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 1C

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 **5**
4.1.1. Industrial applications. 5
4.1.2. Industrial design tools. 6
4.1.3. Some of our industrial cooperations. 6
5. Software **6**
5.1. Orccad 6
5.2. Implementations of synchronous programs 8
5.2.1. Code distribution 8
5.2.2. Fault-tolerance 8
5.3. Prototypes 8
5.3.1. Automatic Controller Generation 8
5.3.2. Compositionality 9
5.3.3. Vehicle Control Tasks 9
6. New Results **9**
6.1. Implementations of synchronous programs 9
6.1.1. Distribution 9
6.1.2. Fault-tolerance 9
6.2. Control/scheduling co-design 10
6.2.1. Scheduling for regulation 10
6.2.2. Regulation for scheduling 10
6.3. Automatic generation of correct controllers 12
6.3.1. The control of multi-mode multi-tasking systems 12
6.3.2. Automated generation of property-enforcing layers 12
6.3.3. Fault-tolerant systems 12
6.4. Compositional modeling and analysis 12
6.5. Reactive and aspect-oriented programming 13
7. Contracts and Grants with Industry **14**
7.1. RNTL Automate 14
7.2. ST Microelectronics 14
8. Other Grants and Activities **14**
8.1. Regional actions 14

8.1.1. JESSICA 14
8.1.2. Local ARC Ctrl-a 14
8.2. National actions 14
8.2.1. Groupe COSED 14
8.2.2. CNRS AS 155 of RTP 24: Hybrid systems 15
8.2.3. CNRS RTP 21: Fault-tolerance 15
8.2.4. CNRS RTP 55: Network controlled systems 15
8.2.5. CNRS SAR (Systèmes à retards) group 15
8.2.6. Cooperations internal to Inria 15
8.2.7. Cooperations with other laboratories 15
8.3. European actions 15
8.3.1. ARTIST European IST network 15
8.3.2. EAST-EEA European ITEA project 15
9. Dissemination 16
9.1. Scientific community 16
9.2. Teaching 16
9.2.1. Courses 16
9.2.2. Advising 17
10. Bibliography 17

1. Team

Project leader
Eric Rutten [CR INRIA]

Project assistants
Françoise de Coninck [part-time, until 8/2003]
Elodie Toïhein [part-time, since 9/2003]

Personnel Inria
Alain Girault [CR]
Gregor Gössler [CR]
Pascal Fradet [CR, since 9/2003]
Olivier Sename [Delegation from INPG-ENSIEG, until 8/2003]
Daniel Simon [CR]

Engineers
Serpil Karakas [Expert engineer (RNTL project AUTOMATE), until 8/2003]
Olivier Testa [Associate engineer, until 8/2003]

PhD students
Hamoudi Kalla [INRIA-EEA grant]
David Robert [MENRT EEATS grant, since 10/2003]

Post-doctoral researcher
Emil Dumitrescu [ARTIST Network, since 9/2003]

Interns
Mohamed Abdennebi [March–July, DEA Informatique, ESIA, Annecy]
Ismail Assayad [March–August, DEA ISC, UJF, Grenoble]
Gwenael Delaval [July–September, ENSIMAG, Grenoble]
Safia Iddir [March–July, DEA Control Theory, INPG]
Ivana Medos [July–August, MIT, Boston]
David Robert [March–July, DEA Control Theory, INPG]
Fethi Bouziani [March–June, DEA Control Theory, INPG]

External Collaborators
Karine Altisen [VERIMAG, INPG-ENSIMAG]
Olivier Sename [LAG, INPG-ENSIEG, since 9/2003]

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 badly 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/tick
[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 adaptative scheduling, with regard to quality of service (in relation with the ORCCAD[4] environment);
- high-level design 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 and aspect-oriented programming.

Our applications are in embedded systems, typically in the robotics, automotive, and telecommunications domains. International and industrial relations feature:

- the ITEA European project EAST-EEA, about embedded electronics in cars.
- the IST European network ARTIST, about advanced real-time systems.
- the RNTL-funded Automate project (ATHYS, COMAU/Renault Automation) on safe control components for the automation of assembly production cells.
- Cooperations with ST Microelectronics and France Télécom R&D.

3. Scientific Foundations

3.1. Embedded systems and their safe design

Key words: *Embedded systems, real-time, control, distribution, 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. We contribute 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)[7].

