INRIA

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 **1**
2. Overall Objectives **1**
3. Scientific Foundations **2**
3.1. Embedded systems and their safe design 2
3.1.1. The safe design of embedded real-time control systems. 2
3.1.2. Models, methods and techniques. 2
3.2. Issues in design automation for complex systems 3
3.2.1. Hard problems. 3
3.2.2. Applicative needs. 4
3.2.3. Our approach. 4
3.3. Main Research Directions 4
3.3.1. Principles 4
3.3.2. Main Directions 5
3.3.2.1. Implementations of synchronous programs. 5
3.3.2.2. Control/scheduling co-design. 5
3.3.2.3. Automatic generation of correct controllers 5
4. Application Domains **6**
4.1.1. Industrial applications. 6
4.1.2. Industrial design tools. 6
4.1.3. Some of our industrial cooperations. 6
5. Software **7**
5.1. Orccad 7
5.2. Implementations of synchronous programs 7
5.2.1. Code distribution 7
5.2.2. Fault-tolerance 7
5.3. Prototypes 7
5.3.1. Automatic Controller Generation 7
5.3.2. Compositionality 8
6. New Results **8**
6.1. Higher-order synchronous data-flow programming 8
6.2. Reliable distributed real-time embedded systems 8
6.2.1. Static schedules tolerant to processor and bus failures 8
6.2.2. Static schedules tolerant to processor and point-to-point communication link failures 9
6.2.3. Reliable static schedules 9
6.3. Control/scheduling co-design 9
6.3.1. Scheduling for regulation 9
6.3.2. Regulation for scheduling 10
6.4. Automatic generation of correct controllers 13
6.4.1. The control of multimode multi-tasking systems 13
6.4.2. Fault-tolerant systems 13
6.5. Component-based Construction 13
6.5.1. Correctness by construction. 14
6.5.2. Component-based embedded systems. 14
6.6. Aspect-oriented programming 14
6.6.1. Trace-based AOP and interaction analysis. 15
6.6.2. Aspects of scheduling and fusion of networks of components. 15
6.6.3. Reactive and aspect-oriented programming 15

6.6.4. Fault-tolerance aspects for real-time software 16
6.7. Programming models and calculi 16
6.7.1. λ-calculus and the Krivine abstract machine 16
6.7.2. γ-calculus and higher-order chemical programming 16
7. Contracts and Grants with Industry 16
7.1. STMicroelectronics 16
8. Other Grants and Activities 17
8.1. Regional actions 17
8.1.1. JESSICA 17
8.1.2. Local Arc C^3O 17
8.1.3. Local Arc Ctrl-A 17
8.2. National actions 17
8.2.1. ACI "Sécurité & Informatique" project Dispo 17
8.2.2. ACI "Sécurité & Informatique" project Alidecs 17
8.2.3. CNRS AS 155 of RTP 24: Hybrid systems 17
8.2.4. CNRS RTP 21: Fault-tolerance 17
8.2.5. CNRS RTP 55: Network controlled systems 17
8.2.6. Cooperations internal to Inria 18
8.2.7. Cooperations with other laboratories 18
8.3. European actions 18
8.3.1. Artist European IST network 18
8.3.2. Artist 2 European IST network 18
8.3.3. AOSD European IST network 18
8.3.4. EAST-EEA European ITEA project 18
9. Dissemination 19
9.1. Scientific community 19
9.2. Teaching 20
9.2.1. Courses 20
9.2.2. Advising 20
10. Bibliography 21

1. Team

Project leader
Eric Rutten [CR INRIA]

Project assistants
Elodie Toïhein

Personnel Inria
Pascal Fradet [CR]
Alain Girault [CR]
Gregor Goessler [CR]
Daniel Simon [CR]

PhD students
Gwenaël Delaval [MENRT grant, since 10/2004]
Hamoudi Kalla [INRIA-EEA grant]
David Robert [MENRT EEATS grant]

Post-doctoral researchers
Tolga Ayav [MENRT/INRIA post-doctoral grant, since 9/2004]
Emil Dumitrescu [ARTIST Network, until 8/2004]

Interns
Gwenaël Delaval [March–September, M2R S&L, Grenoble]
Nicolas Leignel [March–July, M2R ROCO, Grenoble]
Thomas Leveque [June–August, M1, Grenoble]
Nicolas Palix [March–August, ESISAR/INPG and M2R S&L, Grenoble]
Safouan Taha [March–July, M2R S&L, Grenoble]
Huafeng Yu [March–June, M2R S&L, Grenoble]

External Collaborators
Emil Dumitrescu [INSA Lyon, since 9/2004]
Olivier Sename [LAG, INPG-ENSIEG]

2. 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);
- control/scheduling co-design, with cross-interactions between techniques of serving and real-time operating systems (RTOS), in order to obtain an adaptative scheduling, with regard to quality of service (in relation with the ORCCAD[4] 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
[4] http://www.inrialpes.fr/iramr/pub/Orccad

- 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 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 safety 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,
- the RNTL-funded Automate project (ATHYS, COMAU/Renault Automation) on safe control components for the automation of assembly production cells,
- cooperations 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, internationally and in particular, in the European IST network ARTIST (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.

