INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team aoste

Models and Methods for the Analysis and Optimization of Systems with Real-time and Embedded Constraints

Sophia Antipolis - Méditerranée, Paris - Rocquencourt

THEME COM

Activity Report

2008

Table of contents

Aoste is a joint team with UNS (University of Nice/Sophia-Antipolis) and CNRS UMR I3S. It is also co-located between Sophia-Antipolis and Rocquencourt. Project members originate from the former INRIA Tick and Ostre teams, together with the I3S Sports team.

1. Team

Research Scientist

Robert de Simone [Team Leader, DR, HdR]
Yves Sorel [Team Vice-leader, DR]
Charles André [UNS Pr, on leave at INRIA from october, HdR]
Frédéric Mallet [UNS MdC, on leave at INRIA]
Dumitru Potop-Butucaru [CR]

Faculty Member

Marie-Agnès Peraldi-Frati [UNS MdC]
Julien DeAntoni [UNS MdC, since October]

Technical Staff

Julien Boucaron [IE]
Benoît Ferrero [IE]
Fadoi Lakhal [IE, until September]
Djamel Louar [IE, until September]
Daniel de Rauglaudre [IR]
Maxence Guesdon [IR, from Development and Experimentation Service]

PhD Student

Anthony Coadou [MESR scholarship]
Omar Kermia [INRIA scholarship]
Jean-François Le Tallec [BDE scholarship with ScaleoChip]
Aamir Mehmood [Pakistani government scholarship]
Patrick Meumeu Yomsi [INRIA scholarship]
Jean-Vivien Millo [BDE scholarship with ST Microelectronics]
François Lagarde [co-directed with CEA-LIST Lise team, until October]

Administrative Assistant

Patricia Lachaume [TCN, on leave from CNRS]
Nelly Maloisel [TRS, until May]
Christine Anocq [TRS, since June]
Sandra Devauchelle [AI]

2. Overall Objectives

2.1. Overall Objectives

Keywords: *AAA methodology*, *Esterel*, *SynDEx*, *SyncCharts*, *SysML*, *Systems-on-Chip*, *UML*, *adequation*, *architectural models*, *code distribution*, *compilation*, *concurrency*, *embedded systems*, *formal methods*, *formal semantics*, *hardware synthesis*, *hardware/software codesign*, *model-checking*, *model-driven engineering*, *multicore*, *multiprocessors*, *optimization*, *program analysis*, *real-time*, *scheduling*, *synchronous reactive formalisms*, *system-level design*.

Modern embedded systems combine complexity and heterogeneity both at the level of applications (with a mix of control-flow modes and multimedia data-flow streaming), and at the level of execution platforms (with increasing parallelism and multicore architectures). Compilation of the application onto the platform then takes the form of an allocation mapping involving spatial distribution as well as temporal scheduling. Formal models and methods may help to establish the correctness and the efficiency of such transformations. Static and dynamic (model-checking) analyses are also used to provide insights regarding prescribed formal semantics.

The main objective of the Aoste team is thus to promote the formal design of embedded systems, with their intrinsic concurrent, distributed and real-time aspects. For this, we develop a model-based approach, where models here have sound and precise operational semantics. For this we build upon previous experience by team members on synchronous reactive formalisms such as ESTEREL and its graphical SYNCCHARTS version, various GALS or polychronous extensions owing to Concurrency Theory (like Process Networks), and the *Algorithm-Architecture Adequation* methodology (AAA) embodied in the SYNDEX environment.

3. Scientific Foundations

3.1. Models of Computation and Communication (MoCCs)

Keywords: *Esterel*, *Event Graphs*, *Kahn networks*, *Process Networks*, *SyncCharts*, *allocation*, *compilation*, *formal verification*, *optimization*, *refinement*, *scheduling*, *synchronous formalisms*, *synthesis*.

Participants: Charles André, Julien Boucaron, Anthony Coadou, Liliana Cucu [EPI TRIO], Robert de Simone, Jean-Vivien Millo, Dumitru Potop-Butucaru, Yves Sorel.

Because of their formal semantics, the various Models of Computation and Communication (MoCCs) considered in our team can be used in a true effective design flow methodology based on model transformation to represent compilation, synthesis, analysis and optimization of concurrent embedded applications onto parallel and multicore embedded architectures. Allocation seen in that sense comprises a physical distribution/placement as well as a temporal scheduling aspect. Timing constraints and requirements may be expressed and have to be checked and preserved in the process.

This type of incremental design flow may be applied to represent a number of existing, theoretical or practical approaches to the design of embedded systems.

3.1.1. Synchronous reactive formalisms

In synchronous reactive models the various concurrent processes all run at the speed of a common global logical clock, which sets up the instantaneous reaction step. Synchronous formalisms provide an accurate representation of both hardware and scheduled embedded concurrent software; in both cases, simultaneous behaviors in a single global instant are allowed, and even often required.

Examples include ESTEREL/SYNCCHARTS, LUSTRE/SCADE, and SIGNAL/POLYCHRONY. ESTEREL and SYNCCHARTS are control-oriented state-based formalisms [41], [38], while *Lustre* and *Signal* are declarative data-flow based formalisms. Synchronous formalisms were discussed in many articles and book chapters, amongst which [45], [47], [40], [10], [2], [6].

The INRIA spin-off Esterel-EDA now develops and markets the industrial versions ESTEREL STUDIO and SCADE together with their programming environments.

3.1.2. GALS and multiclock extensions

The purely single-clock synchronous formalisms often prove to be excessively demanding for users to write large systems descriptions, with different clock domains. Independent logical clocks may be used to represent (total or partial) asynchrony amongst concurrent processes. Globally-Asynchronous/Locally-synchronous (GALS) and polychronous/multiclock models are handy extensions to provide flexibility and modularity in system design. The recently proposed theory of latency-insensitive design (LID) with elastic time is a good example of such an approach: specific protocol elements may be inserted between existing "black-box" IP block components, at a subsequent design time, to make them comply with imperative latencies on the global communications.

In any case the basic synchronous model remains the basic semantic level for behaviors, where the reaction step is defined. But natural properties (such as endochrony/asynchrony) allows to view the GALS and multiclock descriptions as higher-level versions with a natural synchronous interpretation provided by simple scheduling. The monoclock version is then obtained by dedicated scheduling techniques known as *clock calculus*.

3.1.3. Process Networks

The previous model extensions were often calling for general results from branches of Theoretical Computer Science and Concurrency Theory such as Process Networks and Process Calculi. Process Networks comprise Petri Nets and Kahn Networks, as well as various specializations and generalizations, such as Event/Marked Graphs, Data-Flow Domains (Synchronous, Boolean, CycloStatic, CycloDynamic,...), while Process Algebras such as CCS and CSP gave rise to extensions with simultaneous events in SCCS and Meije. We held former background in this field, and more generally in the use of formal operational semantics in the design of verification and analysis techniques for such systems. We bridged the gap in using such models to study techniques for optimized placement and (static) scheduling of models. The specific features of hardware targets let us phrase these questions in a specific context, where ad-hoc ultimately periodic regimes can be established.