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 labelled transition systems (or finite or infinite state automata) [24][26]. During the years, methodologies for the formal verification [36][28], control synthesis [37] and compilation, and extensions to timed and hybrid systems [33][25] have been developed. Asynchronous models consider the interleaving of events or messages, and are often applied in the field of telecommunications, in

[4] http://www.inrialpes.fr/iramr/pub/Orccad
[5] http://www-verimag.imag.fr/PEOPLE/Florence.Maraninchi/MATOU
[6] http://www.irisa.fr/espresso
[7] http://www.systemes-critiques.org/ARTIST

particular for the study of protocols. A well-known formalism for reactive systems is STATECHARTS [31], which can be encoded in a synchronous model as shown in [2].

The synchronous approach[8] [29][30] to reactive systems design gave birth to complete programming environments, around languages like ARGOS, LUSTRE [9], ESTEREL [10], SIGNAL/POLYCHRONY[11], SYNDEX [12], Lucid Synchrone[13]or Mode Automata[14]. 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 relatively 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 [27][32]; 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

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

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 automation of the use of formal methods. In the field of verification, this concerns in particular the technique of model checking; 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.

[8] http://www.synalp.org
[9] http://www-verimag.imag.fr/SYNCHRONE
[10] http://www.inria.fr/recherche/equipes/tick
[11] http://www.irisa.fr/espresso/Polychrony
[12] http://www-rocq.inria.fr/syndex
[13] http://www-spi.lip6.fr/lucid-synchrone
[14] http://www-verimag.imag.fr/PEOPLE/Florence.Maraninchi/MATOU

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 PhDs 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 [38]) 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

Key words: *dedicated languages, compositionality, distribution, fault tolerance, controller generation.*

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.

3.3.2.1. Implementations of synchronous programs

can be tackled differently depending on the execution platform. Our approach is to obtain, by compilation (thus automatically), founded on a formal model of the program to be implemented:

- 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, in a way that is guaranteed to be correct with respect to the original specification, and optimized;
- fault-tolerance by replication of computations on a multiprocessor architecture, and scheduling of computations according to the faults to be tolerated.

3.3.2.2. Control/scheduling co-design

where the interaction of the very 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

where we apply the technique of discrete controller synthesis, especially by using the tools SIGALI [35] and Mode Automata [34] 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 [5],
- a locally imperative, globally declarative language whose compilation comprises a phase of discrete controller synthesis.

4. Application Domains

Key words: *embedded systems*, *robotics*, *automotive*, *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[15]) 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 [16], SILDEX [17] (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[18]. 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[19].

4.1.3. Some of our industrial cooperations.

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

- ST Micro-electronics 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 towards their ESTEREL compiler;
- ATHYS, where methodological aspect from the ORCCAD approach were taken over, as well as a specialized verification framework.

5. Software

5.1. Orccad

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

ORCCAD [20] 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.