[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/
[10] http://www.systemes-critiques.org/ARTIST

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 (or finite or infinite state automata) [34][39]. During the years, methodologies for the formal verification [51][41], control synthesis [52] and compilation, and extensions to timed and hybrid systems [48][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 [46], which can be encoded in a synchronous model as shown in [3].

The synchronous approach[11] [44][45] 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 cyclic 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 therefore oriented towards precisely this direction.

From the point of view of the executables and execution platforms for the implementation of embedded systems, there are software or middle-ware approaches and hardware-based approaches. Under 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 a character of open loop. 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 [40][47]; 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 which allow the study of different aspects of their behaviors, and the design of controllers. The European network IST Artist 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 which allow the automatic use of formal methods. In the field of verification, this concerns in particular the technique of model checking;

[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-spi.lip6.fr/lucid-synchrone
[17] http://www-verimag.imag.fr/PEOPLE/Florence.Maraninchi/MATOU

the verification intervenes after the design phase, and requires, in case of problematic diagnostics, expensive backtracks on the specification. We want to make 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 which 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 [36]) 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 compromises between, on the one hand, expressiveness and formal power, and on the other hand, usability and automation. We focus on the field 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 mechanic, *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 hand, the functionalities we want to offer. We are interested in questions concerning:

- Dedicated languages and models for automatic control which 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 hand.
- Compositional modeling and analysis which aim at deriving crucial system properties from component properties, without the need to actually build and check the global system.
- Aspect-Oriented Programming which allows to express safety concerns separately and to enforce them on program.

3.3.2.1. 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 (thus 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.

3.3.2.2. 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 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.

3.3.2.3. Automatic generation of correct controllers

. We use techniques of discrete controller synthesis, especially the tools SIGALI [50] and Mode Automata [49] 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

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, MathLab/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 base, such as for example STATEMATE by iLogix.

Regarding the synchronous approach, commercial tools are available: SCADE (based on LUSTRE) and ESTEREL[19], SILDEX[20] (based on SIGNAL), industrial versions of ESTEREL compilers (for example at France Télécom R&D), 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 coherence 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. Some of our industrial cooperations.

Regarding applications and case studies with industrial end-users of our techniques, we cooperate with:

- STMicroelectronics around design assistance for Systems on Chip, *w.r.t.* controller synthesis, automatic distribution of simulations, compositional verification;
- COMAU (formerly Renault Automation), around modeling components for factory automation;
- Excavation systems industry in the framework of the TELEDIMOS European project.

Regarding transfer of our results and know-how to tool-vending industrials, we interact with:

- France Télécom R&D, by transferring automatic distribution technology into their ESTEREL compiler;
- ATHYS, where methodological aspect from the ORCCAD approach were taken over, as well as a specialized verification framework.

[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 maintained by the *Support Expérimentations & Développement (SED)* service of INRIA-Rhône-Alpes. ORCCAD is used by the experimental robotics platform of INRIA. New functionalities are developed jointly by the *SED* service and the researchers of the Pop Art project. 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. Some concepts of task structuring and dedicated interfaces for the programming of robot systems give place to a transfer of expertise to the company ATHYS.

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 collaborating for several years with the INRIA project 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]. In addition, we also consider transfers 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, E. Dumitrescu, A. Girault, E. Rutten [contact person].

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://www.inrialpes.fr/bip/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 6.4.1).

5.3.2. Compositionality

Participants: G. Goessler [contact person], N. Palix.

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

N. Palix has implemented a prototype tool translating models of *systems-on-chip* written in SystemC, into the input format of Prometheus. This prototype has helped to carry out a case study, 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 Marc Pouzet and Paul Caspi [37][38]. This new language, called DECADE [23], 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 starting a PhD on this topic, co-advised by Marc Pouzet from LIP6 and Alain Girault. He will work in the context of the ALIDECS ACI [26].

6.2. Reliable distributed real-time embedded systems

Participants: A. Girault [contact person], H. Kalla, N. Leignel, T. Leveque, H. Yu.

6.2.1. Static schedules tolerant to processor and bus failures

We have designed a new methodology to generate a static schedule of a given data-flow algorithm $\mathcal{A}lg$ onto a given distributed heterogeneous architecture $\mathcal{A}rc$, tolerant to a fixed number of processor and bus failures. This methodology is intended for multi-bus distributed architectures. The algorithm $\mathcal{A}lg$ is given as an acyclic data-flow graph of blocks and data-dependencies, meant to be repeatedly executed in a periodic way. The architecture $\mathcal{A}rc$ is given as a graph of processors and communication links. Also available are the worst case execution times of each software block (resp. data-dependency) onto each processor (resp. communication link). Since the architecture is heterogeneous, these numbers may be distinct. The failures considered are of the fail-silent type.

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

Our methodology is based on software redundancy: it uses the active replication of $\mathcal{A}lg$'s blocks onto the processors of $\mathcal{A}rc$, and the fragmentation of the data-dependencies and their passive replications onto the busses. These choices were made for the context of embedded real-time systems, where only the existing hardware can be used for redundancy (embedded constraint), and where the schedule must be predictable (real-time constraint). It is implemented as a heuristics that tries to minimize the obtained fault-tolerant schedule length, within the SYNDEX CAD tool[27].

