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]

Project assistants
Marie Wiernsperger from 01/005 to 10/005
Stéphanie Berger since 11/005

Inria permanent researchers
Pascal Fradet [CR]
Gregor Goessler [CR]
Daniel Simon [CR]

PhD students
Gwenaël Delaval [MENRT grant]
Hamoudi Kalla [INRIA-EEA grant, until 02/005]
David Robert [MENRT EEATS grant]
Simplice Djoko Djoko [INRIA grant since 10/005]

Post-doctoral fellows
Tolga Ayav [MENRT/INRIA post-doctoral grant]
Yvan Roux [MINEFI NANO 2008 grant, since 1/2005]
Massimo Tivoli [MENRT/INRIA post-doctoral grant, since 10/2005]

Interns
Darina Dimitrova [July–September, Technical University-Sofia, Bulgaria]
Erik Saul [March–June, M2R Grenoble]
Nour Brinis [June–July, Ensi Tunis]
Huafeng Yu [March–June, M2R Grenoble]
Sumit Kumar [June–July, IIT Dehli, India]
Abdul-Malik Khan [since October 15, M2R Grenoble, co-advised with the VASY team]

External Partners
Emil Dumitrescu [INSA Lyon]
Olivier Sename [LAG, INPG-ENSIEG]

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. 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
[3] http://www-rocq.inria.fr/syndex

- control/scheduling co-design, with cross-interactions between techniques of serving 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.

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:

- the ITEA European project EAST-EEA[7], about embedded electronics in cars,
- the IST European networks of excellence:
  - ARTIST 2 [8], about advanced real-time systems,
  - AOSD-Europe[9], about formal methods for Aspect-Oriented Programming,
- two ACIs (actions concertées incitatives), Alidecs (on large-scale critical embedded systems) and Dispo (on security policies for software components),
- collaborations with STMicroelectronics and France Télécom R&D.

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.

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,

[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.east-eea.net/
[8] http://www.artist-embedded.org/FP6/Overview/
[9] http://www.aosd-europe.net/

internationally and in particular, in the European IST network of excellence ARTIST 2 (Advanced Real-Time Systems)[10].

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 [34], [40]. During the past years, methodologies for the formal verification [58], [42], control synthesis [60] and compilation, and extensions to timed and hybrid systems [53], [35] 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 [50], which can be encoded in a synchronous model as shown in [36].

The synchronous approach[11] [48], [49] to reactive systems design gave birth to complete programming environments, around languages like ARGOS, LUSTRE[12], ESTEREL[13], SIGNAL/ POLYCHRONY[14], SYNDEX[15], Lucid Synchrone[16] or Mode Automata[17]. 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 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 [41], [52]; 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

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

(continuous or discrete), and the distance between the various theoretical concepts and results which allow the study of different aspects of their behaviors, and the design of controllers. The European IST network of excellence ARTIST 2 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 [37]) 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) which 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 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.

3.3.2. Main Directions

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;
- Aspect-Oriented Programming that allows to express safety concerns separately from the functional part and to enforce them on program.

3.3.3. 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.

3.3.4. 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.

3.3.5. Automatic generation of correct controllers

We use techniques of discrete controller synthesis, especially the tools SIGALI [55] and Mode Automata [54] 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.

4. Application Domains

4.1. Application Domains

Keywords: *automotive*, *embedded systems*, *robotics*, *telecommunications*.

4.1.1. 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).

4.1.2. Industrial design tools.

The commercially available design tools (such as UML with real-time extensions, Matlab/Simulink/dSPACE[18]) 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[19], SILDEX[20] (based on SIGNAL), specialized environments like CELLCONTROL for industrial automatisms (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 European project TELEDIMOS), or by distribution in the context of free software around the real-time/embedded versions of Linux[21].

4.1.3. Current industrial cooperations.

Regarding applications and case studies with industrial end-users of our techniques, we cooperate with STMicroelectronics on compositional verification for System-on-Chip design assistance.

[18] http://www.dspaceinc.com
[19] http://www.esterel-technologies.com
[20] http://www.tni-valiosys.com
[21] http://www.realtimelinuxfoundation.org/projects/projects.html

5. Software

5.1. Orccad

Participants: S. Arias, D. Simon [contact person].

ORCCAD [22] 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 are developed jointly by the *SED* service and the researchers of the Pop Art team. The current stable version allows for the automatic generation of real-time single-rate controllers running on top of VxWorks, Solaris and Linux. The main current developments allow for the generation of multi-rate controllers and the use of feedback scheduling running on top of Linux/RTAI (hard real-time) or patched Linux kernels (soft real-time using the Posix API).

5.2. Implementations of synchronous programs

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

5.2.1. 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.

Currently our software OCREP is distributed in the form of executable on the web[23]. It consists in 15000 lines of C++ code. 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.

5.2.2. Fault-tolerance

We have been cooperating for several years with the INRIA team AOSTE on the subject of fault-tolerance. In particular, we have implemented several new heuristics for fault-tolerance and reliability within their software SYNDEX[24]. This has taken place within the framework of the European project EAST-EEA in which we participate together with AOSTE.

5.3. Prototypes

5.3.1. 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 (developed at IRISA) and of Mode Automata (developed at VERIMAG [25]).

[22] http://www.inrialpes.fr/iramr/pub/Orccad
[23] http://pop-art.inrialpes.fr/people/girault/Ocrep/
[24] http://www-rocq.inria.fr/syndex
[25] http://www-verimag.imag.fr

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 6.4.2).