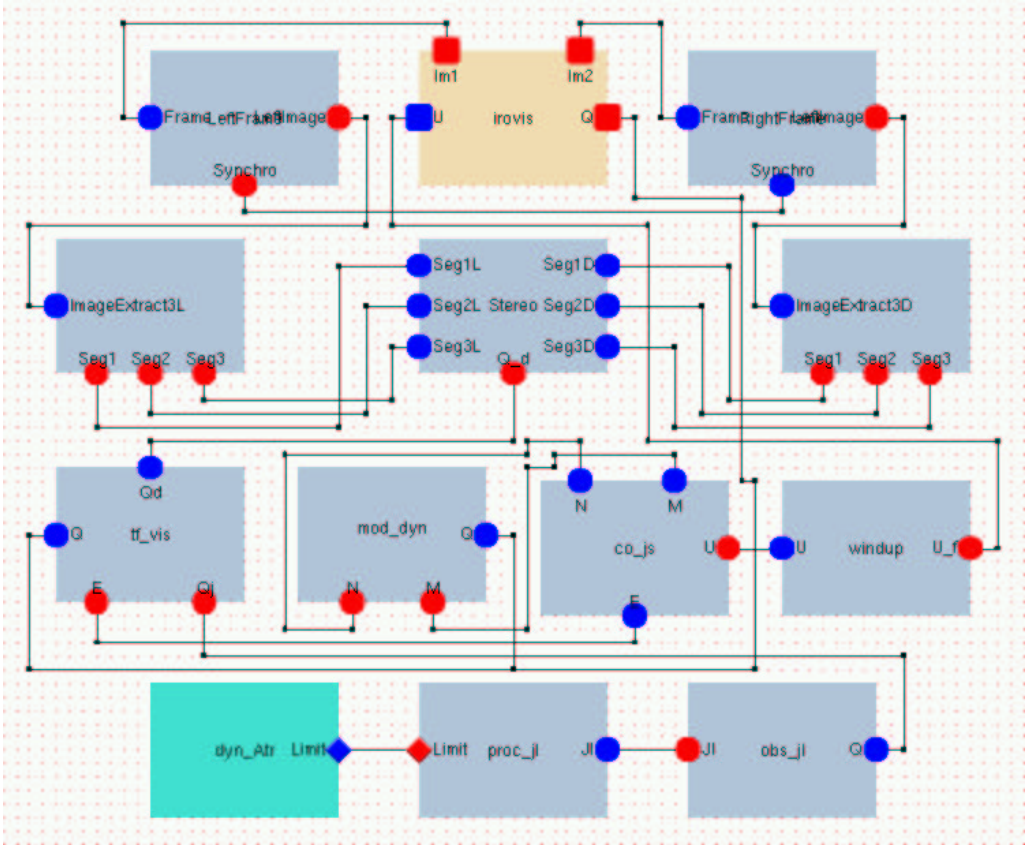

Figure 1. Orccad's GUI for control design.

ORCCAD is maintained by the *Support Expérimentations & Développement (SED)* service of the laboratory. 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. By this we mean that 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[21]. 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 project AOSTE on the subject of fault-tolerance. In particular, we have implemented several new heuristics for fault-tolerance within their software SYNDEX [22]. 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: E. Rutten [contact person], K. Altisen.

On this subject, the development activities are at the beginning, and encompass two aspects.

On the one hand, the software tool for experimentation, allowing the specification of models, the controller synthesis, and the execution or simulation of the results, 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 [23]).

On the other hand, the determination of useful component templates and relevant properties can be given form by libraries of task models, and properties and objectives. To start off, they can be naturally seen in terms

[15] http://www.dspaceinc.com
[16] http://www.esterel-technologies.com
[17] http://www.tni-valiosys.com
[18] http://www.athys.fr
[19] http://www.realtimelinuxfoundation.org/projects/projects.html
[20] http://www.inrialpes.fr/iramr/pub/Orccad
[21] http://www.inrialpes.fr/bip/people/girault/Ocrep
[22] http://www-rocq.inria.fr/syndex
[23] http://www-verimag.imag.fr

of mode automata and the tools of the experimental platform mentioned above, but one can keep in mind their portability towards other platforms.

5.3.2. Compositionality

Participant: G. Gössler.

The first results for compositional modeling and verification (section 6.4) have been implemented in the prototype tool PROMETHEUS, in order to perform case studies to evaluate their potential and limits.

5.3.3. Vehicle Control Tasks

Participants: F. Bouziani, A. Girault [contact person].

Within the framework of our work on the longitudinal control of automatic vehicles [10], we have implemented our control law in the form of an ORCCAD task, which currently runs on the CYCABs of INRIA Rhône-Alpes [21]. We have also designed real-life automated highway simulations with the SHIFT programming language developed at UC Berkeley[24].

6. New Results

6.1. Implementations of synchronous programs

Participants: A. Girault [contact person], H. Kalla, I. Medos, X. Nicollin [INPG, VERIMAG], Y. Sorel [AOSTE, INRIA-ROCQUENCOURT].

6.1.1. Distribution

We have adapted the code distribution method of OCREP to desynchronize LUSTRE programs in order to handle long duration tasks. Such tasks, whose worst case execution time and maximal execution rates are known and bounded, violate intrinsically the synchrony abstraction of LUSTRE. Our distribution method is *clock-driven*, meaning that the program is partitioned according to its clocks, and then desynchronized such that each computing location is scheduled at a different rate, hence allowing long duration tasks to complete without slowing the fast computations [14].