6.2.2. Static schedules tolerant to processor and point-to-point communication link failures

In the same context, we have designed a second methodology, this time intended for architectures with point-to-point communication links. The goal is to generate a static schedule of $\mathcal{A}lg$ onto $\mathcal{A}rc$, tolerant to a fixed number of processor and link failures. Our methodology is also based on software redundancy: it first transforms $\mathcal{A}lg$ into a new graph $\mathcal{A}lg^*$ with redundancies and generates a list of exclusion constraints to be satisfied, and then it distributes/schedules $\mathcal{A}lg^*$ onto $\mathcal{A}rc$ such that the exclusion relations are satisfied. Therefore the blocks if $\mathcal{A}lg$ are actively replicated to tolerate the processor failures, while the data-dependencies are replicated and sent through disjoint routes to tolerate the point-to-point communication link failures [16]. It is implemented as a heuristics that tries to minimize the obtained fault-tolerant schedule length, within the SYNDEX CAD tool.

6.2.3. Reliable static schedules

Still in the same context, we have designed a methodology to generate a static schedule of $\mathcal{A}lg$ onto $\mathcal{A}rc$, satisfying a minimum reliability requirement. We have studied several reliability models and chosen one that is well suited to embedded real-time systems. According to this model, each hardware component of $\mathcal{A}rc$ (processor or communication link) has a possibly distinct reliability level, expressed the probability by time unit that it runs correctly. The architecture is heterogeneous so these figures may be distinct. One problem that arises when scheduling/distributing $\mathcal{A}lg$ onto $\mathcal{A}rc$ is the need to compute efficiently the reliability of the partial schedule at each step of the algorithm. Hence, our method first transforms $\mathcal{A}lg$ into a new graph $\mathcal{A}lg^*$ such that the reliability of $\mathcal{A}lg$ is greater than that of $\mathcal{A}lg^*$ and such that the reliability of $\mathcal{A}lg^*$ is computable in polynomial time. Based on this, we have designed an original bi-criteria heuristics that attempts to minimize the schedule length while at the same time maximizing the system reliability [19]. It is implemented within the SYNDEX CAD tool.

This work is done in collaboration with Yves Sorel from AOSTE project team and Denis Trystram from ID-IMAG.

6.3. Control/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 utilisation 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.

6.3.1. Scheduling for regulation

Control systems are often designed using a set of cooperating periodic modules running under control of a real-time operating system. A correct behavior of the closed-loop controller requires that the system meets timing constraints like periods and latencies, which are often expressed as deadlines. Well known scheduling policies, such as Rate Monotonic for fixed priorities and EDF for dynamic priorities assign priorities according to timing parameters, respectively sampling periods and deadlines. They are said "optimal" as they maximize the number of tasks sets which can be scheduled with respect of deadlines, under some restrictive assumptions.

They hardly take into account precedence and synchronization constraints which naturally appear in a control algorithm. The relative urgency or criticality of the control tasks can be unrelated with the timing

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

parameters. Thus, the timing requirements of control systems *w.r.t.* the desired control goal expressed as a performance index do not fit well with scheduling policies purely based on schedulability tests.

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 [18]. 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 preliminary results have been obtained by providing, in the case of an inverted pendulum, a set of controllers (with different sampling periods), robust according to timing uncertainties (represented as input delays).

6.3.2. Regulation for scheduling

Taking into account the unsuitability of current real-time design to capture feedback control systems requirements naturally leads to use control/scheduling co-design. This closer interaction is particularly needed for control applications requiring high degrees of flexibility, or when computing resources are limited. However, while off-line methods can handle control requirements, they cannot easily handle timing uncertainties due to varying execution times, dynamic reconfigurations, or network induced delays.

Thus it can be useful to consider more *dynamic solutions*, *i.e.*, to adapt the execution of the control task (period and value) according to the availability of the resources. This is called *feedback scheduling*, the purpose of which is to deal with on-line trade-offs between control performance and computing resources (CPU time and communication bandwidth) requirements.

The idea involves adding to the process controller an outer sampled feedback loop ("scheduling regulator") to control the scheduling parameters as a function of 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. Practical solutions will be found in the available control toolbox or in enhancements and adaptation of current control theory.

Feedback scheduling is a dynamic approach allowing the better use of the computing resources, in particular when the workload changes *e.g.*, due to the admission of a new task. We propose in Figure 1 a hierarchical control structure. The feedback scheduler controls the CPU activity according to the computing resource availability (measured through some computing load metric) by adjusting the periods of the tasks used in the process controller(s). The feedback scheduler is here implemented as an application task that runs in parallel with the control tasks, with a higher priority. It executes periodically with a period h_S larger than the sampling periods of all the control tasks, so that the control sampling periods are changed only when the resource availability has been estimated.

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

- While the outer loop should control a composite of QoC and QoS, we up to now focused only on the control of the computing load; indeed, the QoC is thus indirectly controlled through the cost functions relating the control performance and the temporal attributes of the control law. Finding these cost functions 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.