5.3.2. Compositionality

Participants: G. Goessler [contact person], Y. Roux.

Further results for compositional modeling, verification and synthesis (section 6.5) have been implemented in the prototype tool PROMETHEUS, in order to perform case studies to evaluate their potential and limits.

Y. Roux has developed a prototype tool translating models of Systems-on-Chip written in SystemC, into the input format of Prometheus. This prototype is currently being tested, and is intended to serve as a module for a tool platform for compositional verification of Systems-on-Chip.

6. New Results

6.1. Higher-order synchronous data-flow programming

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

Software-defined radio has recently emerged as an important research area for mobile telephone operators. The basic functionalities that both the emitter (*e.g.*, the base station, the wireless network hub, ...) and the receiver (*e.g.*, the cell phone terminal, the PDA, ...) must run are digital to analog conversion, analog to digital conversion, modulation/demodulation, radio frequency conversion, and so on. Software radio means that these functionalities are implemented as *software* modules run on general purpose hardware. For instance, this could allow a mobile terminal to adapt seamlessly to its environment, for instance when moving from a UMTS zone to a WIFI zone. Most of the existing approaches are either based on asynchronous process calculi or on middleware. In this context, we have proposed an evolution of the synchronous language LUCID SYNCHRONE designed by M. Pouzet and P. Caspi [38], [39]. This new language, called DECADE [43], offers dynamic higher-order features, whereas LUCID SYNCHRONE only had static higher-order. Concretely, DECADE allows a function `f` to be parametrized by another function g (higher-order), and more important to replace during the execution g by another function `h` (dynamic higher-order). It is a data-flow language, so all the objects handled by a program are streams, that is, infinite sequences of typed data. Higher-order means that both the data and the functions are streams. To this respect, DECADE is the first purely data-flow programming language. Hence one can define streams of functions of streams. This feature makes DECADE a programming language well suited for software-defined radio. Gwenaël Delaval is doing a PhD on this topic, co-advised by Marc Pouzet from LRI (University of Orsay) and Alain Girault. He works in the context of the ALIDECS ACI [26].

6.1.1. Distribution of high-order synchronous dataflow programs

We are currently studying a synchronous dataflow language, LUCID SYNCHRONE [38]. This language is a high-order dataflow language, where functions are first-class citizens, i.e., can be manipulated as values, for instance as function parameters or results.

We propose to extend this language with primitives allowing the programmer to express the *location* of streams. The goal is then 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 location informations, will be the same as the program without the location annotations.

This work is based on [3], where, given a reactive program, the location of each node of this program is propagated from the locations of its inputs and outputs. This “coloring” process is made on an in-lined model of the program. Unfortunately, this does not work for high-order programs, as such programs cannot be, in general, in-lined in order to perform a semantic computation, such as the distribution process described above.

[26] http://www-verimag.imag.fr/SYNCHRONE/alidecs/

Therefore, a "spatial" type system is proposed to infer locations of values (being scalar, or functions), and to check at compilation-time the consistency of the distribution described with regard to the system's architecture.

6.2. Reliable distributed real-time embedded systems

Participants: A. Girault [contact person], H. Kalla, E. Saule, H. Yu, N. Brinis.

6.2.1. Reliable scheduling for real-time embedded code

We have continued our work on the automatic generation of reliable and distributed 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.

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.

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.

6.3. Control/scheduling co-design

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

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 [15]. 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. Scheduling for control

Within our approach, the control system timing requirements are captured through a partition in control paths, whose fixed priorities are assigned according to their relative urgency. Latencies are managed through precedence constraints and more or less tight synchronization between modules. The implementation uses the fixed-priority based preemption service of an off-the-shelf real-time operating system. Such a system can be modeled with timed event graphs, and its temporal behavior can be analyzed off-line using the underlying (max,plus) algebra [14].

This methodology is supported by the version of ORCCAD under development. It will be further improved using a QoS management of the timing constraints to fully benefit from the intrinsic robustness of closed-loop controllers w.r.t. timing uncertainties. Some studies are presented in [26] for real-time control in robotics.

6.3.2. Control for scheduling

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 [41]. 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.

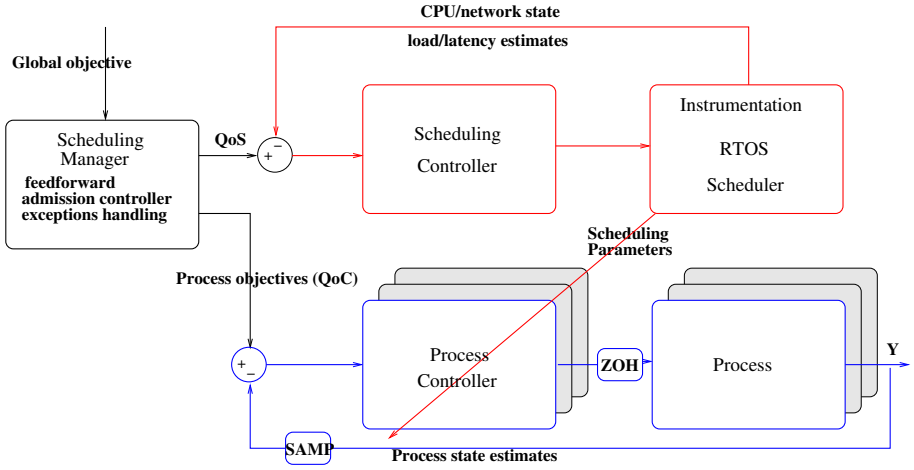

Figure 1. Hierarchical control structure.