3.1.4. Static k-periodic scheduling and routing

Following our *time refinement* approach, early untimed causal models may be transformed into multiclock or GALS ones, then precisely scheduled to a uniform single time. This type of approach is used for instance in the static, k-periodic scheduling of dataflow process networks such as Event Graphs [42], [39], or Synchronous DataFlow graphs [49] and various extensions in UC Berkeley's Ptolemy. We extended the approach by providing means for the designers to provide his/her own time constraints on a given modeling framework, and to express the actual refinement from a given (abstract) time frame to another, more concrete one.

This theory of modulo and k-periodic static scheduling for process networks (mostly Marked/Event Graphs) recently got a renewal of interest due to its application in the context of Latency-Insensitive Design [43] of SoCs. The nature of communication channels, used there for interconnect fabric, demands optimal buffer/place sizing, with corresponding flow control. We contributed several results in this direction, with fine characterization of optimal algorithmic techniques to provide such ultimately k-periodic schedules. They are progressively implemented in the K-PASSA prototype software, described in 5.2.

3.1.5. AAA models

The AAA (*Algorithm-Architecture Adequation*) methodology which is intended for optimizing distributed real-time embedded systems relies on three models.

The Algorithm model is an extension of the well known data-flow model from Dennis [44]. It is a directed acyclic hyper-graph (DAG) that we call "conditioned factorized data dependence graph", whose vertices are "operations" and hyper-edges are directed "data or control dependences" between operations. The data dependences defines a partial order on the operations execution. The basic data-flow model was extended in three directions: first infinite (resp. finite) repetition of a sub-graph pattern in order to specify the reactive aspect of real-time systems (resp. in order to specify the finite repetition of a sub-graph consuming different data similar to a loop in imperative languages), second "state" when data dependences are necessary between different infinite repetitions of the sub-graph pattern introducing cycles which must be avoided by introducing specific vertices called "delays" (similar to z^{-n} in automatic control), third "conditioning" of an operation by a control dependence similar to conditional control structure in imperative languages, allowing the execution of alternative subgraphs. Delays combined with conditioning allow the programmer to specify automata necessary for describing "mode changes".

The Architecture model is a directed graph [4], whose vertices are sequential machines of two types: “processor” (computation resource or sequencer of operations) and “medium” (communication resource or sequencer of communications), and whose edges are directed connections.

The implementation model is also a directed graph, obtained through an external compositional law, where an architecture graph operates on an algorithm graph in order to give, as a result, a new algorithm graph, which corresponds to the initial algorithm graph, distributed and scheduled according to the architecture graph [46].

3.1.6. Distributed Real-Time Scheduling and Optimization

We adress two main issues: monoprocessor real-time scheduling and multiprocessor real-time scheduling where constraints must mandatorily be met otherwise dramatic consequences may occur (hard real-time) and where resources must be minimized because of embedded features.

In our monoprocessor real-time scheduling work, beside the classical deadline constraint, often equal to a period, we take into consideration dependences beetween tasks and several, possibly related, latencies. A latency is a generalization of the typical “end-to-end” constraint [3]. Dealing with multiple real-time constraints raises the complexity of that issue. Moreover, because the preemption leads to a waste of resources due to its approximation in the WCET (Worst Execution Time) of every task as proposed by Liu and Leyland [50], we first studied non-preemtive real-time scheduling with dependences, periodicities, and latencies constraints. Although a bad approximation may have dramatic consequences on real-time scheduling, there are only few researches on this topic. We have been investigating preemptive real-time scheduling since few years, but seeking the exact cost of the preemption such that it can be integrated in schedulability conditions, and in the corresponding scheduling algorithms. More generally, we are interested in integrating in the schedulability analyses the cost of the RTOS (Real-Time Operating System), for which the exact cost of preemption is the most difficult part because it varies according to the instance of each task. Finally, we investigate also the problem of mixing hard real-time and soft real-time constraints that arises in the most complex applications.

The second research area is devoted to distributed real-time scheduling with embedding constraints. We use the results obtained in the monoprocessor case in order to derive solutions for the problem of multiprocessor (distributed) real-time scheduling. In addition to satisfy the multiple real-time constraints mentioned in the monoprocessor case, we have to minimize the total execution time (makespan) since we deal with automatic control applications involving feedback. Furthermore, the domain of embedded systems lead to solve minimization resources problems. Since these optimization problems are of NP-hard complexity we develop exact algorithms (B & B, B & C) which are optimal for simple problems, and heuristics which are sub-obtimal for realistic problems corresponding to industrial needs. We proposed a family of very fast “greedy” heuristics [46] whose results are improved with local neighborhood heuristics, or used as initial solutions for metaheuristics such as variants of “simulated annealing”.

3.2. UML support and Marte Time model

Keywords: *SysML*, *UML*, *executable specification*, *model-driven engineering*, *profile*.

Participants: Charles André, Julien DeAntoni, Robert de Simone, Frédéric Mallet, Marie-Agnès Peraldi-Frati, Yves Sorel.

We promote a model-driven engineering approach for embedded system design based on formal semantics, models and methods. The range of models meant is listed in 3.1. They all consist mainly in hierarchical state diagrams and dataflow/activity diagrams for behavior, and component diagrams with connection ports for compositional structure. This brought to light the idea of using the OMG UML formalisms for graphical representation of models, as it contains all these modeling views.

Still, our considered range of models differ mainly by their timely semantics, ranging from fully asynchronous to monoclock synchronous through intermediate multiclock/polychronous versions. The original UML semantics only considers asynchronous event-based simulation behaviors, mostly because the formalisms does not consider *time* as an essential feature. Whenever time is (scarcely) introduced in the original standard, it is as a non-functional aspect irrelevant to the described semantics.

As part of the MARTE profile for *Modeling and Analysis of Real-Time systems* we introduced a *time model*, which allows to precisely describe the various time threads (or TimeBase, producing logical clocks). These clocks are used to define the precise temporal semantics of the global model. Thus, time becomes a functional aspect, essential to comprehend the proper dynamics of the system.

We thus have defined a specification language CCSL for expressing Clock Constraints. It can be used as input to the TIMESQUARE prototype tool, described in 5.1, to extract feasible simulations, and applied onto UML representation models as part of the MARTE OMG profile.

Then common Models of Computation and Communication (MoCCs) can be built on top of these constructs, to be used directly by the final user. The advanced profile features are aimed primarily at advanced designers and semanticians willing to devise accurate time patterns.

Following the AAA (*Algorithm-Architecture Adequation*) methodology [51], MARTE promotes independent modeling of *applications* called *algorithm* and *embedded platforms* called *architecture* in a first step. The mapping (spatial and temporal) of applications onto embedded platforms is realized only in a subsequent step, through distributed and real-time scheduling analysis and optimizations, relative to the timing constraints and resource costs involved.

MARTE was started as a joint action of Thales, CEA-List and INRIA in their CARROLL collaborative program. The profile RFP (Request For Proposals) was voted early 2005, the initial submission in June 2007, and the (first complete) revised version in middle 2008. The profile is currently undergoing final revision in the Ad-Hoc Finalization Task Force at OMG.

3.3. Modeling standards in embedded system design

Keywords: *AADL*, *AutoSar*, *Esterel*, *SPIRIT IP-Xact*, *SysML*, *SystemC TLM*, *UML Marte*.