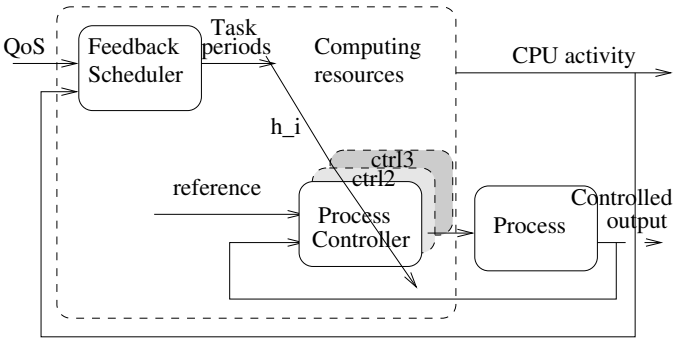

Figure 1. Hierarchical control structure.

The measure or estimation of the computer load remains an unresolved issue, as measuring the actual processing duration of tasks is out of the scope of current off-the-shelf RTOS, *e.g.*, Posix. While the response times are easy to measure with a good precision, the execution times cannot be simply evaluated as the preemption instants are not known from the user-space application. Designing a simple and accurate estimator up to now failed. However, despite the fact that the CPU load estimation from response times is overestimated, it is simple and can be effectively used, *e.g.*, as shown for robot control in [32]. Figure 2 on the left shows simulation results where execution times are assumed to be known while the right column displays execution under ORCCAD and RTAI[28] using the response time approximation.

Patching the kernel of open-source RTOS (Linux/RTAI) to directly measure the tasks execution times is currently investigated to improve the load estimation, with the cost of loss of portability.

As the scheduling controller adapts the periods of the plant control tasks, the plant controller must fit with 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.

The desired closed-loop performances is specified by model matching: it has been shown 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.

Also, as timing uncertainties cannot be avoided and are difficult to model or measure, the choice of control algorithms focus on robustness *w.r.t.* unknown delays, *e.g.*, using recent results in H_∞ control theory. For example, Figure 3 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 the two control tasks.

The approach combining the sampling dependent RST controller, the variable desired performance, and the H_∞ scheduling controller has been successfully simulated [31].

Experiments are implemented using a modified ORCCAD runtime under Linux/RTAI (kernel hard real-time) or under a patched Linux kernel (soft real-time using high resolution Posix timers). The results show that the method provides both robustness *w.r.t.* unmodelled loads and a controlled use of the computing resource. 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 and a full implementation of the system including QoS management issues.