Preliminary studies and experiments have been conducted along the following guidelines:

- Ideally the outer loop should control a composite of QoC and QoS; however, due to the lack of knowledge about the relations between a the control performance and the timing parameters, we have up to now been focusing only on the control of the computing load. Indeed, the QoC is indirectly controlled through the temporal attributes of the control law. Finding effective cost functions that map the control performance into the scheduling parameters can be difficult, especially for non-linear systems.
- As the task periods directly affect the computing load, they have been chosen as actuators. They can be implemented through software variable clocks.

Also, as timing uncertainties cannot be avoided and are difficult to model or measure, we are currently designing robust control algorithms using the H_∞ control theory. For example, Figure 2 shows a robust scheduling controller where template W_e specifies the performances on the CPU load tracking error and template W_x specifies the load allocation between two control tasks. Such H_∞ scheduling controllers have been successfully simulated and experimentally validated [25], [26]

6.3.3. Integrated control/scheduling co-design

As the scheduling controller adapts the periods of the plant control tasks, the plant controllers gains must be sized according to the variable sampling period h in order to preserve stability and to satisfy a given performance index.

Our design objective is to obtain a unique controller as a function of h instead of a map of different controllers as in the past. Therefore, the stability can be theoretically ensured for all control periods h over a desired range. A polynomial pole-placement approach is used and a sampling period dependent RST (two degrees of freedom) discrete-time controller has been designed [25].

The desired closed-loop performance is specified by model matching: it has been shown using the Truetime tools that decreasing the performance (e.g., the response time) while increasing the sampling period allows for preserving the stability over a wide control frequency range.

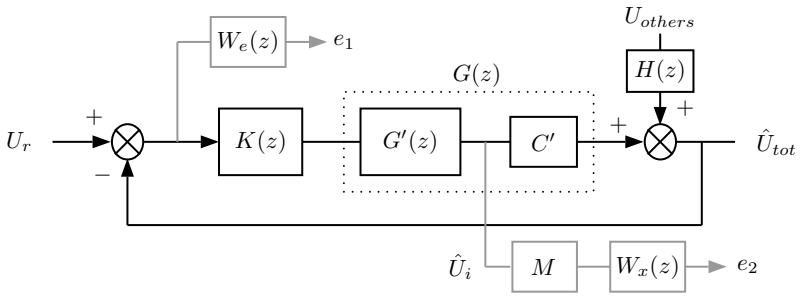

Figure 2. Robust H_∞ feedback scheduling bloc diagram

Control laws using variable sampling are currently under study, e.g., via new extensions of the gain scheduling and Linear Parameter Varying (LPV) design methods, considering here that the sampling period is the varying parameter.

6.3.4. Simulations and experiments

Experiments are implemented using a modified ORCCAD runtime under Linux/RTAI (hard real-time using kernel modules) or under a patched Linux kernel (soft real-time using Posix threads).

Feedback scheduling has been successfully implemented in robot control [26] where simulation results (using Truetime) and experiments (using RTAI) are compared running the so-called Computed Torque robot controller. The results show that our method provides both robustness w.r.t. unmodelled loads and a controlled use of the computing resource with a moderate computing cost (Figure 3) : the upper part pictures the measured tasks periods and the lower part shows the CPU load, in response to a step in the desired total CPU load at time 1.5 sec. The jitter that is observed during the execution in real-time (right part) is due partly from variable latencies during interrupts handling and partly from variations in the tasks computation durations.

It is expected that, as adaptive closed-loop scheduling is somewhat tolerant w.r.t. timing uncertainties, our approach can be compliant with a soft and portable real-time implementation of control systems.

Further work will study improved versions of robust controllers with a moderate complexity, a process requirements based formulation of QoC/QoS criteria, the implementation of execution time measurements in a Posix compliant kernel and a full implementation of the system including QoS management issues. In particular this work will be done in the framework of the SAFE_NECS project (see 8.2.6).

6.4. Automatic generation of correct controllers

Participants: N. Brinis, G. Delaval [contact person], E. Dumitrescu, A. Girault, H. Yu.