Participants: Charles André, Frédéric Mallet, Marie-Agnès Peraldi-Frati, Aamir Mehmood Khan, Jean-François Le Tallec, Julien DeAntoni, Robert de Simone.

The field of model-driven engineering of hardware/software embedded systems is hosting a number of ad-hoc standards dedicated to specific domains. These standards are instrumental in shaping up the technical and economic activities of model exchange between various industrial and academic partners. The main ones considered in our team are:

AADL for avionic systems;

AutoSar for automotive systems;

IP-Xact for System-on-Chip design;

SystemC, Esterel as Electronic System-Level modeling and programing languages.

These standards may be helpful in performing a number of analyses, such as early component integration, performance/schedulability analysis, and so forth. We conducted a number of comparative studies establishing how generic and specific concepts embodied in these standards could be reflected in MARTE, thereby allowing model transformations and exchanges, in a domain-agnostic fashion.

4. Application Domains

4.1. Multicore System-on-Chip design

Synchronous formalisms and GALS or multiclock extensions are natural model representations of hardware circuits at various abstraction levels. They may compete with HDLs (Hardware Description Languages) at RTL and even TLM levels. The main originality of languages built upon these models is to be based on formal *synthesis* semantics, rather than mere simulation forms.

The flexibility in formal Models of Computation and Communication allows to specify modular Latency-Insensitive Designs, where the interconnect structure is built up and optimized around existing IP components, respecting some mandatory computation and communication latencies prescribed by the system architect. This allows a real platform view development, with component reuse and timing-closure analysis. The design and optimization of interconnect fabric around IP blocks transform at modeling level an (untimed) asynchronous versions into a (scheduled) multiclock timed one.

Also, Network on Chip design may call for computable switching patterns, just like computable scheduling patterns were used in (predictable) Latency-Insensitive Design. Here again formal models, such as Cyclo-static dataflow graphs and extended Kahn networks with explicit routing schemes, are modeling elements of choice for a real synthesis/optimization approach to the desig of systems.

Multicore embedded architecture platform may be represented as Marte UML component diagrams. The semantics of concurrent applications may also be represented as Marte behavior diagrams embodying precise MoCCs. Optimized compilation/synthesis rely on specific algorithms, and are represented as model transformations and allocation (of application onto architecture).

Our current work aims thus primarily at providing Theoretical Computer Science foundations to this domain of multicore embedded SoCs, with possibly efficient application in modeling, analysis and compilation wherever possible due to some natural assumptions. We also deal with a comparative view of Esterel and SystemC TLM for more practical modeling, and the relation between the Spirit IP-Xact interface standard in SoC domain with its Marte counterpart.

4.2. Automotive and avionic embedded systems

Model-Driven Engineering is progressively pertaining to these fields. The formalisms AADL (for avionics) and AutoSar [48] are providing support for this, unfortunately not always with a clean and formal semantics. Yet, some interesting issues are involved there in the mix of event-triggered and time-triggered processing means, the various related protocols, and the coexistence of periodic and aperiodic tasks, with distinct periodicity if ever. The process of scheduling and allocation of multiple heterogeneous and communicating applications onto complex embedded architectural paltforms require adequate model and synthesis/analysis/verification techniques to help the designers converge to acceptable solutions.

5. Software

5.1. TimeSquare

Participants: Charles André [correspondant], Benoît Ferrero, Frédéric Mallet, Julien DeAntoni.

TimeSquare is a software environment for modeling and analyzing timed systems. It supports an implementation of the Time Model introduced in the MARTE UML profile (see Sec.3.2), and its companion Clock Constraint Specification Language (CCSL).

TimeSquare has four main functionalities:

1. interactive clock-related specifications, through dialog boxes,
2. clock constraint checking,
3. generation of a consistent temporal structure, using a Boolean solver,
4. displaying and exploring waveforms, written in the IEEE standard VCD format.

TimeSquare is a plug-in developed with Ganymede Eclipse Modeling Tools. ANTLR for constraint parsing, and JavaBDD for the solver are also used. It is integrated in the OpenEmbeDD platform and can be downloaded from the team site (http://www-sop.inria.fr/aoste/dev/time_square).

5.2. K-Passa

Participants: Julien Boucaron [correspondant], Anthony Coadou, Jean-Vivien Millo, Robert de Simone.

This software is dedicated to the simulation, analysis, and effective regular scheduling of Event/Marked Graphs, SDF and KRG extensions. A graphical interface allows to edit the Process Networks and their time annotations (*latency, ...*). Linear programming as well as simulations and other analytic methods allow to compute the necessary addition of integer and fractional extra latencies/registers. Critical cycles are displayed. In the case of KRG the (ultimately k-periodic) routing patterns can also be input and transformed.

K-PASSA currently relies in part on the MASCOPT libary for graph algorithms developed in the Mascotte EPI. It also uses Ilog's CPLEX as its underlying linear constraint solver.

This software was developed as a result of researches on Latency-Insensitive Design conducted in the context of the CIM PACA initiative, with the support of industrial partners providing motivations.

5.3. SynDEx

Participants: Maxence Guesdon, Yves Sorel [correspondant], Cécile Stentzel.

SynDEx is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. It can be downloaded free of charge, under copyright, at the url: http://www.syndex.org.

The AAA methodology requires the specification of 3 main ingredients: an application algorithm, an architectural platform, and real-time features or requirements regarding their combination. Given these, SYNDEX will explore the space of possible allocations (distribution and scheduling) from application elements to architecture resources and services to match the real-time requirements, using heuristic techniques. It will generate automatically distributed real-time code running on the embedded platform.

Application algorithms can be edited graphically as directed acyclic task graphs (DAG), or they may be obtained by translation from various sources, such as (formal) synchronous reactive languages, Scilab/Scicos and UML/MARTE.

Architectures are represented as graphical block diagrams composed of programmable (processors) and non-programmable (ASIC, FPGA) computing components, interconnected by communication media (shared memories, links and busses for message passing).

Application real-time features for elementary operations, relative to hardware component, range amongst *execution and transfer time, period, memory, etc*. Requirements are generally constraints on latency, throughput, etc.

Exploration of alternative allocations of the algorithm onto the architecture may be performed manually or with the help of optimization heuristics; results are visualized as timing diagrams simulating the distributed real-time implementation.

Implementation deployments use dedicated distributed real-time executives, or general purpose real-time operating systems such as Linux/RTAI or Osek for instance. These executives are deadlock-free, based on off-line scheduling policies. Dedicated executives induce minimal overhead, and are built from processor-dependent executive kernels. Presently, executives kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be ported at reasonable cost following these patterns.

5.4. SAS

Participants: Daniel de Rauglaudre, Patrick Meumeu Yomsi, Yves Sorel.

The SAS (Simulation and Analysis of Scheduling) software allows the user to perform the schedulability analysis of periodic task systems in the monoprocessor case. The parameters of the system can either be set using the graphical interface, or from a file. The main contribution, compared to other commercial and academic tools of the same kind, is that SAS takes into account the exact cost of preemption during the schedulability analysis. Beside the usual deadline constraints, the user may also set other constraints such as precedence, strict periodicity, and latency. Several scheduling policies can be selected: Deadline Monotonic, Rate Monotonic, particular priorities set by the user, and a specific version of Audsley's algorithm. The classical Audsley's algorithm is useless and no longer optimal when the cost of preemtion is considered. The specific version is optimal and shows all the possible solutions.

