foundations of Aspect-Oriented Languages Workshop*). He was co-editor of the special issue of Science of Computer Programming on *Foundations of aspect-oriented programming* published by Elsevier in december 2006 [11]. He has given a course with Jean-Pierre Banâtre on Chemical Programming at *Ecole des Jeunes Chercheurs en Programmation*, Luchon, juin 2006.
- A. Girault serves as associate editor for the *Eurasip Journal on Embedded Systems*. He has co-edited two Special Issues for this journal, on Formal Methods for GALS Systems, and on Synchronous Paradigm for Embedded Systems. He has organized the *International Open Workshop on Synchronous Programming*, and he maintains the *SYNchronous Applications, Languages, and Programs* web site[37].
- D. Simon is a member of the RTNS'06 and RTNS'07 (international conference on real-time and network systems) program committee. He has been rapporteur for the PhD of M. Ben Gaïd at ESIEE (Evry) about *Optimal scheduling and control for distributed real-time systems*.
- E. Rutten is co-editor of the special issue of Discrete Event Dynamical Systems (jDEDS) on Control and Modeling of Reactive Systems. He was member of the PhD committees of A. Kerbaa (CERIMAG, Grenoble) as a referee, and of O. Labbani (LIFL/INRIA Futurs, Lille).

9.2. Teaching

9.2.1. Courses

- Alain Girault and Daniel Simon: *Real-time and reactive programming*, 18h, Master of Science IVR, INPG.
- Alain Girault: *Algorithmics and programming in Java*, 26h, INPG Telecom Department.
- Gregor Goessler: *Software Engineering and Compilation project*, 2nd year engineering, 55h, INPG / ENSIMAG.
- Alain Girault and Pascal Raymond: *Synchronous programming*, 28h, Master of Science, Université Joseph Fourier.
- Daniel Le Métayer: *Systematic security analysis*, at the FOSAD'06 conference (Foundations of Software analysis and Design), september 2006; and at the Ecole des Mines de Nantes, december 2006. 7h.

9.2.2. Advising

PhDs:

- Gwenaël Delaval, co-advised by Alain Girault (with M. Pouzet, LRI Orsay), since 9/2004. PhD in computer science, INPG.

[37] http://www.synalp.org

- Tristan Le Gall, co-advised by Bertrand Jeannet (with T. Jéron, VERTECS IRISA) since 9/2004. PhD in computer science, University of Rennes I.
- Simplice Djoko Djoko, co-advised by P. Fradet (with R. Douence, OBASCO, Ecole des Mines de Nantes), since 9/2005, PhD in computer science, University of Nantes.
- Stéphane Hong Tuan Ha, advised by Pascal Fradet, since 9/2002, PhD in computer science, University of Rennes I.
- Yann Radenac, co-advised by P. Fradet (with J.-P. Banâtre, IRISA), since 9/2003, PhD in computer science, University of Rennes I.
- David Robert, co-advised by Daniel Simon (with O. Sename, LAG Grenoble), since 9/2003, PhD in control theory, INPG.
- Mouaiad Alras, co-advised by Alain Girault (with P. Raymond, VERIMAG Grenoble), since 10/2006, PhD in computer science, UJF, Grenoble.
- Gérald Vaisman, co-advised by Alain Girault (with P.-F. Dutot, MOAIS UR Rhône-Alpes), since 10/2006, PhD in computer science, INPG.
- Huafeng Yu, co-advised by E. Rutten (with J.-L. Dekeyser, LIFL/INRIA Futurs Lille), since 10/2005. PhD in computer science, University of Lille 1.

Masters:

- Abdul Malik Khan, co-advised by G. Goesler (with F. Lang, VASY project), in 2005/2006. Master of Science in computer science, UJF, Grenoble.

10. Bibliography

Major publications by the team in recent years

[1] K. ALTISEN, A. CLODIC, F. MARANINCHI, É. RUTTEN. *Using Controller-Synthesis Techniques to Build Property-Enforcing Layers*, in "Proceedings of the European Symposium on Programming, ESOP'03", Lecture Notes in Computer Science (LNCS), n^{o} 2618, Springer Verlag, April 2003, p. 174–188.