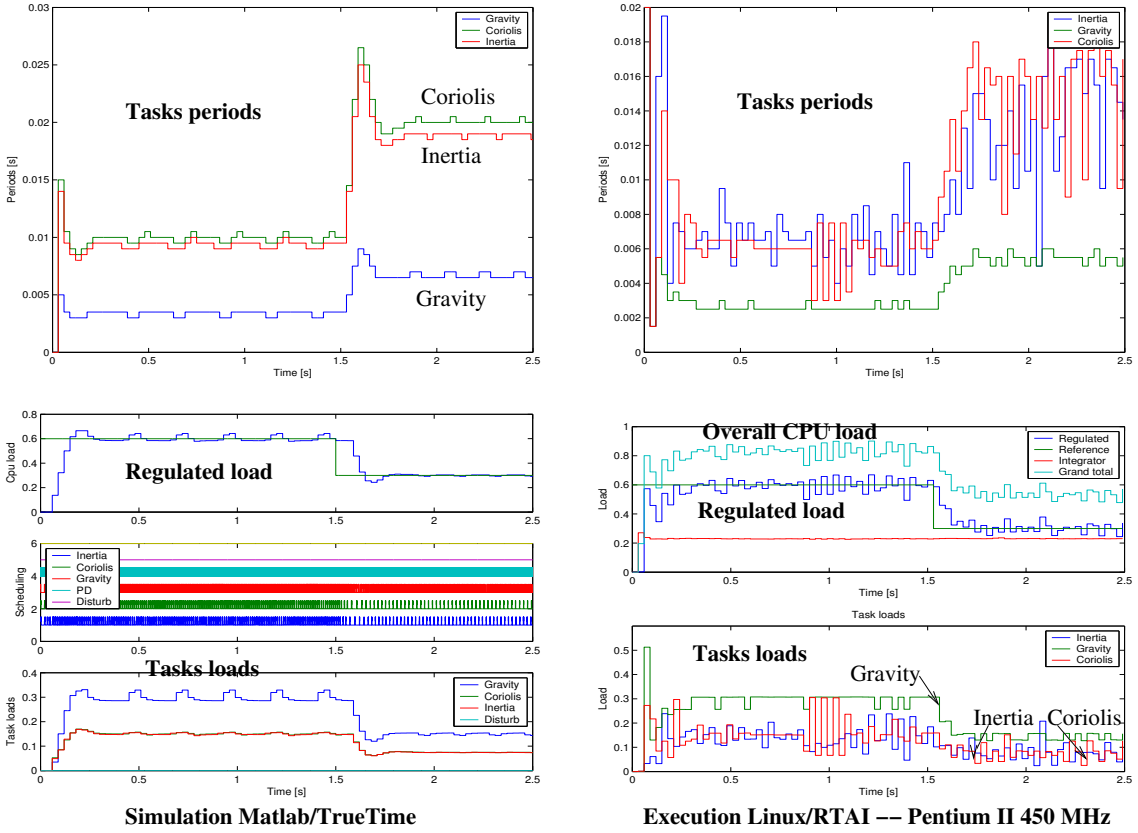

Figure 3. Feedback scheduling a robot controller

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

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 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* [59].

We propose a simple programming language, called NEMO ([21], [28]), 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 [2]. 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 [54] and the Sigali synthesis tool [55].

6.4.2. The control of multi-mode multi-tasking systems

Work in the last few years has produced a methodology for the automatic generation of correct controllers for multi-task systems, in the form of property-enforcing layers [61]. The model of commonly found task control patterns is proposed in terms of labeled transition systems, representing idle, waiting, or active states, and transitions in reaction to requests, authorizations and termination events. Quantitative weights can be associated to active states, representing costs (time, power consumption) or quality level. Standard properties of the interactions between such components are formulated, possibly using observers, in terms of invariants or configurations that should be always reachable. When a system is modeled by composing instantiations of such patterns, discrete controller synthesis is applied to obtain automatically (if it exists) the controller of activations such that the properties are satisfied, and the weights are optimized. This work is done in cooperation with VERIMAG (Synchronous team) and IRISA/INRIA-Rennes (VERTECS project team). An application of this framework concerns fault-tolerance (see Section 6.4.3).

Ongoing work deals with the definition of a domain-specific language, NEMO, where a user can describe a multi-task system using constructs in terms of resources and their characteristics (implying implicit properties to be enforced), tasks and their essential control aspects (modes, controllability of start and stop), additional properties (explicitly stated, between tasks), and applications (sequencings of tasks). This language is compiled into an automaton-based model (concretely: Mode Automata) and associated synthesis objectives, which are processed by a controller synthesis tool (SIGALI), in order to produce the result. A prototype has been implemented (see Section 5.3.1).

6.4.3. Fault-tolerant systems

In order to obtain automatically fault-tolerant real-time systems, we investigate a new solution based on the application of discrete controller synthesis. 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 controller synthesis, 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 [46]. Work in progress involves fine-tuning algorithms for optimal synthesis along paths, and its application to the control of sequences of reconfigurations.

We have also managed to take into account the value failures of sensors, hence allowing us to control a liquid tank system equipped with four level sensors and three valves to fill and empty the tank. The fact that the sensors are subject to value failures is modeled thanks to uncontrollable inputs: when a sensor is faulty, instead of outputting a value accurate *w.r.t.* its state (*i.e.*, wet or dry), it outputs an uncontrollable input. To obtain automatically a controller ensuring that the tank is never empty nor over flooding, we have added a synchronous observer and used the discrete controller synthesis tool SIGALI.

Finally, we have revisited the classical Byzantine Generals problem of Lamport et al [51]. Several divisions of the Byzantine army, each commanded by its own general, are camped outside an enemy city. The generals must agree on a common plan of action (attack or retreat) by exchanging only oral messages. But when a general is a traitor, he can send incoherent orders. We have modeled this fact with uncontrollable inputs. Lamport et. al have proved by induction on the number of generals that this problem is solvable if and only if more than two-thirds of the generals are loyal. To obtain the same result with discrete controller synthesis, we have added, to the models of the generals, an environment model the most permissive possible that allows the generals to become traitors or not. Then we have shown that, among four generals, such an environment model constrained by SIGALI only allows one general to become a traitor and still guarantee that the loyal generals will reach the consensus. This result is consistent with the one of Lamport et. al.

This work is conducted in collaboration with É. Rutten (Dart project team from Inria Futurs, Lille) and E. Dumitrescu (Insa Lyon).

6.5. Component-based Construction

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

Component-based modeling is crucial to overcome the complexity of embedded systems. 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 technique of model-checking allows to verify or falsify correctness of the system with respect to some properties, but it has two drawbacks: its cost and the fact that this method is not constructive. The goal of *compositional* modeling is to guarantee correctness of real-time systems at a reasonable cost. The idea of compositionality is to infer properties of a model from the properties of its components. It is therefore necessary to find properties on the structure of the components and on their composition that imply the required properties of the composed model.