6.1.2. Fault-tolerance

In this field, our work is more precisely aimed at generating a static schedule of a given data-flow graph of tasks onto a distributed heterogeneous architecture, taking into account the execution characteristics of the tasks (resp. data-dependencies) on the processors (resp. communication links) of the architecture. We have worked in three directions:

1. On the **tolerance of processor failures**, we have designed and implemented a new heuristics for SYNDEX, which distributes a data-flow graph of tasks onto an heterogeneous architecture such that the obtained schedule tolerates a given number of processor failures. It performs significantly better than other comparable heuristics found in the literature [12][13].
2. On the **tolerance of communication media failures**, we have conducted a bibliographical study and are now envisioning three research directions depending on the type of communication links (point-to-point or bus) and on the level of replication of the communications (active or hybrid active/passive).
3. On the **generation of reliable schedules**, we have designed and implemented an original bi-criteria heuristics for SYNDEX, aiming at both minimizing the critical path of the distributed schedule, and maximizing its reliability w.r.t. the characteristics of the target heterogeneous architecture [20].

[24] http://www-path.eecs.berkeley.edu/shift

We have also worked on fully distributed algorithms to find multiple disjoint paths in networks of processors, from a given source node to a given destination node. The goal is that whenever a path will fail, a spare one will be available immediately. Here, the difficulty arises because no node knows the complete topology of the network, but only its immediate neighbors. We are currently implementing our distributed algorithm in NS (Network Simulator[25]).

6.2. Control/scheduling co-design

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

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 underutilization 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.

6.2.1. 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 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. This methodology is supported by the development version of Orccad. 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), able to be robust according to timing uncertainties (represented as input delays) [18].

6.2.2. 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) utilization.

The idea consists in 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

[25] http://www.isi.edu/nsnam/ns

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 to better use the computing resources, in particular when the workload changes e.g. due to the admission of a new task. We propose in figure 2 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 task, with a higher priority. It executes as a periodic task, with a period h_S, larger than the sampling periods of the control tasks, in order to change the sampling period only when resource availability changes have been observed.

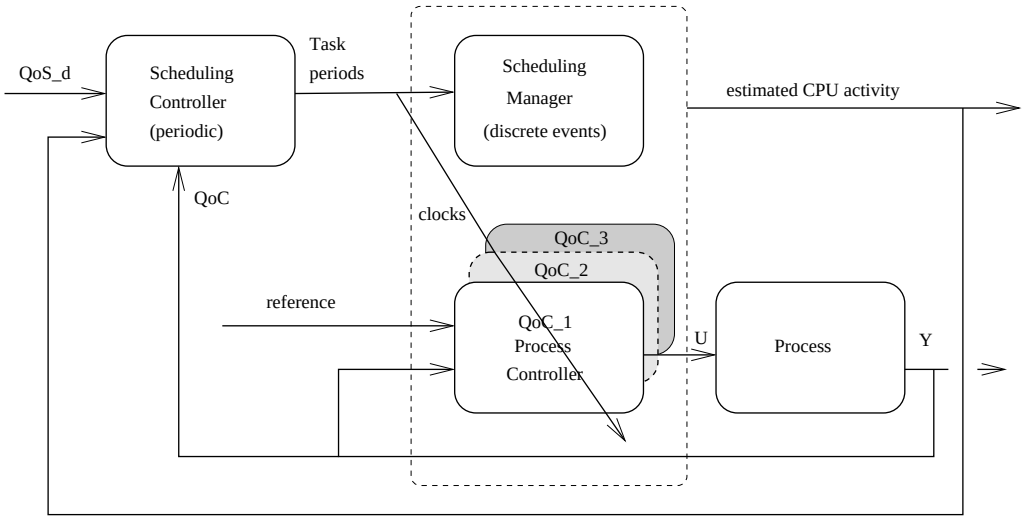

Figure 2. Hierarchical control structure.

Preliminary studies and experiments have been conducted along the following guidelines [18][17][23]:

- while the outer loop should control a composite of QoC and QoS we focussed only on the control of the computing load;
- as the task periods directly affect the computing load they have been chosen as actuators. They can be implemented through software variable clocks;
- several methods for measuring the computing load have been evaluated;
- as timing uncertainties cannot be avoided and are difficult to model or measure, the choice of control algorithms focuses on robustness w.r.t. unknown delays, e.g. using recent results in H_∞ control theory.

Experiments are implemented using a modified ORCCAD runtime under Linux/RTAI. The results show that the method provides both robustness w.r.t. unmodelled delays and a controlled utilization of the computing resource. Further work will study improved versions of robust controllers, a process requirements based formulation of QoC/QoS criteria and a full implementation of the system including QoS management issues.

6.3. Automatic generation of correct controllers

Participants: M. Abdennebi, K. Altisen [VERIMAG], E. Dumitrescu, A. Girault, G. Gössler, E. Rutten [contact person].