[28] http://www.aero.polimi.it/~rtai/

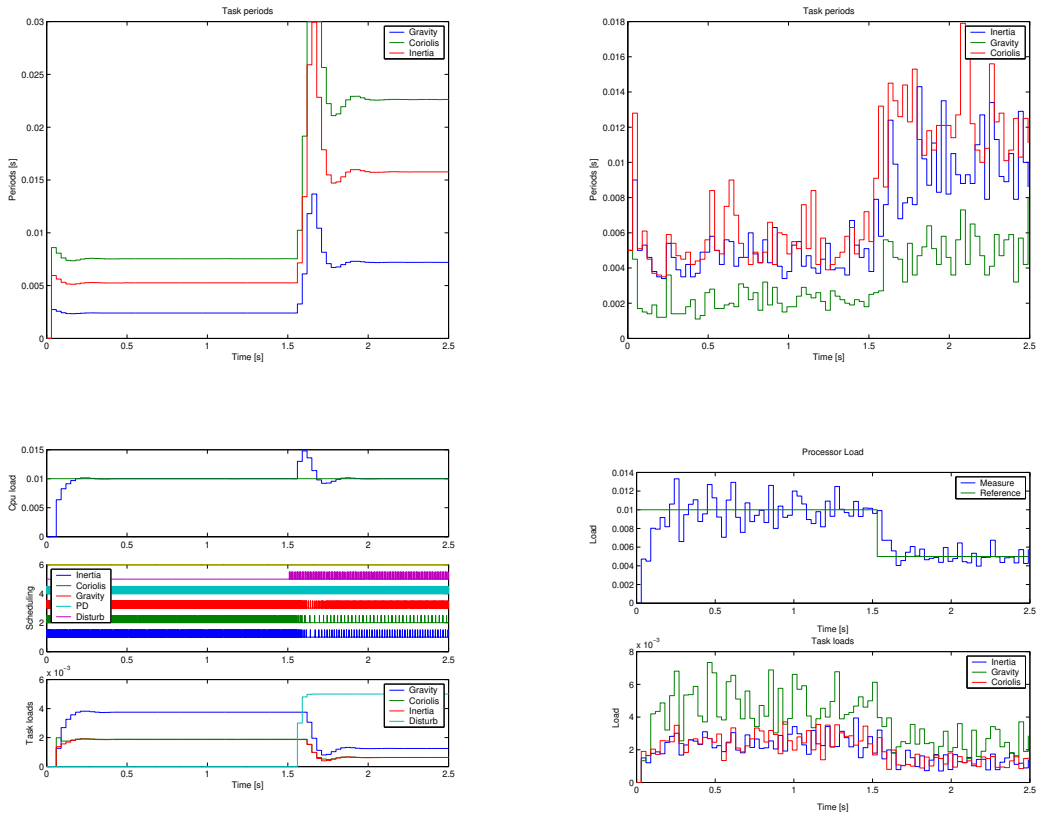

Figure 2. Feedback scheduling in robot control

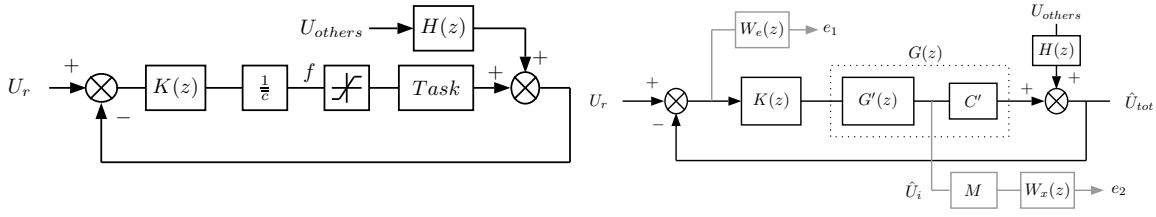

Figure 3. Control scheme for CPU resources

6.4. Automatic generation of correct controllers

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

6.4.1. The control of multimode 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 [33]. 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 6.4.2).

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).

6.4.2. 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 [29]. Work in progress involves fine-tuning algorithms for optimal synthesis along paths, and its application to the control of sequences of reconfigurations.

We considered the taking into account of explicit modes of operations in complex systems, and their association with different control objectives. This is in particular true of fault tolerance policies which can vary according to the allowed degradation. We proposed a method consisting of using conditional invariance objectives.

In parallel, another methodological approach related to fault tolerance concerned the use of controller synthesis in the exploration of the design space, and exhibition of solutions, before a manual implementation [26]. An advantage is that it provides concrete help to a designer, without suffering from the complexity issues as much as the distributed controller synthesis problem.

6.5. Component-based Construction

Participants: P. Fradet, A. Girault, G. Goessler [contact person].

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.

6.5.1. Correctness by construction.

We have developed a general model for component-based construction of real-time systems [17]. Every component — which can itself be composite — consists of three layers. The first one describes the *behavior* of the component. Two kinds of constraints on the integration of components are described by the interaction model and the execution model, represented by two layers on top of the behavior. The interaction model describes the topology of the system and the types of interactions between the components. The execution model specifies constraints relative to scheduling and resource management. A commutative and associative composition operation allows for incremental modeling of the system, and verification of properties on systems that are not yet complete.

We have shown that generalized static and dynamic *priority relations* are expressive enough to describe the safety and scheduling constraints that preserve deadlock freedom. Their composability makes them a powerful tool for incremental modeling [30]. We have proposed results guaranteeing by construction properties like safety, liveness and determinism of components in the composed system. These results have been implemented in the PROMETHEUS tool [43].

6.5.2. Component-based embedded systems.

We have recently started a joint activity on component-based construction of embedded systems, as part of the ACI ALIDECS 8.2.2. The activity includes a survey of existing component-based approaches, the development of methods guaranteeing correctness by construction (in particular based on controller synthesis, compositionality and composability), and the efficient deployment of components and executives ensuring their correct execution. The activity has led to several proposals of master's projects.

6.6. Aspect-oriented programming

Participants: T. Ayav, P. Fradet [contact person], A. Girault, E. Rutten.

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 would otherwise yield 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:

- proving safeguards to the programmer (interaction analysis),
- considering aspects as formal trace properties (scheduling or fault tolerance aspects),
- studying aspects in the simpler and formal setting of transition systems (upon which many synchronous reactive languages are based).

6.6.1. Trace-based AOP and interaction analysis.

We have advocated an approach which — by means of expressive trace-based aspect languages and restrictions on inserts — enables reasoning about different aspect properties [14]. *Trace-Based Aspects* are defined on traces of events occurring during program execution. These aspects are more expressive than those based on atomic points because relations between execution events – possibly involving information from the corresponding execution states — can be expressed.

Aspects are not always orthogonal and aspect interaction is a fundamental problem of AOP. By restricting aspects to regular expressions, we have shown that it is possible to statically detect whether several aspects interact [14]. We have proposed three extensions of our interaction analysis for trace-based aspects [25]:

- Support for variables in aspects. This allows the definition of even more precise aspects and avoid detecting spurious conflicts.
- Generic composition operators for aspects. This enables us to provide expressive support for the resolution of conflicts among interacting aspects.
- Applicability conditions for aspects. This makes interaction analysis more precise and paves the way for reuse of aspects by making explicit requirements on contexts in which aspects must be used.

This work is done in collaboration with Rémi Douence and Mario Südholt from the OBASCO project team at Ecole des Mines de Nantes.

6.6.2. Aspects of scheduling and fusion of networks of components.

We have studied the application of aspects to the efficient implementation of component-based systems [27]. In our approach, a component-based software is represented by a *Kahn Process Network* (KPN). To implement efficiently such networks, we introduce the notion of aspects of scheduling and make use of invasive composition techniques. The network and aspects are woven into a unique sequential, and hopefully, efficient program. More precisely:

- Even if a network of components defines a deterministic program, it has many possible executions. The programmer can define constraints on the execution (sequencing, bounds of FIFOS, etc.) in a scheduling aspect.
- An automatic process (based on static analysis and program transformation techniques) enforces these constraints on the network.
- The scheduling is completed to sequentialize the network and to produce a unique sequential program.