[2] K. ALTISEN, G. GÖSSLER, J. SIFAKIS. *Scheduler Modeling Based on the Controller Synthesis Paradigm*, in "Journal of Real-Time Systems, special issue on "control-theoretical approaches to real-time computing"", vol. 23, n^{o} 1/2, 7-9 2002, p. 55–84.

[3] I. ASSAYAD, A. GIRAULT, H. KALLA. *A Bi-Criteria Scheduling Heuristics for Distributed Embedded Systems Under Reliability and Real-Time Constraints*, in "International Conference on Dependable Systems and Networks, DSN'04, Firenze, Italy", IEEE, June 2004, p. 347–356.

[4] J.-J. BORRELLY, E. COSTE MANIÈRE, B. ESPIAU, K. KAPELLOS, R. PISSARD-GIBOLLET, D. SIMON, N. TURRO. *The Orccad Architecture*, in "International Journal on Robotic Research", vol. 17, n^{o} 4, 1998, p. 338–359.

[5] P. CASPI, A. GIRAULT, D. PILAUD. *Automatic Distribution of Reactive Systems for Asynchronous Networks of Processors*, in "IEEE Trans. on Software Engineering", vol. 25, n^{o} 3, May 1999, p. 416–427.

[6] T. COLCOMBET, P. FRADET. *Enforcing trace properties by program transformation*, in "Proc. of Principles of Programming Languages, Boston", ACM Press, January 2000, p. 54-66.

[7] P. FRADET. *Approches langages pour la conception et la mise en œuvre de programmes*, Habilitation thesis, Université de Rennes 1, November 2000.

[8] P. FRADET, S. HONG TUAN HA. *Network Fusion*, in "Proceedings of Asian Symposium on Programming Languages and Systems (APLAS'04)", LNCS, vol. 3302, Springer-Verlag, November 2004, p. 21–40.

[9] A. GIRAULT, H. KALLA, M. SIGHIREANU, Y. SOREL. *An Algorithm for Automatically Obtaining Distributed and Fault-Tolerant Static Schedules*, in "International Conference on Dependable Systems and Networks, DSN'03, San-Francisco (CA), USA", IEEE, June 2003.

[10] G. GÖSSLER, J. SIFAKIS. *Priority Systems*, in "proc. FMCO'03", F. DE BOER, M. BONSANGUE, S. GRAF, W.-P. DE ROEVER (editors). , LNCS, vol. 3188, Springer-Verlag, 2004, p. 314-329.

Year Publications

Books and Monographs

[11] P. FRADET, R. LÄMMEL (editors). *Science of Computer Programming - Special Issue on Foundations of Aspect-Oriented Programming*, Elsevier, december 2006.

Doctoral dissertations and Habilitation theses

[12] A. GIRAULT. *Contributions à la Conception Sûre des Systèmes Embarqués Sûrs*, Habilitation thesis, INPG, Grenoble, France, September 2006, ftp://ftp.inrialpes.fr/pub/bip/pub/girault/Publications/HDR06/main.pdf.

Articles in refereed journals and book chapters

[13] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Generalised multisets for chemical programming*, in "Mathematical Structures in Computer Science", vol. 16, n^{o} 4, August 2006, p. 557–580.

[14] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Programming Self-Organizing Systems with the Higher-Order Chemical Language*, in "International Journal of Unconventional Computing", 2006.

[15] P. FRADET, S. HONG TUAN HA. *Systèmes de gestion de ressources et aspects de disponibilité*, in "L'Objet - Logiciel, bases de données, réseaux", (In French), vol. 12, n^{o} 2-3, September 2006, p. 183–210.

[16] A. GIRAULT, X. NICOLLIN, M. POUZET. *Automatic Rate Desynchronization of Embedded Reactive Programs*, in "ACM Trans. on Embedded Computing Systems", vol. 5, n^{o} 3, August 2006, p. 687–717.