Once the parameters are loaded, the user launches the simulation and the analysis either step by step, or for the whole set of tasks according to the constraints. The resulting schedule is displayed as a typical Gantt chart with a transient and a permanent phase, or as a disk named "dameid" that clearly shows the idle slots of the processor in the permanent phase. For systems with a large least common multiple of periods it is possible to zoom parts of the results.

When the system is schedulable, the following results are displayed: classical utilization factor, permanent exact utilization factor, preemption cost in the permanent phase, and the worst response time for each task. An extra graphic can be displayed showing the response time in function of the time. All the graphics can be converted into PostScript for printing.

The software has been written in OCAML, using CAMLP5 (syntactic preprocessor) and OLIBRT (a graphic toolkit under X) both written by Daniel de Rauglaudre.

6. New Results

6.1. Optimal balanced scheduling of formal modeling of latency-insensitive systems

Participants: Jean-Vivien Millo, Benoît Ferrero, Robert de Simone.

Our former work on Latency-Insensitive Design of SoC led us to consider formal models of interconnected IP components, as Process Networks and Marked/Event Graphs with latency information. We studied dynamic, then static off-line scheduling techniques for such models, owing to ultimately k-periodic regular regimes.

This year we established that the naive computation of schedules by asap execution from *any* initial data allocation could suffer from several misfunctions. First, the global throughput could be impaired by back pressure flow control mechanisms; second, the interconnect channel utilization could sometimes be suboptimal. Even if these phenomena could be considered as extremely rare, we wanted to obtain a theoretically optimal schedule allocation.

We then studied the theory of so-called *balanced* periodic binary words, originally due to Christoffel (1885) and Bernouilli (1776). We introduced two specific operations on such words: *rotations*, representing integral delays between activations of successive computation blocks; *transpositions*, representing (residual) fractional delays. We established deep structure theorems linking both. With this mathematical apparatus we were able to prove that balanced schedules could be imposed and correctly propagated across the process network in a way that every computation node executes exactly when input data occur at its entry ports. The scheduling produced requires also exactly the same number of data as originally present in the model, so that the full system is completely consistent. These results were presented in Jean-Vivien Millo's PhD thesis, defended in December [12].

The last remaining problem now consists in finding a “short” asynchronous sequence of execution steps by individual IP components reaching a balanced data marking from any original initial one.

6.2. K-periodically Routed extended Event Graphs

Participants: Julien Boucaron, Anthony Coadou, Robert de Simone.

The Process Network models based on Event Graphs and varying synchronous or asynchronous interpretations allow powerful results in static scheduling and distribution allocations. Nevertheless they always postulate a uniform data flow. We tried to relax this strong assumption, while preserving the determinism/confluence of computations, as in Kahn Process Networks. Several models were inspirational (such as Boolean and CycloStatic DataFlow Graphs). But the main originality of our KRGs is to rely on two *Select/Merge* operator nodes only (for *mux/demux* effects on data streams). And, *most importantly* the switching patterns for these conditional nodes have to be infinitary periodic binary words, thus using exactly the same description formalisms as our previous *schedule words*.

We have proven a number of powerful algebraic results on such models. They may best be understood by analogy with Boolean Algebra, and the existence of normal forms (such as sum of products, or product of sums). Here, the expansion and factorization of expressions and variables amounts to sharing and unsharing of links and channels in the interconnect fabric representation of the communications across the networks.

Our main results during this year are reported in [27] (submitted for publication). We believe this model of KRGs to be an important step for the Theoretical Computer Science modeling of modern Networks-on-Chip *NoCs*.

6.3. Time Model for UML

Participants: Charles André, Frédéric Mallet, Julien DeAntoni, Robert de Simone, Benoît Ferrero.

The Marte time model, part of the OMG standards, is now in its Finalization Task Force (FTF) phase. The latest Marte specification has been released in August 2008. Frédéric Mallet attends all the OMG technical meetings, actively contributing to Marte and SysML standardizations and he is in charge of the convergence between the two specifications. He also participates with Charles André in the issue resolutions related to Time and Allocation chapters of Marte.

The main concepts of the Time model and their UML representations have been presented at MoDELS'07 [1]. The applicability of Marte has been demonstrated by a study [17], in collaboration with Thales and the CEA.

Besides this standardization effort, Aoste promotes a *timed causality model for UML* addressing both semantic and pragmatic points of view. A formal semantics for CCSL has been proposed [13], [26]. The TimeSquare environment, described in Sec.5.1, has been developed. It fully supports the Marte Time Model, CCSL, and timed simulations [14]. We plan to use this simulation capability in the ANR project RT-SIMEX that will start on January 1st, 2009.

6.4. Time in Models of Computation and Communication

Participants: Charles André, Julien DeAntoni, Robert de Simone, Frédéric Mallet, Marie-Agnès Peraldi-Frati.

6.4.1. Executable timed specifications

The Marte Time Model together with its constraint language (CCSL) provides a way to specify timed causality in a UML model. More generally, this broadly expressive time model is devised to define models of computations and communications (MoCC) by extending the UML. The expected benefit of relying on the UML is to reuse existing graphical editors and integrate various MoCC within the same framework. Classical UML models like state machines or activities can then be executed according to the specific semantics of a given MoCC. For instance, UML state machines can be given a synchronous semantics to behave like SyncCharts, and UML activities can behave like Scade dataflow diagrams.

Towards this goal, equivalent Marte/CCSL constructs have been defined in Signal and in Time Petri nets [36], [29]. These two well-known languages have been chosen as representatives of synchronous and asynchronous languages commonly used in the domain for formal analysis.

Besides, the SOS semantics of CCSL enables the creation of new specific tools. For instance, we have built TimeSquare [34], a java-based Eclipse plug-in that provides assistance to construct Marte/CCSL models and run simulations. Following a Model-Driven Engineering process, we have also provided a metamodel of CCSL whose formal semantics has been weaved directly in the models by using KerMeta. This metamodeling work widens the scope of application of CCSL and should help to link a specific model with its MoCC.