The heterogeneous nature comes from the fact that it is usually necessary to compose different parts of the system on different levels of abstraction, and using different *models of computation* (e.g., timed and untimed automata), *models of interaction* (e.g., blocking or non-blocking, rendez-vous or broadcast), and *models of execution*. The modeling formalism and the composition operation has to support this heterogeneous nature of the components.

The goal of the recent master's thesis of A. Khan is to develop a method and tool to connect PROMETHEUS [47] with the CADP tool [45] developed by the VASY team, in order to combine different and complementary approaches for compositional verification.

6.5.1. Correctness by construction.

Within the component model of [7], we have further improved the results of [13] on component-based construction.

We have developed a new algorithm for the compositional construction of schedulers ensuring reachability of states. The algorithm can be used at design time for symbolic simulation of reactive systems, and to ensure reachability properties of embedded systems at run-time.

These results have been implemented in the PROMETHEUS tool.

6.5.2. Compositional verification of SystemC models.

Within the VERCORS project with STMicroelectronics, we have developed a back-end tool for the generation of PROMETHEUS code, connected to the SystemC [57] parser PINAPA [56]. The goal is to apply the compositional verification algorithms of PROMETHEUS to verify properties of the SystemC model.

6.5.3. Modeling and compositional verification of genetic networks.

Proteins fulfill a huge number of functions in living organisms. 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 is thus defined by a regulatory network which usually encompasses a multitude of highly complex feedback loops. Being able to analyze its structure and behavior is crucial for understanding the functions of the proteins and their interactions.

We have been studying the component-based modeling of genetic regulatory networks within our component framework, in order to compositionally verify properties of the network such as existence of equilibria and reachability of states. We have carried out several case studies with strongly encouraging results. This ongoing work benefits from the contact with H. de Jong (HELIX team at INRIA-Rhône-Alpes).

6.5.4. Synchronous components.

In the context of the ACI ALIDECS, we started a research project on the definition of a language and framework for the construction of safe embedded systems based on synchronous components. This project is the main focus of M. Tivoli's post-doc.

6.6. Aspect-oriented programming

Participants: T. Ayav, S. Djoko Djoko, P. Fradet [contact person], A. Girault.

The goal of Aspect-Oriented Programming (AOP) is to isolate aspects (such as security, synchronization, or error handling) that 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 of AOP (semantics, analysis, verification) and by considering domain-specific aspects (availability or fault tolerance aspects) as formal properties.

6.6.1. Semantics and analysis of AOP.

We are designing a common semantics base to describe precisely AOP languages and features. It will allow us to compare different proposals and to establish formal foundations for static analysis. Our structural operational semantics takes the form of a monitor filtering events produced by the execution of the base program and inserting aspects [31]. One objective is to remain as generic as possible so that it can model various standard AOP languages for object-oriented languages (e.g. AspectJ, Caesar, Composition Filter) as well as more exotic ones (aspects for functional languages, domain-specific aspects).

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 (i.e., $x > 0$), temporal properties (i.e., eventually x will be 0) or even non functional properties (i.e., the worst case execution time of method m is less than 42). The verification and analysis of aspect-oriented programs is the subject of S. Djoko Djoko's PhD thesis.

This work is conducted within the Formal Methods Lab of the network of excellence AOSD-Europe. It is done in collaboration with R. Douence and D. Le Botlan from the OBASCO project team at École des Mines de Nantes.

6.6.2. 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. 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). Aspects specify time limits or orderings in the allocation of resources. They can be seen as the specification of an availability policy. 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 (e.g., UPPAAL) to verify that aspects enforce the required availability properties [22].

The final definition of the aspect language and the formalization of the different steps (abstraction, weaving, translations) should be completed by the end of the year.

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

6.6.3. 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. The fault-intolerant initial system consists of a set of independent periodic tasks scheduled onto a set of fail-silent processors. We transform the tasks such that, assuming the availability of an additional spare processor, the 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. This 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, related to the ALIDECS project (see section 8.2.2), is the main focus of T. Ayav's post-doc.

6.7. Programming models and calculi

Participant: P. Fradet.

We have been interested for a long time in formal calculi in order to study programming language issues in the simplest possible setting. We present here our work within the λ-calculus (compilation of higher-order sequential languages) and the γ-calculus (higher-order parallel and non-deterministic programming).

6.7.1. λ-calculus and the Krivine abstract machine

The Krivine machine is a simple and natural implementation of the call-by-name λ-calculus. While its original description has remained unpublished, this machine has served as a basis for many variants, extensions and theoretical studies. We have presented the Krivine machine and some well-known variants in a common framework [11]. We have characterized the essence of the Krivine machine and have located it in the design space of functional language implementations. This work is based on the framework that we had previously developed for the systematic study of functional language implementations [44].

This is joint work with R. Douence from the OBASCO project team (École des Mines de Nantes).

6.7.2. γ-calculus and higher-order chemical programming

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

We have proposed a formal and basic calculus, the γ-calculus [19], which allows the definition of γ-abstractions (*i.e.*, rewritings) as first class citizens (in the same sense as λ-abstractions in the λ-calculus). This calculus can of course express the classical Gamma formalism, but its higher-order nature makes it easy to describe notions such as code mobility, distribution, adaptation, etc. We have illustrated these advantages by specifying an autonomic mail system as a solution (multiset) of data and reaction rules [18]. A distinctive

feature of our specification is its modularity. Each autonomic property (self-organization, self-healing, self-optimization, self-protection, self-configuration) was implemented by adding new reactive molecules in the solution.