[17] T. LEGALL, B. JEANNET, H. MARCHAND. *Contrôle de systèmes symboliques, discrets ou hybrides*, in "Technique et Science Informatiques", vol. 25, n^{o} 3, 2006.

[18] O. MICHEL, J.-P. BANÂTRE, P. FRADET, J.-L. GIAVITTO. *Challenging Questions for the Rationals of Non-Classical Programming Languages*, in "International Journal of Unconventional Computing", (to appear), 2006.

[19] D. SIMON, O. SENAME, D. ROBERT. *Systèmes Temps Réel Tome II : Ordonnancement, Réseaux, Qualité de Service*, vol. 2, chap. Conception conjointe commande/ordonnancement et ordonnancement régulé, Hermes, 2006.

Publications in Conferences and Workshops

[20] T. AYAV, P. FRADET, A. GIRAULT. *Implementing Fault-Tolerance in Real-Time Systems by Program Transformations*, in "Proceedings of the Sixth ACM & IEEE International Conference on Embedded Software, EMSOFT'06", 2006, p. 205–214.

[21] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *A Generalized Higher-Order Chemical Computation Model with Infinite and Hybrid Multisets*, in "Proceedings of 1st International Workshop on New Developments in Computational Models (DCM'05)", ENTCS, vol. 135(3), Elsevier, March 2006, p. 3–13.

[22] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Towards Grid Chemical Coordination (short paper)*, in "Proceedings of the 2006 ACM Symposium on Applied Computing (SAC'06)", ACM press, April 2006, p. 445–446.

[23] G. DELAVAL, E. RUTTEN. *A domain-specific language for multi-task systems, applying discrete controller synthesis*, in "Proceedings of the 21st ACM Symposium on Applied Computing, SAC 2006, Special Track on Embedded Systems: Applications, Solutions, and Techniques, Dijon, France, April 23-27 2006", 2006, p. 901–905.

[24] G. DELAVAL, E. RUTTEN. *A Domain-specific Language for Task Handlers Generation, Applying Discrete Controller Synthesis*, in "SAC'06: Proceedings of the 2006 ACM Symposium on Applied computing", ACM Press, April 2006, p. 901–905.

[25] A. GIRAULT, H. KALLA, Y. SOREL. *Transient Processor/Bus Fault Tolerance for Embedded Systems*, in "IFIP Working Conference on Distributed and Parallel Embedded Systems, DIPES'06, Braga, Portugal", Springer, October 2006, p. 135–144.

[26] A. GIRAULT, H. YU. *A Flexible Method to Tolerate Value Sensor Failures*, in "International Conference on Emerging Technologies and Factory Automation, ETFA'06, Prague, Czech Republic", IEEE, September 2006, p. 86–93.

[27] K. KAPELLOS, D. SIMON, R. PISSARD-GIBOLLET, B. ESPIAU. *The Orccad Robot Control Architecture and Tools in Space Applications*, in "9th ESA Workshop on Advanced Space Technologies for Robotics and Automation ASTRA'06, Noordwijk, The Netherlands", ESTEC, november 28-29-30 2006.

[28] T. LEGALL, B. JEANNET, T. JÉRON. *Verification of Communication Protocols Using Abstract Interpretation of FIFO queues*, in "Algebraic Methodology and Software Technology, AMAST '06", LNCS, vol. 4019, July 2006.

[29] D. ROBERT, O. SENAME, D. SIMON. *Synthesis of a sampling period dependent controller using LPV approach*, in "5th IFAC Symposium on Robust Control Design ROCOND'06, Toulouse, France", july 5-7 2006.

[30] D. SIMON, R. PISSARD-GIBOLLET, S. ARIAS. *Orccad, a framework for safe robot control design and implementation*, in "1st National Workshop on Control Architectures of Robots: software approaches and issues CAR'06, Montpellier", april 6-7 2006.

[31] D. SIMON, D. ROBERT, O. SENAME. *Control and Real-time Scheduling Co-design : Application to Robust Robot Control*, in "3rd Taiwanese-French Conference on Information Technology TFIT'06, Nancy", march 2006.