6.4.2. Model interpretation via CCSL semantics

CCSL has been used as a pivot language to give a timed semantics to several formalisms from different application domains:

- In the **automotive** domain, AutoSar is a broad scope emerging standard. EAST-ADL2 (Electronic Architecture and Software Tools) is an architecture description language that provides a structure for the engineering information involved in automotive software development. Aoste contributes to this effort through the MeMVaTEx project (Sec.8.2.3). By defining transformations from East-ADL timing requirements to CCSL relations, we gave an operational semantics to East-ADL, which makes the requirements executable [37], [15].
- In the **avionic** domain, we considered the SAE standard AADL. Indeed, one of the Marte requirements was to integrate UML and AADL models. Aoste has studied the AADL communication model, which mixes time-triggered and event-based messages, and has defined systematic transformation rules to CCSL relations [25], [22], [21] thus allowing Marte models to be partially analyzed by schedulability analysis tools often used with AADL (like Cheddar).
- In the **Electronic Design Automation** (EDA) domain, IP-Xact is *the* standard for describing and handling *intellectual properties* (IP). IP-Xact describes static properties of IP including type of ports, usage of the address space, interconnections, etc, but it entirely relies on programming languages such as Verilog-HDL or SystemC for the description of the behavior. Early analysis of systems of IPs requires an abstract description of timing and behavioral aspects. UML can provide both structural and behavioral aspects. Following a MDE approach, Aoste has proposed a metamodel for IP-Xact, defined a dedicated UML profile, and provided an automated model transformation from UML (profiled with Marte and our IP-Xact profile) to IP-Xact [16].
- On a methodological plane, F. Lagarde in his thesis [11] has studied UML profiles as possible candidates to Domain Specific Languages (DSL) design. He has defined *metrics* for profile asssessment and proposed a set of dedicated high-level constructs that embody OCL expressions to constrain models on which several profiles are applied [20]. Another contribution, in collaboration with F. Mallet and C. André, is the use of a *multilevel paradigm* (i.e., not restricted to the Class-Instance relationship). This approach has brought a fresh view to the Marte Time profile [28], a posteriori justifying the existence of two distinct stereotypes (ClockType and Clock) for the Clock concept.

6.5. Preemptive real-time scheduling in the monoprocessor case

Participants: Liliana Cucu [EPI TRIO], Patrick Meumeu, Thomas Ponchant, Dumitru Potop-Butucaru, Yves Sorel.

Last year, we proposed a necessary and sufficient schedulability condition for a system of periodic tasks, all released at the same time (simultaneous) and scheduled according to *RMA* while taking into account the exact cost due to preemption. Then, we extended this result to the case of the scheduling problem which consists of periodic tasks with precedence and strict periodicity constraints and with periods not necessarily forming an harmonic sequence.

This year, we have introduced a new model in order the solve the general scheduling problem of hard real-time systems with various kinds of constraints such as precedence, strict periodicity, latency and jitter while taking into account the exact cost due to preemption for any scenario of first release for all tasks (simultaneous or not simultaneous). The schedulability analysis is based on the definition of a binary scheduling operation $\oplus$ whose the operands are called *otasks*. We have showed the correspondance between an otask and a periodic task. Now, as a convention in the definition of operation $\oplus$, the left-hand otask is always assigned the highest priority and the right-hand otask the lowest, thus $\oplus$ is not commutative. Each otask is an ordered multiset which consists of either a finite or an infinite sequence of symbols belonging to a finite set called the *generator*. For the generator $\{a, e\}$, symbol "a" always corresponds to an *available time unit* and symbol "e" corresponds to an *already executed time unit* when it belongs to the left-hand otask and to an *executable time unit* when it belongs to the right-hand otask of $\oplus$. For a given set of otasks, operation $\oplus$ is used as many times as there are tasks in the system thanks to its associativity. We have considered two approaches. First, the priority of each otask belonging to the system is known, thus leading to a decreasing order of priorities of the *otasks*. Second, the priority of each otask is not known, in this case we have first derived an *optimal* assignment of priorities to the otasks. This model has been used to handle the scheduling problem for hard real-time systems with precedence, strict periodicity and latency constraints. We have closely studied the impact of the scenario of first releases of all tasks on the schedulability analysis. We have proved on the one hand that the scenario where the first releases of all tasks are simultaneous does not correspond to the worst-case scenario, and on the other hand that there is not such a scenario when the cost of the preemption is considered. Then, we have proved that the assignment of priorities to tasks based on Audsley's policy is no longer applicable and nor optimal. Here optimal means that if there exits an assignment which leads to a valid schedule for the given set of tasks then the proposed assignment of priorities will also lead to a valid schedule. In order to take into account the global cost of the RTOS (Real-Time Operating System), for which the preemption cost is the variable part, we have simulated many sets of tasks with the *Linux/RTAI* RTOS in order to evaluate its impact on the schedulability analysis.

Currently, we are introducing jitter constraints to our model, and the preliminary results we got were about non-schedulability conditions.

6.6. Non-preemptive real-time scheduling in the multiprocessor case

Participants: Omar Kermia, Yves Sorel.

The last two years we have performed a schedulability study which led to a schedulability condition taking into account only two tasks at the same time. From this condition we proposed, first, a greedy heuristic for non-preemptive multiprocessor scheduling of systems with precedence and strict periodicity constraints. Secondly, in order to improve the first algorithm, we introduced a back-tracking procedure leading to a better schedulability ratio.

This year, we continued the schedulability study by proposing a more general schedulability condition which checks whether a task satisfies or not its periodicity and precedence constraints on a processor where some tasks have already been scheduled [19]. Since the problem of deciding that such systems are schedulable or not is NP-hard in the strong sense, this condition constitutes an important outcome. It can be used in partitioned scheduling approaches for solving the multiprocessor scheduling problem. In this approach tasks are partitioned among the processors which transforms the multiprocessor problem to several monoprocessor problems. We proposed a new heuristic using this condition. It has a much more effective schedulability ratio than the previous versions, and a satisfying execution time.

The novelty, this year, is the introduction of latency constraints in the schedulability study. The considered latency constraint corresponds to a delay between whatever pair of tasks in the system. Most of the time it is imposed between an input event consumed by the system and an output event produced by the system. As the scheduling problem under latency constraints is also an NP-hard problem, we chose to first study schedulabilty under latency and precedence constraints first, and then to deal with schedulability under all constraints. As a result we proposed a schedulability condition and a heuristic for multiprocessor scheduling of tasks under precedence, strict periodicity and latency constraints.

All previous scheduling heuristics aim at minimizing the makespan which is the total execution time of all the tasks, in addition to satisfy all the constraints. However, the makespan can be more minimized with a load balancing heuristic. Moreover, since memory is limited in embedded systems, we need a heuristic to perform an efficient memory usage. Thus, we proposed a heuristic for load balancing and efficient memory usage of homogeneous distributed real-time embedded systems [18]. Basically, it is achieved by grouping the tasks into blocks, and moving them to the processor such that the block start time decreases, and this processor has enough memory capacity to execute the tasks of the block. We have shown that the proposed heuristic has a polynomial complexity which ensures a fast execution time. We performed also a theoretical performance study which bounds the total execution time decreasing, and shows that our heuristic is a $(2 - \frac{1}{M})$-approximation algorithm for the memory usage, with M the number of processors.

6.7. Globally asynchronous implementations of synchronous specifications

Participants: Dumitru Potop-Butucaru, Robert de Simone, Yves Sorel.

We have continued in 2008 our research effort concerning the deterministic globally asynchronous implementation of synchronous specifications. Our approach is to consider multi-clock synchronous specifications, and then encode the absence and non-execution of the synchronous model with actual absence of communication and non-execution in the globally asynchronous implementation. This raises complex correctness issues that must be solved. The primary practical application is the extension of the class of implementations AAA/SynDEx can support through the definition of new synchronization schemes and associate implementation algorithms. Here, the objective is to allow operations and communication lines to be inactive in certain logical instants (repetitions of the graph pattern representing the algorithm) depending on the state and input data.