Another generalization of the Gamma language stands in the introduction of multisets with infinite cardinality and multisets with a negative cardinality. These new kind of data structures, combined with the above higher-order properties, have been integrated in the Higher-Order Chemical Language HOCL [16], [27]. We are currently working on the application of HOCL to Grid programming and, more generally, to the programming of distributed applications [20]. 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.

The *Grand Challenge in Non-Classical Computation* workshop has been an occasion to expose our model [17] and to have a large overview on non-conventional models of computation. We have also written for the same event, a position paper raising fundamental questions about non-classical programming languages [24].

This work is conducted in collaboration with J.-P. Banâtre and Y. Radenac from the PARIS project team at IRISA.

7. Contracts and Grants with Industry

7.1. STMicroelectronics

In the context of the MINEFI NANO 2008 programme, we have a cooperation VERCORS (compositional verification of transactional models of systems-on-chip) with STMicroelectronics (Crolles), SysArt team. The goal is to translate and formally verify SystemC models on the transaction level into Prometheus code, in order to compositionally verify correctness of the model with respect to deadlock freedom, liveness, and reachability properties.

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.

8. Other Grants and Activities

8.1. Regional actions

8.1.1. JESSICA

Jessica[27] is a national program funded by the Ministry for Industry: it aims at helping small and medium companies for the integration of electronics (hardware and embedded software) in their products. Through its regional branches it provides training and technical expertise on specific innovative projects. In this framework we provide expertise about embedded real-time systems upon request of ESISAR/INPG, one of the managers of Jessica for the South-East part of France.

[27] http://www.jessica-puce.prd.fr

8.1.2. 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 [28]. It supports research on feedback scheduling together with the development of dedicated software tools.

8.2. National actions

8.2.1. ACI "Sécurité & Informatique" project Dispo

The DISPO project[29] 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.

8.2.2. ACI "Sécurité & Informatique" project Alidecs

The objective of the ALIDECS project[30] 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 are interested in weaving-like techniques for enforcing fault-tolerance properties to reactive systems. With the arrival of M. Tivoli on a post-doctoral position, we have also started to study static analysis of networks of real-time components.

8.2.3. CNRS AS 155 of RTP 24: Hybrid systems

Action Spécifique CNRS AS 155, related to RTP 24 (*Mathématiques du signal et des Systèmes*), is entitled: *Approches formelles pour l'analyse et la synthèse sûre de contrôle des systèmes dynamiques hybrides*, and is a working group on the analysis and synthesis of hybrid systems, under a control theory perspective approach.

8.2.4. CNRS RTP 21: Fault-tolerance

We are collaborating to this RTP entitled *Sûreté de fonctionnement des systèmes informatiques complexes ouverts*[31].

8.2.5. CNRS RTP 55: Network controlled systems

NECS (NEtworked Control Systems)[32] 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).

8.2.6. ARA-SSIA Safe_NECS

SAFE_NECS is an « Action de Recherche Amont - Sécurité, Systèmes embarqués et Intelligence Ambiante » funded by the ANR. It has been labeled at the end of 2005 on the topic of fault tolerant control of distributed process under resource 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.

[28] http://pop-art.inrialpes.fr/people/simon/c3o/
[29] http://www.irisa.fr/lande/jensen/dispo.html
[30] http://www-verimag.imag.fr/SYNCHRONE/alidecs/
[31] http://www.laas.fr/RTP21-SdF
[32] http://www-lag.ensieg.inpg.fr/canudas/necs.htm

- VERTECS at IRISA/INRIA-Rennes is working with us on applications of discrete controller synthesis, and in paticular on the tool SIGALI.
- P. Fradet cooperates with T. Jensen and 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).
- G. Goessler cooperates with H. de Jong (Helix project, UR Rhône-Alpes) on modeling genetic networks.

8.2.8. Cooperations with other laboratories

- A. Girault cooperates with X. Nicollin (VERIMAG), M. Pouzet (LRI, University of Paris VI), Denis Trystram 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).
- D. Simon cooperates with O. Sename (LAG).

8.3. European actions

8.3.1. Artist 2 European IST network

ARTIST 2 is a European Network of Excellence on embedded system design[33]. 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.

8.3.2. AOSD European IST network

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).

8.3.3. EAST-EEA European ITEA project

The EAST-EEA project (Embedded Electronics Architecture) aims at proposing a methodology in order to develop complex real-time embedded applications in the field of transportation, especially for automobiles. The PhD of Hamoudi Kalla has been funded by this project.

9. Dissemination

9.1. Scientific community

- P. Fradet has participated in the program committees of JFDLPA'05 (*Seconde Journée Francophone sur le Développement de Logiciels Par Aspects*), EIWAS'05 (*European Interactive Workshop on Aspects in Software*) and FOAL'05 (*Foundations of Aspect-Oriented Languages Workshop*). He is co-editor of *Unconventional Programming Paradigms*, revised selected and invited papers, Volume 3566 of Lecture Notes in Computer Science, Springer-Verlag [10]. He has given a course with Jean-Pierre Banâtre on Chemical Programming at *Ecole des Jeunes Chercheurs en Programmation*, Saint-Malo, juin 2005.