6.3.1. 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 [5]. 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, in terms of invariants or configurations that should be always reachable. When a system is modeled by composing instances 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[26]).

We have begun considering the possible complementarities between the application of controller synthesis on the global model, and the use of composition ¨glue¨, also in terms of an automaton, that would enforce a given property between components. The idea is that, for some simple properties, this technique can avoid the costly synthesis, which would however be necessary for others. This issue raises considerations of compositionality as in Section 6.4.

6.3.2. Automated generation of property-enforcing layers

A generalization of the methodology has been formalized and defined as the automated generation of a property-enforcing layer [11]. The component automata model the local constraints; the product of these automata is a first approximation of the set of constraints that should be respected. The constraints that involve several components are expressed as temporal logic properties of this product. We then use general controller synthesis techniques and tools in order to combine the set of communicating parallel automata with the global constraint.

6.3.3. 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, heterogenous 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. The advantage is that, once the system is modeled, it becomes possible to study several fault-tolerance policies [19].

We also approach controller synthesis for fault-tolerance from another angle, related to production systems, in cooperation with LAG[27] (H. Alla). The case study concerned a set of machine-tools, which could operate at full speed in their nominal mode, and at a lower speed when a failure occurred or some tool became worn off [22].

6.4. Compositional modeling and analysis

Participant: G. Gössler.

Component-based modeling is crucial to overcome the complexity of embedded systems. However, two major obstacles need to be addressed: the heterogenous nature of the models, and the lack of results to guarantee correction of the composed system.

[26] http://www.irisa.fr/vertecs
[27] http://www.lag.ensieg.inpg.fr

The technique of model-checking allows to verify or falsify correctness of the system with respect to some property, 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 heterogenous 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 heterogenous nature of the components.

We have developed a general model for component-based construction of real-time systems [1]. The latter are modeled by transition systems. Two kinds of constraints on the integration of components are described by the interaction model and the execution model. 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. We have so far proposed results to guarantee by construction safety, deadlock-freedom of the system, and deadlock-freedom of the components in the system [16][15]. These results are conservative approximations. When they fail to establish correctness, help from other methods such as controller synthesis (section 6.3) may be required. We are therefore interested in combining both approaches.

6.5. Reactive and aspect-oriented programming

Participants: P. Fradet [contact person], 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. For historical reasons, most aspect languages are very expressive and dedicated to object-oriented languages (e.g. Java). The formal foundations of AOP are very difficult to establish in such a complex setting.

Bringing together aspect-oriented programming and reactive programming has three 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.
- The second objective is to study the relationship between aspect weaving and controller synthesis (see Section 6.3). Both techniques aim at ensuring properties on programs. Their common points, differences and potential cross-fertilizations deserve to be studied.
- Each programming paradigm has its own abstractions, concerns and therefore, aspects. So, the last objective is to discover and study the specific aspects needed by reactive systems.

This new line of research is initiated by the arrival of P. Fradet in POP ART last September and by the local ARC with VERIMAG: *Ctrl-a, Aspect-oriented programming and reactive languages* (cf. section 8.1.2). For more information on P. Fradet's on-going work on AOP and other topics, the reader is referred to the activity report of the Lande[28] team (INRIA Rennes).

[28] http://www.irisa.fr/lande

7. Contracts and Grants with Industry

7.1. RNTL Automate

ATHYS is a start-up company of INRIA Rhône-Alpes, working on the techniques for control and application of synchronous languages in robotics; it was created in June 2000, after a year and a half incubation, and its definition was heavily influenced by our experience around ORCCAD.

We cooperate in the RNTL Automate project, also involving Comau (formerly Renault Automation). The basic idea consists in using predefined and validated components to ease and make faster and safer the design of complex applications in industry. Starting from the functional description of each component, e.g. a robot gripper, its discrete event based behavior is modeled and encoded using ESTEREL Studio. Then this behavior can be validated using the associated formal verification tools. Validated components behaviors can be further composed to check the overall behavior of the application, e.g the assembly cell. A set of user-oriented predefined properties has been developed to help the user in the verification process at each step of the design.

This project was funded by the RNTL for one year and 52 kEUR. It valorizes our competence in synchronous techniques and application to production systems, contributes to ATHYS' development of the Cell Control environment and should influence work on domain specific verification and synthesis.

7.2. ST Microelectronics

We have an ongoing collaboration with ST Microelectronics (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. We will apply our synchronous program distribution techniques [14]. An engineering internship and a Masters project should take place in 2004 on this topic.