We focused on the semantics-preserving execution of a single synchronous program/core in an asynchronous environment. More precisely, we need to preserve the function of the synchronous program/core (as an I/O stream mapping) while allowing for elastic timing.

Our first contribution here has been the characterization of the largest class of synchronous programs/cores that produce deterministic implementations using a very general execution machine based on: (1) the chosen signal absence encoding, and (2) an ASAP (as soon as possible) reaction triggering policy. The characterization is a form of *confluence* and *determinism*. We also characterized the largest sub-class of such programs that is closed under synchronous composition (thus offering the basis for realistic techniques for incremental development). These results can be found in [30].

The second contribution has been the definition of algorithms allowing us to check whether a general synchronous program (specified in the Signal language developed by the Espresso Team-Project) is weakly endochronous. For a weakly endochronous program, the algorithms also produce the set of atomic behaviors, allowing the simple construction of the GALS wrapper. When the program is not weakly endochronous, the output of the algorithm allows the transformation of the specification into a weakly endochronous one through the addition of a minimal number of messages carrying signal absence information over the existing communication lines.

6.8. Correct and efficient real-time implementation of synchronous specifications

Participants: Dumitru Potop-Butucaru, Yves Sorel.

Recent evolutions in the classes of specifications and desired implementations of SynDEx resulted in a need for revisiting and improving the formal support of the AAA/SynDEx methodology. A preliminary analysis of the AAA/SynDEx has revealed that the high abstraction level of the hardware architecture models of SynDEx results in:

- Very large time overheads in the implementation, mainly due to over-synchronization in a distributed environment.
- Difficult construction of correct and efficient communication libraries and difficult synthesis of correct and efficient synchronization code.

It is therefore imperative to invest more in the definition of lower-level models of the implementation architecture, in order to ensure the correctness and efficiency of the generated code.

We started by considering the problem of optimizing existing static schedules of conditional communications over a broadcast asynchronous bus. Preliminary results in this direction are presented in [31].

6.9. Development of CyCab applications with SynDEx

Participants: Omar Kermia, Nik Khairail Afzan Nik Mohamed [intern], Bertrand Vié Ducasse [intern], Yves Sorel.

The previous years we developed with SynDEx a "visual control of autonomous CyCabs for platooning" application. This year we improved the image processing algorithm which detects the followed CyCab from a webcam in order to achieve longitudinal as well as lateral control. The latter improve the detection when the followed CyCab is turning. In addition, the CyCab detection algorithm was integrated in the control algorithm to obtain a unique algorithm specified with SynDEx whereas the detection algorithm did not appear in the control algorithm previously. Complex problems have arisen due to the mix of two types of real-time executives. Indeed, the detection algorithm is executed under Linux/RTAI and the control algorithm is executed under a real-time executive completely synthetized by SynDEx. Moreover, the webcam uses a Linux driver which must communicate the images to Linux/RTAI through a shared memory. Finally, a first version of the complete application specified and implemented with SynDEx on the distributed architecture of the CyCab (several MPC555 microcontrollers and an embedded PC communicating through a CAN bus) was obtained.

6.10. Improvements in SynDEx

Participants: Maxence Guesdon, Omar Kermia, Yves Sorel, Cécile Stentzel.

The previous years two main improvements have been achieved in SynDEx: a new data structure associated to a better GUI (User Graphical Interface), and a multi-periodic version of the optimized distribution and scheduling heuristic, called adequation in SynDEx, whereas it was mono-periodic before. However, these two developments were achieved separately, mainly because the new multi-periodic heuristic was developed by the PhD student Omar Kermia whereas the new structure and GUI was developed by a software engineer. Consequently, this year was mainly devoted to merge both software developments in a consistent way, to test the new GUI and the new adequation, and to update the documentation in consequence. This important work should lead to a new major release of SynDEx, called V7, at the end of the year or the beginning of the next one.

In addition, for the code generator of SynDEx, we studied the possible improvements in terms of code length and synchronization optimizations.

7. Contracts and Grants with Industry

7.1. Texas Instruments

Participants: Julien Boucaron, Anthony Coadou, Robert de Simone.

This collaboration takes the form of a series of one-year grants (4 so far). We explore Latency-Insensitive Design (LID), based on the original work of Luca Carloni (now at Columbia University), with whom we share an associated-team programme supported by INRIA.

This year we worked on the extension of our modeling framework to encompass alternative routing and signal redirection. In the case of ultimately k-periodic schedules of LID systems, we also consider k-periodic switching schemes. The results are utterly interesting in that we prove that algebraic transformation holds, by which we can either share less channels with more interleaving and demanding scheduling, or progress communications faster with more channeling resources, where easier schedules may be feasible. The combination of scheduling time periodicity and routing space periodicity allows us to compute predictable buffering needs to accomodate the throughput. One can then hope in the future to devise techniques for re-allocating and re-routing, similar to nowadays retiming/recycling approach.

We also considered the relevance of these models in the light of existing tools for high-level synthesis currently introduced at Texas Instruments, mainly Synfora's PICO EXPRESS and Esterel-EDA's ESTEREL STUDIO.

7.2. ID/TL-M project with ST MIcroelectronics

Participants: Charles André, Julien Boucaron, Robert de Simone, Benoît Ferrero.

This contract started at the end of the year, in the context of the Nano2012 programme and the overall INRIA-ST partnership agreement. The Kick-off meeting was held on December 5th. The DaRT EPI from Lille is also taking part in this collaboration.

The main topic of ID/TL-M is to study the introduction of model-driven engineering techniques (MDE) at the transaction-level modeling (TLM) level of SoC design. Bridges between the OMG profile MARTE and the dedicated standard IP-XACT for Electronic System-Level (ESL) design shall be establised and realized. Formal Models of Computation and Communication (MoCCs) wil be represented, and model transformations across levels shall also be established and realized. The PAPYRUS UML modeler by CEA, and ECLIPSE environments shall be considered to support these implementations.

7.3. Thales TRT

Participants: Patrick Meumeu, Thomas Ponchant, Yves Sorel.

We started a collaboration with the Thales Research and Technology group which designs the new distributed architectures for embedded systems that are used in the different business units of Thales. They are interested in multi-core archictecture dedicated to critical embedded systems. A first study focused on predictable RTOS that will be used in each core.

8. Other Grants and Activities

8.1. Regional collaborations

8.1.1. CIM PACA

Participants: Jean-François Le Tallec, Jean-Vivien Millo, Robert de Simone.

This ambitious regional initiative is intended to foster collaborations between local PACA industry and academia partners on the topics of microelectronic design, though mutualization of equipments, resources and R&D concerns. We are actively participating in the Design Platform (one of the three platforms launched in this context). Other participants are UNS, CNRS (I3S and LEAT laboratories), L2MP Marseille, CMP-ENSE Gardanne on the academic side, and Texas Instruments, Philips, ST Microelectronics, ATMEL, and Esterel-EDA on the industrial side.

Inside this platform we are coordinating a dedicated project, named Spec2RTL, on methodological flows for high-level SoC synthesis. Participants are Texas Instruments, NXP, ST Microelectronics, Synopsys, Esterel-EDA, and Scaleo Chip as industrial partners, INRIA, I3S (CNRS/UNSA) and ENST on the academic side. A pool of PhD students are funded on a par basis between industrial partners and local PACA PhD grants under the BDI programme. There are currently 7 such students, two of them hosted by the Aoste team.