Internal Reports

[32] T. AYAV, P. FRADET, A. GIRAULT. *Implementing Fault-Tolerance in Real-Time Systems by Program Transformations*, Research Report, n^{o} 5919, INRIA, may 2006, http://hal.inria.fr/inria-00077156.

[33] G. GÖSSLER. *Componend-based Design of Heterogeneous Reactive Systems in* PROMETHEUS, Research Report, n^{o} 6057, INRIA, 2006, https://hal.inria.fr/inria-00119245.

[34] M. TÖRNGREN, D. HENRIKSSON, O. REDELL, C. KIRSCH, J. EL-KHOURY, D. SIMON, Y. SOREL, H. ZDENEK, K.-E. ÅRZÉN. *Co-design of Control Systems and their real-time implementation - A Tool Survey*, Technical report, n^{o} TRITA - MMK 2006:11, Royal Institute of Technology, KTH, Stockolm, 2006.

Miscellaneous

[35] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Chemical Programming of Self-Organizing Systems*, vol. 64, January 2006, http://www.ercim.org/publication/Ercim_News/enw64.

[36] L. BESNARD, H. MARCHAND, E. RUTTEN. *The Sigali Tool Box Environment*, July 2006, Tools session, Workshop on Discrete Event Systems, WODES'06.

[37] R. DOUENCE, S. DJOKO DJOKO, P. FRADET, D. LE BOTLAN. *Towards a Common Aspect Semantic Base (CASB)*, Deliverable 54, AOSD-Europe, EU Network of Excellence in AOSD, august 2006.

[38] A. GIRAULT. *System-Level Design of Fault-Tolerant Embedded Systems*, vol. 67, October 2006.

[39] A. KHAN. *Connection of Compositional Verification Tools for Embedded Systems*, Technical report, UJF Grenoble, 2006.

[40] D. ROBERT, O. SENAME, D. SIMON. *Co-design commande/ordonnancement pour le contrôle sous contraintes de ressources*, june 2006, Poster - Colloque EmSoC-Recherche.

References in notes

[41] P. APKARIAN, P. GAHINET, G. BECKER. *Self-scheduled H_∞ Control of Linear Parameter-varying Systems: A Design Example*, in "Automatica", vol. 31, n^{o} 9, 1995, p. 1251–1262.

[42] A. ARNOLD. *Systèmes de transitions finis et sémantique des processus communicants*, Masson, 1992.

[43] E. ASARIN, O. BOURNEZ, T. DANG, O. MALER, A. PNUELI. *Effective Synthesis of Switching Controllers of Linear Systems*, in "Proceedings of the IEEE", vol. 88, 2000, p. 1011–1025.

[44] J.-R. BEAUVAIS, E. RUTTEN, T. GAUTIER, R. HOUDEBINE, P. LE GUERNIC, YAN-MEI. TANG. *Modelling Statecharts and Activity Charts as Signal Equations*, in "ACM Transactions on Software Engineering and Methodology", vol. 10, n^{o} 4, October 2001, p. 397–451.

[45] R. BRYANT. *Graph-based algorithms for boolean function manipulation*, in "IEEE Transactions on Computers", vol. C-35, n^{o} 8, 1986, p. 677–692.

[46] CEI (COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE). *Norme Internationale – Automates programmables : Langages de programmation*, Technical report, n^o IEC 1131 partie 3, CEI/IEC (International Electrotechnical Commission), 1993.

[47] P. CASPI, M. POUZET. *Synchronous Kahn networks*, in "ICFP '96: Proceedings of the first ACM SIGPLAN international conference on Functional programming, New York, NY, USA", ACM Press, 1996, p. 226–238, http://doi.acm.org/10.1145/232627.232651.

[48] P. CASPI, M. POUZET. *Lucid Synchrone: une extension fonctionnelle de Lustre*, in "Journées Francophones des Langages Applicatifs (JFLA)", INRIA, Feb 1999.