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 is aimed 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 region.

8.1.2. 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*[30], related to Sections 6.3 and 6.5.

8.2. National actions

8.2.1. Groupe COSED

COSED[31] (*commande opérationnelle des systèmes à événements discrets*) is a working group of the EEA association of Electronics and Control Theory Teachers, seen from an operational point of view, notably PLCs.

[29] http://www.jessica-puce.prd.fr
[30] http://www.inrialpes.fr/pop-art/people/rutten/parties/ARC-loc-ctrl-a/ctrl-a
[31] http://japura.lurpa.ens-cachan.fr/cosed

It is now transformed into the INCOS group (*Ingénierie de la Commande et de la Supervision des SED*) related to GDR MACS, pôle STP[32].

8.2.2. 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.

8.2.3. CNRS RTP 21: Fault-tolerance

We are collaborating to this RTP titled *Sûreté de fonctionnement des systèmes informatiques complexes ouverts*[33].

8.2.4. CNRS RTP 55: Network controlled systems

NECS project related to RTP 55[34].

8.2.5. CNRS SAR (Systèmes à retards) group

O. Sename and D. Simon participate in the SAR group[35].

8.2.6. Cooperations internal to Inria

- The MR2V service at INRIA-RHÔNE-ALPES is maintaining ORCCAD.
- 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 about the applications we make 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) and with J.-P. Banâtre and Y. Radenac (Paris, IRISA/INRIA-Rennes).

8.2.7. Cooperations with other laboratories

- P. Fradet cooperates with R. Douence and M. Südholt (Ecole des Mines de Nantes).
- A. Girault cooperates with X. Nicollin (VERIMAG) and M. Pouzet (LIP6, University of Paris 6).
- G. Gössler cooperates with J. Sifakis (VERIMAG).
- E. Rutten cooperates with K. Altisen and F. Maraninchi (VERIMAG), and with H. Alla (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. 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.

[32] http://www.univ-valenciennes.fr/GDR-MACS
[33] http://www.laas.fr/RTP21-SdF
[34] http://www-lag.ensieg.inpg.fr/canudas
[35] http://www.ec-lille.fr/sar

9. Dissemination

9.1. Scientific community

- K. Altisen and E. Rutten organize a local working group on discrete controller synthesis[36] where researchers meet from VERIMAG, LAG, INRIA-RHÔNE-ALPES, and INSA Lyon, and potentially elsewhere in Rhône-Alpes.
- P. Fradet has participated in the program committees of ESOP'04 (*13th European Symposium on Programming*) and JLFA'04 (*15ème Journées Francophones des Langages Applicatifs*).
- A. Girault has been awarded the prize "Communication et Systèmes"[37] for his work on automatic distribution of synchronous programs.
  He participates in organizing SLAP'03 (*Synchronous Languages, Applications, and Programming*) [9], and in program committees for SLAP'03 and 04 and FTRTFT'04, and maintains the *SYNchronous Applications, Languages, and Programs* web site[38].
- E. Rutten participates in organizing SLAP'03, and in program committees for SLAP'03 and'04, ECRTS'03 and 04 (*Euromicro Conference on Real-Time Systems*), salon RTS'03 and 04 (*RTS EMBEDDED SYSTEMS*), MSR'03 and 05 (*Modélisation des Systèmes Réactifs*), WPDRTS'03 and 04 (*Workshop on Parallel and Distributed Real-Time Systems*, satellite of IPDPS), TSI special issue on Real-Time Systems 03, WODES'04 (*IFAC International Workshop on Discrete Event Systems*), SFEDL'04 (*Semantic Foundations of Engineering Design Languages*, satellite of ETAPS'04).
- O. Sename has participated in the program committee of the IFAC Workshop on Time-Delay Systems TDS'03 (Rocquencourt, France) and of the IFAC Workshop on Automotive Control 2004 (Salerno, Italy).
- D. Simon is expert in the Jessica program for technically supporting small companies. He is a member of INRIA Rhône-Alpes "Commission Postes d'Accueil".
- E. Rutten is a member of INRIA evaluation commission and *détachements* commission, INRIA Rhône-Alpes scientific employment commission, Univ. of Brest specialists commission (27th section).
- In addition to conferences mentioned in the publications list, we participated in:
  - SLAP'03 and ECRTS'03 (Porto, July 2003)
  - Synchron'03 (Marseille, December 2003).