Jean-Vivien Millo, supported in part by ST Microelectronic, defended his PhD thesis on December 17th. His main research topic was static balanced scheduling of data traffic in the LID design of GALS systems, with applications on modular SoC design.

Jean-François Le Tallec started his PhD thesis in connection with Scaleo Chip, a local SME company developing SoC platform simulators. The PhD topic is to investigate new *virtual* platform environments at ESL TLM level, and their relation to formal modeling in multiclock ESTEREL.

8.2. Nation-wide collaborations

8.2.1. RNTL platform OpenEmbeDD

Participants: Charles André, Benoît Ferrero, Fadoi Lakhal, Robert de Simone, Yves Sorel.

OpenEmbeDD is a large platform project aimed at connecting several formalisms with model-driven engineering tools, in the embedded domain. The project partners are: INRIA, CEA-List, Thales, Airbus, France Telecom, CS, LAAS, and VERIMAG. Four INRIA teams are involved (ATLAS, Triskell, Aoste and DaRT).

The focus is on the use of model-driven approaches to combine various specification formalisms, analysis and modeling techniques, into an interoperable framework. We contribute to this in several directions:

- first, we provide the definition and implementation of the Marte Time subprofile, as described in 3.2, with the support of our TIMESQUARE tool for resolution and animation of Clock Constraints;
- second, we contribute our work on compilation-by-transformation of asynchronous to multiclock to plain synchronous programs;
- third, we developed an effective coupling of SYNDEX to MARTE models. The editor of SynDEx models based on TopCased under Eclipse with the UML/MARTE profile was improved in order to represent application algorithms with conditioning and hierarchical descriptions. On the other hand we also improved the translator from XMI to the .sdx format for these new features.

The various partner contributions in this project are assembled together by a dedicated engineer team of two people located at IRISA, as part of an INRIA forge.

8.2.2. FUI Lambda

Participants: Charles André, Julien DeAntoni, Robert de Simone, Frédéric Mallet.

The Lambda project is headed by Thales, with ST Microelectronics, Airbus, Esterel-EDA, CoFluent, CEA-LIst, and several other partners.

Our contribution is initially rather light. We bring expertise to help with the definition of a model transformation between SYNCCHARTS and UML State Diagrams. We then contribute to the combination of SYSML and MARTE paradigms, in the context of SoC design as well as the SPIRIT IP-XACT standard.

8.2.3. RNTL project MeMVaTEx

Participants: Djamel Louar, Fadoi Lakhal, Marie-Agnès Peraldi-Frati, Dumitru Potop-Butucaru, Yves Sorel.

The partners of the MeMVaTEx project are: Continental, INRIA, CEA-List, CNRS-UTC, and Sherpa Engineering. The project focuses on developing a design methodology, centred on the requirements, their traceability and their validation. The application domain is the design of complex real-time automotive systems. The methodology is based on the standards EAST-ADL2, AUTOSAR, SYSML and MARTE. The project is currently centred on the heterogeneous phase. This phase integrates the Simulink and SynDEx tools in order to provide the validation of requirements and models.

During this year, we worked on the solution http://www.artist-embedded.org/artist/model and we integrated in the methodology the joint modeling of the hardware architecture models and the timing modeling at the different levels of the development cycle (i.e. Design Level and Autosar Level). We established the link between the solution functional model and the requirement model for these particular model elements (hardware and timing).

We worked also on the MeMVaTEx demonstrator for illustrating the previous results by developing the different models and by implementing in the Artisan software tool the Autosar profile and the time package of the MARTE profile.

We reported all these results in the deliverables JSP1T2-I-b, JSP1T2-II-a, JSP1T5-II-a, JSP1T6-I-b of the project.

8.2.4. ANR RT-Simex

Participant: Frédéric Mallet.

This new project is dedicated to the reverse engineering of analysis traces of simulation and execution back up to the source code, or in our case most likely into the original models in a MARTE profile representation. The prime contractor is the OBEO company. A kick-off meeting is due soon.

8.3. European collaborations

8.3.1. IST Network of Excellence ARTIST2 & ARTIST-Design

Participants: Julien Boucaron, Robert de Simone, Frédéric Mallet.

We conducted a comparative work in this context on Process Networks and MoCCs, submitted for publication.

Frédéric Mallet attended the UML&FM workshop held in Japan with the strong support of ARTIST.

9. Dissemination

9.1. Leadership within scientific community

- Robert de Simone was programme committee member for MemoCode'08. He was on the Selection Board of experts for the ANR programme ARPEGE 2008. He was a member of the *Commission de Spécialiste 27^{e} section* from the University of Nice-Sophia Antipolis, and INRIA representative to the UNS Doctoral School. He represents INRIA in the CA board of ARCSIS, the ruling association for the CIM PACA, as well being a member of its Strategic Council. He holds the same positions in the Design Platform branch of this organization. He was reviewer for the PhD theses of Ludovic Samper and Jerôme Cornet (Verimag).
- Charles André was reviewer of the PhD theses of Sébastien Revol (INPG), Huafeng Yu (LIFL), Cécile Hardebolle (LRI).
- Yves Sorel is programme committee member for the following conferences: DASIP, EUSIPCO, GRESTSI. He is member of the OCDS/SYSTEM@TIC Paris-Region Cluster Steering Committee. He participated to the Habilitation Thesis jury of Laurent George.
- Marie-Agnès Peraldi-Frati is a member of the UNS *Commission de Spécialiste 27^{e} section*. She is member of the CNRS/I3S conseil de laboratoire and member of the CERTEC (conseil d'études et de la recherche technologique) of the IUT of Nice-Sophia Antipolis.

9.2. Teaching

- Charles André is a professor at the university of Nice-Sophia Antipolis, department of Electrical Engineering. He taught Computer Architecture, Hardware Description Languages and Real-time programming (1 term).
- Julien DeAntoni gives courses at different cursus levels of the polytechnic school of the UNS: A course and labs on micro-controller and Real-Time operating system programming in the last year of the engineering cursus. He also teaches object oriented programmation through C++ in the 4th year of the mathematical and modeling cursus as well as linux shell programming in the second year of the engineering cursus.
- Robert de Simone taught courses on Formal Methods and Models for Embedded Systems in the STIC Research Master program of the university of Nice-Sophia Antipolis (UNS), for approximately 15h.

- Marie-Agnès Peraldi-Frati gives courses at different cursus levels of UNS: A course and labs on UML for real-time in the TSM master (Telecommunication, System and Microelectronics) at the University of Nice. She gives different courses (Systems and networks, Programming, Web development, Computer architecture) at the L1 level of the IUT Informatique.
- Yves Sorel gives courses in last year of the ESIEE (Engineering School located in Noisy-le-Grand), in the SETI Research Master at the University of Orsay Paris 11, and in last year of the ENSTA (Engineering School located in Paris), on topics comprising the AAA methodology and the optimization of distributed real-time embedded systems.