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

- A. Girault serves as associate editor for the *Eurasip Journal on Embedded Systems*. He has participated in the program committees for EMSOFT'05 (*Embedded Software*) and SLAP'05 (*Synchronous Languages, Applications and Programming*), and maintains the *SYNchronous Applications, Languages, and Programs* web site[34]. He has been reviewer for the PhD of A. Curic (Verimag/UJF, Grenoble, France) and for the Master Thesis of S. Dayaratne (University of Auckland, New Zealand).
- D. Simon is a member of the RTNS'06 (14th international conference on real-time and network systems) program committee. He has been examiner in the PhD commission of Th. Garcia-Fernandez (LINA, Nantes) about *Conception et développement de composants pour logiciels temps-réel embarqués*.

9.2. Teaching

9.2.1. Courses

- P. Fradet: *Algorithms and Functional Programming* (Introduction to programming), 32h, Université Joseph Fourier.
- Alain Girault: algorithmics and programming in Java, 26h, INPG Telecom Department.
- Gregor Goessler: compilation project, 2nd year engineering, 55 h, ENSIMAG Grenoble.
- Daniel Simon gave a talk during ETR'05 (École d'été temps-réel), Nancy, September 2005.

9.2.2. Advising

PhDs:

- Gwenaël Delaval, co-advised by Alain Girault (with M. Pouzet, LRI), since 9/2004, PhD in computer science, INPG.
- Stéphane Hong Tuan Ha, co-advised by Pascal Fradet (with T. Jensen, IRISA), since 10/2002, PhD in computer science, Université de Rennes I.
- Yann Radenac, co-advised by Pascal Fradet (with J.-P. Banâtre, IRISA), since 9/2003, PhD in computer science, Université de Rennes I.
- Hamoudi Kalla, co-advised by Alain Girault (with Y. Sorel, AOSTE Team), since 1/2001, PhD in computer science, INPG.
- David Robert, co-advised by Daniel Simon and Olivier Sename, since 9/2003, PhD in Control Theory, INPG.
- Simplice Djoko Djoko, co-advised by P. Fradet (with R. Douence, OBASCO, École des Mines de Nantes), since 10/2005, PhD in computer science, Université de Nantes.

Masters:

- Darina Dimitrova, *Automatic multitasking code generator for Orccad*, projet de fin d'études, Technical University-Sofia, Bulgaria, advised by D. Simon.
- Huafeng Yu, *Synthèse de contrôleur pour la tolérance aux fautes de capteurs*, INP Grenoble, advised by A. Girault.
- Nour Brinis, *Synthèse d'un contrôleur pour le problème des généraux byzantins*, École Nationale des Sciences de l'Informatique, La Manouba, Tunisie, co-advised by A. Girault and M. Yeddes.
- Erik Saule, *Ordonnancement fiable pour la génération de code temps-réel embarqué*, INP Grenoble, co-advised by A. Girault and D. Trystram.

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

10. Bibliography

Major publications by the team in recent years

[1] 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° 1/2, 7-9 2002, p. 55-84.

[2] 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° 4, 1998, p. 338–359.

[3] 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° 3, May 1999, p. 416–427.

[4] 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.

[5] P. FRADET. *Approches langages pour la conception et la mise en œuvre de programmes*, Document d'habilitation à diriger des recherches, Université de Rennes 1, November 2000.

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

[7] 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.

[8] D. SIMON, B. ESPIAU, E. CASTILLO, K. KAPELLOS. *Computer-Aided Design of a Generic Robot Controller Handling Reactivity and Real-Time Control Issues*, in "IEEE Trans. on Control Systems Technology", vol. 1, n° 4, December 1993.

[9] D. SIMON, A. GIRAULT. *Synchronous programming of Automatic Control Applications using* ORCCAD *and* ESTEREL, in "40th Conference on Decison and Control", 2001.

Books and Monographs

[10] J.-P. BANÂTRE, P. FRADET, J.-L. GIAVITTO, O. MICHEL (editors). *Unconventional Programming Paradigms,*, Springer-Verlag, LNCS, Vol. 3566, Revised Selected and Invited Papers of the International Workshop UPP 2004, Le Mont-Saint-Michel, France, 2005.

Articles in refereed journals and book chapters

[11] R. DOUENCE, P. FRADET. *The next 700 Krivine Machines*, in "Higher-Order and Symbolic Computation", to appear, 2006.

[12] A. GIRAULT, X. NICOLLIN, M. POUZET. *Automatic Rate Desynchronization of Embedded Reactive Programs*, in "ACM Trans. on Embedded Computing Systems", to appear, 2006.

[13] G. GÖSSLER, J. SIFAKIS. *Composition for Component-based Modeling*, in "Science of Computer Programming", vol. 55, nº 1-3, 3 2005, p. 161-183.

[14] D. SIMON, F. BENATTAR. *Design of real-time periodic control systems through synchronisation and fixed priorities*, in "Int. Journal of Systems Science", vol. 36, nº 2, 2005, p. 57-76.

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

Publications in Conferences and Workshops

[16] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *A Generalized Higher-Order Chemical Computation Model with Infinite and Hybrid Multisets*, in "In Pre-Proceedings of 1st International Workshop on New Developments in Computational Models (DCM 2005)", ENTCS Elsevier, September 2005, p. 5–14.