This AOP-inspired technique permits to keep the construction of component-based systems separate from scheduling and efficiency issues. This work is done in collaboration with Stéphane Hong Tuan Ha from the LANDE project team at IRISA/INRIA-Rennes.

6.6.3. Reactive and aspect-oriented programming

Bringing together aspect-oriented programming and reactive programming has two main objectives:

- The first objective is to propose a formal semantics of aspects and weaving in the reactive programming framework. This framework is promising since it is based on simple formal models (transition systems, automata) and a large class of aspects can be seen as temporal properties to enforce on such models.
- Each programming paradigm has its own abstractions, concerns and therefore, aspects. The second objective is to discover and study the specific aspects needed by reactive systems.

This work has been conducted within the local ARC CTRL-A in collaboration with Karine Altisen and Florence Maraninchi from VERIMAG.

6.6.4. Fault-tolerance aspects for real-time software

This line of research, related to the ALIDECS project (see section 8.2.2) has started recently with the arrival of T. Ayav on a post-doctoral position. Our goals are to design a language for specifying fault-tolerance aspects and efficient techniques (based on static analysis, program transformation and/or instrumentation) to weave them into real-time programs.

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 languages issues in the simplest possible setting. We present here work within the λ-calculus (compilation of higher-order sequential languages) and the γ-calculus (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 [13]. We have characterized the essence of the Krivine machine and have located it in the design space of functional language implementations. We have showed that, even within the particular class of Krivine machines, hundreds of variants can be designed. This work is based on the framework that we had previously developed for the systematic study of functional language implementations [42].

This is joint work with Rémi Douence from the OBASCO project team (Ecole 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 often referred to as the Chemical Reaction Model. We have proposed a formal and basic calculus, the γ-calculus [22], 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). In this formalism, the execution of a program can be seen as the evolution of a solution of molecules (*i.e.*, multiset of γ-expressions) which react until the solution becomes inert.

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 [20]. A distinctive feature of our specification was 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.

A summarized account of this research has been presented at the *Unconventional Programming Paradigms* workshop [21]. We are currently working on the design of a more expressive higher-order programming language and its application to Grid and reactive programming.

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

7. Contracts and Grants with Industry

7.1. STMicroelectronics

We have an ongoing collaboration with STMicroelectronics (Crolles), SysArt team. The goal is to extract structured information concerning the execution rates, from a SoC design in the Transaction Level Model, in order to partition the design according to these execution rates. A cooperation in the framework of the Nano 2008 program is being prepared.

8. Other Grants and Activities

8.1. Regional actions

8.1.1. JESSICA

Jessica[29] 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.

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 [30].

8.1.3. Local Arc Ctrl-A

This is a locally funded (by INRIA-Rhône-Alpes) cooperation with VERIMAG (synchronous team, F. Maraninchi, K. Altisen), around the topic *Aspect-oriented programming and reactive languages*[31], presented in Section 6.6.3.

8.2. National actions

8.2.1. ACI "Sécurité & Informatique" project Dispo

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 Ecole 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[33] is to study an integrated development environment for the construction and use of safe embedded components. The consortium includes LIP6 (Paris), 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.

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 titled: *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, approached from a control theory perspective.

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

[29] http://www.jessica-puce.prd.fr
[30] http://www.inrialpes.fr/pop-art/people/simon/c3o/
[31] http://www-verimag.imag.fr/~altisen/CTRL-a/
[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

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. Cooperations internal to 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 notably of 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).

8.2.7. Cooperations with other laboratories

- P. Fradet and E. Rutten cooperate with K. Altisen and F. Maraninchi (VERIMAG).
- A. Girault cooperates with X. Nicollin (VERIMAG), M. Pouzet (LIP6, University of Paris VI), Denis Trystram from (ID-IMAG), and C. Dima (Université of Paris XII).
- G. Goessler cooperates with J. Sifakis (VERIMAG).
- E. Rutten cooperates with P. Amblard (TIMA) and with H. Alla (LAG).
- D. Simon cooperates with O. Sename (LAG).

8.3. European actions

8.3.1. Artist European IST network

ARTIST is a European IST network, lead by VERIMAG. It concerns real-time systems, particularly hard real-time, and adaptive techniques for quality of service management. It lasts 2 years (2002-2004), and funds one year of post-doc on the topic of controller synthesis for fault-tolerance.

8.3.2. Artist 2 European IST network

ARTIST 2 is a European Network of Excellence on embedded system design[36]. It's goal is to establish Embedded Systems Design as a discipline, combining competencies from electrical engineering, computer science, applied mathematics, and control theory. We collaborate as a core partner within the Hard Real Time cluster, lead by Albert Benveniste.

8.3.3. 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).

8.3.4. 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, specially for automobiles. The main goals are: independence between hard and soft, standard components and tools, and cooperation between actors. The PhD of Hamoudi Kalla is funded by this project.