9.2. Teaching

9.2.1. Courses

- Daniel Simon, Alain Girault, Eric Rutten: course on real-time techniques, 18 h. (in all), DEA IVR (Image Vision Robotique) Grenoble;
- Alain Girault, Eric Rutten: compilation, 2nd year engineering, 18 h. (each), ENSIMAG Grenoble.
- Alain Girault: compilation project, 2nd year engineering, 30 h., ENSIMAG Grenoble.

[36] http://www.inrialpes.fr/pop-art/people/rutten/parties/gt-sdc
[37] http://www.c-s.fr
[38] http://www.synalp.org

9.2.2. Advising

PhDs

- 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 10/2003, PhD in Control Theory, INPG.

Masters e.a.

- Ismail Assayad: *Reliable Scheduling Heuristics for Embedded Real-Time Systems*, DEA ISC, UJF-INPG, co-advised by Alain Girault and Hamoudi Kalla;
- MohamedAbdennebi: *Discrete Controller Synthesis for Fault-Tolerant Embedded Systems*, DEA ESIA Annecy, co-advised by Alain Girault and Eric Rutten;
- David Robert: *Robust Discrete/Continuous Control of a Robotic Arm*, DEA Control Theory, UJF-INPG, co-advised by Daniel Simon and Olivier Sename;
- Fethi Bouziani: *Study and Implémentation of Insertion Stratégies for an Automatic Highway*, DEA Control Theory, UJF-INPG, advised by Alain Girault;
- Safia Iddir: *Discrete Controller Synthesis for Fault-Tolerant Production Systems*, DEA Control Theory, UJF-INPG, co-advised by Eric Rutten and Hassane Alla;
- Gwenael Delaval, 2nd year ENSIMAG/INPG, co-advised by Alain Girault and Daniel Simon;
- Ivana Medos, graduate student MIT-Boston, advised by Alain Girault;
- Jean-Christophe Alberti, Abdelmajid Laachachi: TER, Maîtrise Informatique, UJF, co-advised by Gregor Gössler and Eric Rutten.

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" », number 1/2, volume 23, 2002, pages 55-84.

[2] J.-R. BEAUVAIS, E. RUTTEN, T. GAUTIER, R. HOUDEBINE, P. LE GUERNIC, YAN-MEI. TANG. *Modelling Statecharts and Activitycharts as Signal equations.* in « ACM Transactions on Software Engineering and Methodology », number 4, volume 10, October, 2001, pages 397–451.

[3] 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 », number 4, volume 17, 1998, pages 338–359.

[4] P. CASPI, A. GIRAULT, D. PILAUD. *Automatic Distribution of Reactive Systems for Asynchronous Networks of Processors.* in « IEEE Trans. on Software Engineering », number 3, volume 25, May, 1999, pages 416–427.

[5] H. MARCHAND, É. RUTTEN. *Managing multi-mode tasks with time cost and quality levels using optimal discrete control synthesis.* in « Proceedings of the 14th Euromicro Conference on Real-Time Systems, ECRTS'02, *June 19th - 21th, 2002, Vienna, Austria* », pages 241–248, 2002.

[6] 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 », number 1, volume 41, 2001, pages 85–104.

[7] 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 », number 4, volume 1, December, 1993, http://www.inrialpes.fr/iramr/pub/Orccad.

[8] 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

[9] *Workshop on Synchronous Languages, Programming, and Applications.* A. GIRAULT, F. MARANINCHI, E. RUTTEN, editors, series ENTCS, volume 88, Elsevier Science, Grenoble, France, 2003, http://www.elsevier.nl/locate/entcs.

Articles in referred journals and book chapters

[10] A. GIRAULT. *Design of an Hybrid Controller for Autonomous Vehicles Driving on Automated Highways.* in « Transportation Research Part C: Emerging Technologies », 2004, to appear.

Publications in Conferences and Workshops

[11] 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* », series Lecture Notes in Computer Science (LNCS), number 2618, Springer Verlag, pages 174–188, 2003, LNCS nr. 2618.

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

[13] A. GIRAULT, H. KALLA, Y. SOREL. *Une heuristique d'ordonnancement et de distribution tolérante aux pannes pour systèmes temps-réel embarqués.* in « Modélisation des Systèmes Réactifs, MSR'03 », Hermes, pages 145–160, Metz, France, 2003.