[49] C. CASSANDRAS, S. LAFORTUNE. *Introduction to Discrete Event Systems*, Kluwer, 1999.

[50] A. CERVIN, J. EKER, B. BERNHARDSSON, K.-E. ARZEN. *Feedback-Feedforward Scheduling of Control Tasks*, in "Real Time Systems", vol. 23, n^o 1, 2002, p. 25–54.

[51] D. CHASE, M. WEGMAN, F. ZADECK. *Analysis of Pointers and Structures*, in "Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation", ACM Press, 1990, p. 296–310.

[52] E. CLARKE, E. EMERSON, A. SISTLA. *Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications*, in "ACM Transactions on Programming Languages and Systems", vol. 8, n^o 2, 1986, p. 244-263.

[53] D. CLARKE, T. JÉRON, V. RUSU, E. ZINOVIEVA. *STG: a Symbolic Test Generation tool*, in "(Tool paper) Tools and Algorithms for the Construction and Analysis of Systems (TACAS'02)", LNCS, vol. 2280, 2002.

[54] J.-L. COLAÇO, A. GIRAULT, G. HAMON, M. POUZET. *Towards a Higher-Order Synchronous Data-Flow Language*, in "4th International Conference on Embedded Software, EMSOFT'04, Pisa, Italy", G. BUTTAZZO (editor). , ACM, September 2004, ftp://ftp.inrialpes.fr/pub/bip/pub/girault/Publications/Emsoft04/.

[55] P. COUSOT, R. COUSOT. *Abstract Interpretation and Application to Logic Programs*, in "Journal of Logic Programming", vol. 13, n^o 2–3, 1992, p. 103–179.

[56] P. COUSOT, N. HALBWACHS. *Automatic discovery of linear restraints among variables of a program*, in "5th ACM Symposium on Principles of Programming Languages, POPL'78, Tucson (Arizona)", January 1978.

[57] P. D'ARGENIO, B. JEANNET, H. JENSEN, K. LARSEN. *Reduction and Refinement Strategies for Probabilistic Analysis*, in "Process Algebra and Probabilistic Methods - Performance Modelling and Verification, PAPM-PROBMIV'02, Copenhagen (Denmark)", LNCS, vol. 2399, July 2002.

[58] E. DUMITRESCU, A. GIRAULT, E. RUTTEN. *Validating Fault-Tolerant Behaviors of Synchronous System Specifications by Discrete Controller Synthesis*, in "IFAC Workshop on Discrete Event Systems, WODES'04, Reims, France", September 2004, ftp://ftp.inrialpes.fr/pub/bip/pub/girault/Publications/Wodes04/.

[59] F. GAUCHER, E. JAHIER, B. JEANNET, F. MARANINCHI. *Automatic State Reaching for Debugging Reactive Programs*, in "5th Int. Workshop on Automated and Algorithmic Debugging, AADEBUG'03", September 2003.

[60] A. GIRAULT, É. RUTTEN. *Discrete Controller Synthesis for Fault-Tolerant Distributed Systems*, in "Proceedings of the Ninth International Workshop on Formal Methods for Industrial Critical Systems, FMICS 04", Tech. Rep of Kepler University Linz & ENTCS Eslevier, September 2004.

[61] T. GRÖTKER, S. LIAO, G. MARTIN, S. SWAN. *System Design with SystemC*, Kluwer, 2002.

[62] G. GÖSSLER, J. SIFAKIS. *Composition for Component-based Modeling*, in "Science of Computer Programming", vol. 55, n^{o} 1-3, 2005, p. 161-183.

[63] G. GÖSSLER. PROMETHEUS — *A Compositional Modeling Tool for Real-Time Systems*, in "Proc. Workshop RT-TOOLS'01", P. PETTERSSON, S. YOVINE (editors). , Technical report 2001-014, Uppsala University, Department of Information Technology, 2001.

[64] G. GÖSSLER, J. SIFAKIS. *Priority Systems*, in "proc. FMCO'03", F. DE BOER, M. BONSANGUE, S. GRAF, W.-P. DE ROEVER (editors). , LNCS, vol. 3188, Springer-Verlag, 2004, p. 314-329.