[36] ARTIST 2: http://www.artist-embedded.org/FP6

9. Dissemination

9.1. Scientific community

- P. Fradet has participated in the program committees of ESOP'04 (*13th European Symposium on Programming*), JLFA'04 (*15ème Journées Francophones des Langages Applicatifs*) and JFDLPA'04 (*Première Journée Francophone sur le Développement de Logiciels Par Aspects*). He is in the program committee of FOAL'05 (*Foundations of Aspect-Oriented Languages Workshop*).
  He has co-organized the *Unconventional Programming Paradigms (UPP'04)* workshop from september 14 to 17 in Le-Mont-Saint-Michel. This workshop (on invitation only) was organized under the auspices of the EU (FET programme) and NSF. Around 40 participants presented and discussed research on Autonomic Computing, Amorphous Computing, Generative Computing, Bio-Inspired Computing and Chemical Computing. The pre-proceedings are available [11] and the post-proceedings will be published by Springer in the hot topics subline of LNCS mid-2005.
- A. Girault has participated in the program committees for SLAP'04 and FTRTFT'04, and maintains the *SYNchronous Applications, Languages, and Programs* web site[37]. He has co-organized the SLAP'04 workshop in Barcelona.
- G. Goessler has participated in the program committee of the international workshop SVERTS'04 (Specification and Validation of UML models for Real Time and Embedded Systems).
- E. Rutten co-chairs (with H. Alla, LAG) the programming committee of MSR 05 (*Modélisation des Systèmes Réactifs*) and participates in organizing it, and is in program committees for SLAP'04, ECRTS'04 (*Euromicro Conference on Real-Time Systems*), salon RTS'04 and RTS'05 (*RTS EMBEDDED SYSTEMS*), WPDRTS'04 (*Workshop on Parallel and Distributed Real-Time Systems*, satellite of IPDPS), WODES'04 (*IFAC International Workshop on Discrete Event Systems*), SFEDL'04 (*Semantic Foundations of Engineering Design Languages*, satellite of ETAPS'04).
  He gave an invited talk at the *journée QSL (Qualité et Sûreté de Logiciel)* at INRIA Lorraine in Nancy, 29 october 2004.
  He was in the PhD commissions of :

  – Fabrice Peix, *Distribution de programmes synchrones : le cas d'Esterel*, INRIA-Sophia-Antipolis, Université de Nice-Sophia-Antipolis, 6 july 2004. (reviewer)
  – Bassam Kattan, *Synthèse structurelle d'un contrôleur basé sur le Grafcet*, LAG, INPG/ENSIEG, 3 september 2004.
  – Oulaid KAMACH, *Approche Multi-modèle des SED : application à la gestion des modes*, LAI, INSA Lyon, 16 décembre 2004.

  He is a member of INRIA evaluation commission, INRIA Rhône-Alpes scientific employment commission, Univ. of Brest specialists commission (27th section).
- D. Simon is expert in the Jessica program for technically supporting small companies about embedded and real-time operating systems including Linux. He also has been invited to present a talk on feedback scheduling at the embedded real-time systems day organized by the ACM-SIGOPS french chapter.
- In addition to conferences mentioned in the publications list, we participated in:

  – FDL 04 (Lille, september) and IEEE ASE 04 (Linz, september)
  – Synchron 04 (Dagstuhl, december 2004)

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

9.2. Teaching

9.2.1. Courses

- Alain Girault: compilation project, 2nd year engineering, 51 h., ENSIMAG Grenoble.
- Alain Girault: algorithmics and programming in Java, 26h, INPG Telecom Department.

9.2.2. Advising

PhDs:

– Gwenaël Delaval, co-advised by Alain Girault (with M. Pouzet, LIP6), since 9/2004, PhD in computer science, INPG.

– Stéphane Hong Tuan Ha, co-advised by Pascal Fradet (with T. Jensen, IRISA), since 9/2002, 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.

Masters:

– Negre Amaury, 2nd year ENSIMAG/INPG, co-advised by Daniel Simon and Roger Pissard (SED);

Nicolas Leignel: *Reliable scheduling for the generation of real-time embedded code*, M2R ROCO, INPG/Université Joseph Fourier, co-advised by Denis Trystram and Alain Girault.

Gwenaël Delaval: *Synthèse de contrôleurs discrets pour systèmes embarqués multi-tâche*, M2R S&L, INPG/Université Joseph Fourier, advised by Eric Rutten;

Nicolas Palix: *Component-based modeling and translation of systems on chip*, 3rd year ES-ISAR/INPG and M2R S&L, advisor: G. Goessler;

Safouan Taha: *Synthèse de contrôleurs discrets pour systèmes embarqués tolérants aux pannes*, M2R S&L, INPG/Université Joseph Fourier, co-advised by Alain Girault and Eric Rutten;

Huafeng Yu: *A scheduling/distribution heuristic for real-time embedded systems tolerant to sensor failures*, M2R S&L, INPG/Université Joseph Fourier, co-advised by Alain Girault and Hamoudi Kalla.

Yidong Zhu: *Systèmes réactifs, synthèse de contrôleurs discrets et contrôle de circuit*, TER (Travaux d'Études et de Recherche), Master 1 Informatique, co-advised by Eric Rutten and Paul Amblard (TIMA).

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, April 7 - 11, 2003,Warsaw, Poland", Lecture Notes in Computer Science, n° LNCS 2618, Springer Verlag, 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° 1/2, 2002, p. 55-84.

[3] 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.

[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° 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° 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*, Document d'habilitation à diriger des recherches, Université de Rennes 1, November 2000.