[14] A. GIRAULT, X. NICOLLIN. *Clock-Driven Automatic Distribution of Lustre Programs.* in « 3rd International Conference on Embedded Software, EMSOFT'03 », series LNCS, volume 2855, Springer-Verlag, R. ALUR, I. LEE, editors, pages 206–222, Philadelphia, USA, 2003.

[15] G. GÖSSLER, J. SIFAKIS. *Component-Based Construction of Deadlock-Free Systems (Extended Abstract).* in « Proc. FSTTCS'03 », series LNCS, Springer-Verlag, 2003, to appear.

[16] G. GÖSSLER, J. SIFAKIS. *Composition for Component-Based Modeling.* in « Proc. FMCO'02 », series LNCS, Springer-Verlag, 2003, to appear.

[17] O. SENAME, D. SIMON, D. ROBERT. *Feedback scheduling for real-time control of systems with communication delays.* in « ETFA'03 9th IEEE International Conference on Emerging Technologies and Factory Automation », Lisbonne, 2003.

[18] D. SIMON, O. SENAME, D. ROBERT, O. TESTA. *Real-time and delay-dependent control co-design through feedback scheduling.* in « CERTS'03 Workshop on Co-design in Embedded Real-time Systems », ECRTS, Porto, july, 2003.

Miscellaneous

[19] M. ABDENNEBI. *Synthèse de contrôleurs discrets pour systèmes embarqués tolérants aux pannes.* Rapport de DEA Informatique, ESIA, Université de Savoie, Annecy, France, 2003.

[20] I. ASSAYAD. *Heuristique d'Ordonnancement Fiable pour Systèmes Embarqués Temps-Réel.* Rapport de DEA Informatique, EDMI, UJF, Grenoble, France, 2003.

[21] F. BOUZIANI. *Étude et Implémentation de Stratégies d'Insertion pour une Autoroute Automatisée.* Rapport de DEA Automatique, EEATS, INPG, Grenoble, France, 2003.

[22] S. IDDIR. *Synthèse de contrôleurs discrets pour systèmes de production tolérants aux fautes.* Rapport de DEA Automatique, INPG-ENSIEG, Grenoble, France, 2003.

[23] D. ROBERT. *Contrôle/Commande temps-réel robuste d'un bras de robot.* Rapport de DEA Automatique, INPG, Grenoble, France, 2003.

Bibliography in notes

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

[25] E. ASARIN, O. BOURNEZ, T. DANG, O. MALER, A. PNUELI. *Effective synthesis of switching controllers of linear systems.* in « Proceedings of the IEEE », volume 88, 2000, pages 1011–1025.

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

[27] A. CERVIN, J. EKER, B. BERNHARDSSON, K.-E. ARZÉN. *Feedback-Feedforward Scheduling of Control Tasks.* in « Real Time Systems », number 1, volume 23, 2002, pages 25–54.

[28] 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 », number 2, volume 8, 1986, pages 244-263.

[29] N. HALBWACHS. *Synchronous programming of reactive systems.* Kluwer, 1993.

[30] 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* », Springer-Verlag, 1998, LNCS nr. 1427.

[31] D. HAREL. *Statecharts: A Visual Formalism for Complex Systems.* in « Science of Computer Programming », volume 8, 1987, pages 231-274.

[32] C. LU, J.-A. STANKOVIC, G. TAO, S.-H. SON. *Feedback Control Real-Time Scheduling: Framework, Modeling, and Algorithms.* in « Real Time Systems », number 1, volume 23, 2002, pages 85–126.

[33] O. MALER, A. PNUELI, J. SIFAKIS. *On the Synthesis of Discrete Controllers for Timed Systems.* in « Proc. of STACS'95 », series LNCS, volume 900, Springer Verlag, 1995.

[34] F. MARANINCHI, Y. RÉMOND. *Mode-Automata: a new Domain-Specific Construct for the Development of Safe Critical Systems.* in « Science of Computer Programming », number 3, volume 46, 3, 2003, pages 219-254.

[35] 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 », number 4, volume 10, October, 2000, pages 325–346.

[36] J.-P. QUEILLE, J. SIFAKIS. *Specification and Verification of Concurrent Systems in CESAR.* in « proc. International Symposium on Programming », series LNCS, volume 137, Springer-Verlag, pages 337-351, 1982.

[37] P. J. RAMADGE, W. M. WONHAM. *The Control of Discrete Event Systems.* in « Proceedings of the IEEE », number 1, volume 77, 1989.

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