[65] N. HALBWACHS. *Synchronous Programming of Reactive Systems*, Kluwer, 1993.

[66] N. HALBWACHS. *Synchronous Programming of Reactive Systems – a Tutorial and Commented Bibliography*, in "Proc. of the Int. Conf. on Computer-Aided Verification, CAV'98, Vancouver, Canada", LNCS Vol. 1427, Springer-Verlag, 1998.

[67] D. HAREL. *Statecharts: A Visual Formalism for Complex Systems*, in "Science of Computer Programming", vol. 8, 1987, p. 231-274.

[68] B. JEANNET, P. D'ARGENIO, K. LARSEN. *RAPTURE: A tool for verifying Markov Decision Processes*, in "Tools Day, International Conference on Concurrency Theory, CONCUR'02, Brno (Czech Republic)", Technical Report, Faculty of Informatics at Masaryk University Brno, August 2002.

[69] B. JEANNET. *Dynamic Partitioning In Linear Relation Analysis. Application To The Verification Of Reactive Systems*, in "Formal Methods in System Design", vol. 23, n^{o} 1, July 2003, p. 5–37.

[70] B. JEANNET, T. JÉRON, V. RUSU, E. ZINOVIEVA. *Symbolic Test Selection based on Approximate Analysis*, in "11th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'05), Edinburgh (UK)", LNCS, vol. 3440, April 2005.

[71] C. LU, J.-A. STANKOVIC, G. TAO, S.-H. SON. *Feedback Control Real-Time Scheduling: Framework, Modeling, and Algorithms*, in "Real Time Systems", vol. 23, n^{o} 1, 2002, p. 85–126.

[72] O. MALER, A. PNUELI, J. SIFAKIS. *On the Synthesis of Discrete Controllers for Timed Systems*, in "Proc. of STACS'95", LNCS, vol. 900, Springer Verlag, 1995.

[73] F. MARANINCHI, Y. RÉMOND. *Mode-Automata: a new Domain-Specific Construct for the Development of Safe Critical Systems*, in "Science of Computer Programming", vol. 46, n^{o} 3, March 2003, p. 219-254.

[74] H. MARCHAND, P. BOURNAI, M. LE BORGNE, P. LE GUERNIC. *Synthesis of Discrete-Event Controllers based on the Signal Environment*, in "Discrete Event Dynamical System: Theory and Applications", vol. 10, n^{o} 4, October 2000, p. 325–346.

[75] J.-P. QUEILLE, J. SIFAKIS. *Specification and Verification of Concurrent Systems in CESAR*, in "proc. International Symposium on Programming", LNCS, vol. 137, Springer-Verlag, 1982, p. 337-351.

[76] P. J. RAMADGE, W. M. WONHAM. *Supervisory control of a class of discrete event processes*, in "SIAM J. Control Optim.", vol. 25, n^{o} 1, 1987, p. 206–230.

[77] P. J. RAMADGE, W. M. WONHAM. *The Control of Discrete Event Systems*, in "Proceedings of the IEEE", vol. 77, n^{o} 1, 1989.

[78] D. ROBERT. *Contribution à l'interaction commande/ordonnancement*, Ph. D. Thesis, INPG, Grenoble, France, janvier 2007.

[79] D. ROBERT, O. SENAME, D. SIMON. *A reduced polytopic LPV synthesis for a sampling varying controller : experimentation with a T inverted pendulum*, in "European Control Conference ECC'07", submitted, 2007.

[80] H. TOPCUOGLU. *Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Scheduling*, in "IEEE Trans. on Parallel and Distributed Systems", vol. 13, n^{o} 3, March 2002, p. 260–274.

[81] H. DE JONG, J.-L. GOUZÉ, C. HERNANDEZ, M. PAGE, T. SARI, J. GEISELMANN. *Qualitative Simulation of Genetic Regulatory Networks Using Piecewise-Linear Models*, in "Bulletin of Mathematical Biology", vol. 66, 2004, p. 301–340.