[8] H. MARCHAND, É. RUTTEN, M. LE BORGNE, M. SAMAAN. *Formal Verification of Programs specified with* SIGNAL *: Application to a Power Transformer Station Controller*, in "Science of Computer Programming", vol. 41, n° 1, 2001, p. 85–104.

[9] 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, http://www.inrialpes.fr/iramr/pub/Orccad.

[10] 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

[11] J.-P. BANÂTRE, J.-L. GIAVITTO, P. FRADET, O. MICHEL (editors). *Unconventional Programming Paradigms, pre-proceeding of UPP'04, Le Mont Saint Michel*, To be published in the hot topics subline of LNCS, Springer-Verlag, September 2004.

[12] F. MARANINCHI, A. GIRAULT, M. POUZET (editors). *International Workshop on Synchronous Languages, Programming, and Applications, SLAP'04*, ENTCS, Elsevier Science, Barcelona, Spain, March 2004.

Articles in referred journals and book chapters

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

[14] R. DOUENCE, P. FRADET, M. SÜDHOLT. *Trace-Based Aspects*, in "Aspect-Oriented Software Development", M. AKSIT, S. CLARKE, T. ELRAD, R. E.FILMAN (editors)., Addison-Wesley, 2004, p. 201-217.

[15] A. GIRAULT. *Design of an Hybrid Controller for Autonomous Vehicles Driving on Automated Highways*, in "TRC", vol. 12, n° 6, December 2004, p. 421–452.

[16] A. GIRAULT, H. KALLA, Y. SOREL. *A Scheduling Heuristics for Distributed Real-Time Embedded Systems Tolerant to Processor and Communication Media Failures*, in "IJPR", vol. 42, n° 14, July 2004, p. 2877–2898.

[17] G. GÖSSLER, J. SIFAKIS. *Composition for Component-Based Modeling*, in "SCP, FMCO 2002 special issue", to appear, 2004.

[18] D. SIMON, F. BENATTAR. *Design of Real-Time Periodic Control Systems Through Synchronisation and Fixed Priorities*, in "Int. Journal of Systems Science", to appear.

Publications in Conferences and Workshops

[19] 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.

[20] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Chemical Specification of Autonomic Systems*, in "Proceedings of the 13th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE'04)", July 2004.

[21] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Higher-order Chemical Programming Style*, in "Proceedings of Unconventional Programming Paradigms (UPP'04)", To be published in the Hot Topics subline of LNCS, Springer-Verlag, September 2004.

[22] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Principles of Chemical Programming*, in "Fifth International Workshop on Rule-Based Programming (RULE'04)", Electronic Notes in Theoretical Computer Science, June 2004.

[23] 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.

[24] C. DIMA, A. GIRAULT, Y. SOREL. *Static fault-tolerant scheduling with "pseudo-topological" orders*, in "Joint Conference on Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time

and Fault Tolerant System, FORMATS-FTRTFT'04, Grenoble, France", LNCS, vol. 3253, Springer-Verlag, September 2004.

[25] R. DOUENCE, P. FRADET, M. SÜDHOLT. *Composition, reuse and interaction analysis of stateful aspects*, in "Proc. of the 3rd Int' Conf. on Aspect-Oriented Software Development(AOSD'04)", K. LIEBERHERR (editor)., ACM Press, March 2004, p. 141-150.

[26] 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.

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

[28] A. GIRAULT, H. KALLA, Y. SOREL. *An Active Replication Scheme that Tolerates Failures in Distributed Embedded Real-Time Systems*, in "IFIP Working Conference on Distributed and Parallel Embedded Systems, DIPES'04, Toulouse, France", Kluwer Academic, August 2004.

[29] 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.

[30] 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.

[31] D. ROBERT, O. SENAME, D. SIMON. *Sampling Period Dependent RST Controller Used in Control/Scheduling Co-Design*, in "IFAC 2005 World conference", to appear.

[32] D. SIMON, D. ROBERT, O. SENAME. *Robust Control/Scheduling Co-Design: Application to Robot Control*, in "RTAS'05", to appear.

Internal Reports

[33] 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.

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

[37] P. CASPI, M. POUZET. *Synchronous Kahn Networks*, in "ACM SIGPLAN International Conference on Functional Programming, Philadelphia, Pensylvania", May 1996.

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

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

[40] A. CERVIN, J. EKER, B. BERNHARDSSON, K.-E. AARZÉN. *Feedback-Feedforward Scheduling of Control Tasks*, in "Real Time Systems", vol. 23, nº 1, 2002, p. 25–54.

[41] 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, nº 2, 1986, p. 244-263.

[42] R. DOUENCE, P. FRADET. *A Systematic Study of Functional Language Implementations*, in "ACM Transactions on Programming Languages and Systems", vol. 20, nº 2, 1998, p. 344–387.

[43] 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.

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

[45] 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.

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

[47] 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, nº 1, 2002, p. 85–126.

[48] 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.

[49] 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, nº 3, March 2003, p. 219-254.

[50] 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, nº 4, October 2000, p. 325–346.

[51] 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.

[52] P. J. RAMADGE, W. M. WONHAM. *The Control of Discrete Event Systems*, in "Proceedings of the IEEE", vol. 77, nº 1, 1989.