[17] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Higher-order Chemical Model of Computation*, in "The Grand Challenge in Non-Classical Computation", April 2005, http://www.cs.york.ac.uk/nature/workshop/papers/BanatreFradetRadenac.p

[18] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Higher-order Chemical Programming Style*, in "Proceedings of Unconventional Programming Paradigms", Springer-Verlag, LNCS, Vol. 3566, 2005, p. 84–98.

[19] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Principles of Chemical Programming*, in "Proceedings of the 5th International Workshop on Rule-Based Programming", S. ABDENNADHER, C. RINGEISSEN (editors). , ENTCS, vol. 124(1), Elsevier, June 2005, p. 133–147.

[20] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Towards Grid Chemical Coordination (short paper)*, in "Proceedings of Symposium on Applied Computing (SAC'06)", to appear, 2006.

[21] 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", to appear, ACM press, April 2006, http://pop-art.inrialpes.fr/people/delaval/pub/article-nemo.pdf.

[22] P. FRADET, S. HONG TUAN HA. *Systèmes de gestion de ressources et aspects de disponibilité*, in "2ème Journée Francophone sur le Développement de Logiciels Par Aspects (JFDLPA 2005), Lille, France", (In French), September 2005.

[23] A. GIRAULT. *A Survey of Automatic Distribution Method for Synchronous Programs*, in "International Workshop on Synchronous Languages, Applications and Programs, SLAP'05, Edinburgh, UK", F. MARANINCHI, M. POUZET, V. ROY (editors). , ENTCS, Elsevier Science, April 2005.

[24] O. MICHEL, J.-P. BANÂTRE, P. FRADET, J.-L. GIAVITTO. *Challenging Questions for the Rationals of Non-Classical Programming Languages*, in "The Grand Challenge in Non-Classical Computation", April 2005, http://www.cs.york.ac.uk/nature/workshop/papers/Michel.pdf.

[25] D. ROBERT, O. SENAME, D. SIMON. *Sampling period dependent RST controller used in control/scheduling co-design*, in "16th IFAC 2005 World Conference, Prague", July 2005.

[26] D. SIMON, D. ROBERT, O. SENAME. *Robust control/scheduling co-design: application to robot control*, in "RTAS'05 IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco", March 2005, p. 118-127.

Internal Reports

[27] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Generalized Multisets for Chemical Programming*, Research Report, n° 5743, INRIA, November 2005, http://www.inria.fr/rrrt/rr-5743.html.

[28] G. DELAVAL, E. RUTTEN. *A Domain-Specific Language for Multi-task Systems, applying Discrete Controller Synthesis*, Research Report, n° 5690, INRIA, sep 2005, http://www.inria.fr/rrrt/rr-5690.html.

[29] G. GÖSSLER. *Reach Scheduling for Embedded Systems*, Research Report, n° 5651, INRIA, France, 2005, http://www.inria.fr/rrrt/rr-5651.html.

Miscellaneous

[30] N. BRINIS. *Synthèse d'un Contrôleur pour le Problème des Généraux Byzantins*, Masters Thesis, École Nationale des Sciences de l'Informatique, La Manouba, Tunisie, July 2005.

[31] R. DOUENCE, S. DJOKO DJOKO, P. FRADET, D. LE BOTLAN, T. STAIJEN. *Towards a Common Aspect Semantic Base (CASB)*, Milestone 8.2, AOSD-Europe, EU Network of Excellence, October 2005.

[32] E. SAULE. *Ordonnancement Fiable pour la Génération de Code Temps-Réel Embarqué*, Masters Thesis, INPG, Grenoble, France, June 2005.

[33] H. YU. *Synthèse de Contrôleurs pour la Tolérance aux Fautes des Capteurs*, Masters Thesis, Institut National Polytechnique de Grenoble, Grenoble, France, June 2005.

Bibliography in notes

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

[35] 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.

[36] 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° 4, October 2001, p. 397–451.

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

[38] 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.

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

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

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

[42] 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° 2, 1986, p. 244-263.

[43] 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/.

[44] R. DOUENCE, P. FRADET. *A Systematic Study of Functional Language Implementations*, in "ACM Transactions on Programming Languages and Systems", vol. 20, n° 2, 1998, p. 344–387.

[45] J.-C. FERNANDEZ, H. GARAVEL, A. KERBRAT, R. MATEESCU, L. MOUNIER, M. SIGHIREANU. *CADP: A Protocol Validation and Verification Toolbox*, in "Proc. CAV '96", R. ALUR, T. HENZINGER (editors). , LNCS, vol. 1102, Springer-Verlag, 8 1996, p. 437-440.

[46] 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.

[47] 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.

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

[49] 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.

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

[51] L. LAMPORT, R. SHOSTAK, M. PEASE. *The Byzantine Generals Problem*, in "ACM Trans. on Programming Languages and Systems", vol. 4, n° 3, July 1982, p. 382–401.

[52] 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° 1, 2002, p. 85–126.

[53] 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.

[54] 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° 3, March 2003, p. 219-254.

[55] 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° 4, October 2000, p. 325–346.

[56] M. MOY, F. MARANINCHI, L. MAILLET-CONTOZ. *PINAPA: An Extraction Tool for SystemC descriptions of Systems-on-a-Chip*, in "proc. EMSOFT'05", 2005.

[57] OPEN SYSTEMC INITIATIVE. *SystemC*, http://www.systemc.org.

[58] 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.

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

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

[61] E. RUTTEN, H. MARCHAND. *Automatic Generation of Safe Handlers for Multi-Task Systems*, Rapport de Recherche, n° 5345, INRIA, October 2004, http://www.inria.fr/rrrt/rr-5345.html.