10. Bibliography

Major publications by the team in recent years

[1] C. André, F. Mallet, R. de Simone. *Modeling Time(s)*, in "MoDELS'2007 10th Intern. Conf. on Model Driven Engineering Languages and Systems", 2007.

[2] A. Benveniste, P. Caspi, S. Edwards, N. Halbwachs, P. Le Guernic, R. de Simone. *Synchronous Languages Twelve Years Later*, in "Proceedings of the IEEE", January 2003.

[3] L. Cucu, N. Pernet, Y. Sorel. *Periodic real-time scheduling: from deadline-based model to latency-based model*, in "Annals of Operations Research", 2007, http://www-rocq.inria.fr/syndex/pub/aor07/aor07.pdf.

[4] T. Grandpierre, Y. Sorel. *From Algorithm and Architecture Specification to Automatic Generation of Distributed Real-Time Executives: a Seamless Flow of Graphs Transformations*, in "Proceedings of First ACM and IEEE International Conference on Formal Methods and Models for Codesign, MEMOCODE'03", June 2003.

[5] P. Meumeu Yomsi, Y. Sorel. *Non-Schedulability Conditions for Off-line Scheduling of Real-Time Systems Subject to Precedence and Strict Periodicity Constraints*, in "Proceedings of 11th IEEE International Conference on Emerging technologies and Factory Automation, ETFA'06, WIP, Prague, Czech Republic", September 2006, http://www-rocq.inria.fr/syndex/pub/etfa06/etfa06.pdf.

[6] D. Potop-Butucaru, S. Edwards, G. Berry. *Compiling Esterel*, Springer, 2007.

[7] D. Potop-Butucaru, R. de Simone. *Optimizations for Faster Execution of Esterel Programs*, in "MEMOCODE'03", 2003.

[8] E. Vecchié, R. de Simone. *Syntax-driven optimizations for Reachable State Space construction of Esterel programs*, in "International Journal of Embedded Systems", April 2005.

[9] R. de Simone, O. Tardieu. *Loops in Esterel*, in "ACM Transactions on Embedded Computing Systems", 2005.

[10] R. de Simone, D. Potop-Butucaru, Jean-Pierre. Talpin. *The Synchronous Hypothesis and Synchronous Languages*, in "Embedded Systems Handbook", chapter of the Embedded Systems Handbook, CRC Press, CRC Press, 2005.

Year Publications

Doctoral Dissertations and Habilitation Theses

[11] F. LAGARDE. *Contribution à la conception de langages de modélisation spécifiques fondés sur UML*, Ph. D. Thesis, Université de Nice-Sophia Antipolis, Novembre 2008.

[12] JEAN-VIVIEN. MILLO. *Ordonnancements périodiques dans les réseaux de processus : Application à la conception insensible aux latences*, Ph. D. Thesis, Université de Nice-Sophia Antipolis, Décembre 2008.

Articles in International Peer-Reviewed Journal

[13] F. MALLET, C. ANDRÉ, R. DE SIMONE. *CCSL: specifying clock constraints with UML/Marte*, in "Innovations in Systems and Software Engineering (ISSE)-Special Issue on UML & Formal Methods", vol. 4, n^{o} 3, 2008, p. 309–314.

Invited Conferences

[14] F. MALLET, R. DE SIMONE. *Marte: A profile for RT/E Systems Modeling, Analysis (and Simulation?)*, in "Int. Conf. on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools)", March 2008.

International Peer-Reviewed Conference/Proceedings

[15] A. ALBINET, H. DUBOIS, MARIE-AGNÈS. PERALDI-FRATI, Y. SOREL. *The MeMVaTEx Methodology: "From Requirements to Models in Automotive Application Design*, in "4th Int. Conf. on Embedded Real Time Software (ERTS 2008)", January 2008.

[16] C. ANDRÉ, F. MALLET, A. MEHMOOD KHAN, R. DE SIMONE. *Modeling Spirit IP-Xact in UML Marte*, in "IC on Design, Automation and Test in Europe (DATE), Marte Workshop", IEEE, March 2008, p. 35–40.

[17] S. DEMATHIEU, F. THOMAS, C. ANDRÉ, S. GÉRARD, F. TERRIER. *First Experiments Using the UML Profile for Marte*, in "11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2008)", IEEE Computer Society, 2008, p. 50-57.

[18] O. KERMIA, Y. SOREL. *Load Balancing and Efficient Memory Usage for Homogeneous Distributed Real-Time Embedded Systems*, in "Proceedings of the 4th International Workshop on Scheduling and Resource Management for Parallel and Distributed Systems, SRMPDS'08, Portland, Oregon, USA", September 2008, http://www-rocq.inria.fr/syndex/pub/srmpds08/srmpds08.pdf.

[19] O. KERMIA, Y. SOREL. *Schedulability Analysis for Non-Preemptive Tasks under Strict Periodicity Constraints*, in "Proceedings of 14th International Conference on Real-Time Computing Systems and Applications, RTCSA'08, Kaohsiung, Taiwan", August 2008, http://www-rocq.inria.fr/syndex/pub/rtcsa08/rtcsa08.pdf.

[20] F. LAGARDE, H. ESPINOZA, F. TERRIER, C. ANDRÉ, S. GÉRARD. *Leveraging Patterns on Domain Models to Improve UML Profile Definition*, in "Fundamental Approaches to Software Engineering (FASE 2008)", 2008, p. 116-130.

[21] SU-YOUNG. LEE, F. MALLET, R. DE SIMONE. *Dealing with AADL End-to-end Flow Latency with UML Marte*, in "13th International Conference on Engineering of Complex Computer Systems (ICECCS 2008)", IEEE Computer Society, 2008, p. 228–233.

[22] F. MALLET, R. DE SIMONE, L. RIOUX. *Event-triggered vs. time-triggered communications with UML Marte*, in "Specification, Verification and Design Languages (FDL 2008)", September 2008, p. 154–159.

[23] MARIE-AGNÈS. PERALDI-FRATI, Y. SOREL. *From High-Level Modelling of Time in Marte to Real-Time Scheduling Analysis*, in "MoDELS'08 Workshop on Model Based Architecting and Construction of Embedded Systems on ACES-MB", September 2008, p. 129–143.

[24] G. PERROUIN, F. CHAUVEL, J. DEANTONI, JEAN-MARC. JÉZÉQUEL. *Modeling the Variability Space of Self-Adaptive Applications*, in "2nd Dynamic Software Product Lines Workshop (SPLC 2008, Volume 2), Limerick, Ireland", S. THIEL, K. POHL (editors), IEEE Computer Society, September 2008, p. 15–22.

Scientific Books (or Scientific Book chapters)

[25] C. ANDRÉ, F. MALLET, R. DE SIMONE. *Modeling of AADL data-communications with UML Marte*, in "Embedded Systems Specification and Design Languages", LNEE, vol. 10, Springer, May 2008, p. 150–170.

Research Reports

[26] C. ANDRÉ, F. MALLET. *Clock Constraints in UML Marte CCSL*, Technical report, n^{o} 6540, INRIA, May 2008.

[27] J. BOUCARON, A. COADOU, B. FERRERO, JEAN-VIVIEN. MILLO, R. DE SIMONE. *Kahn-extended Event Graphs*, Technical report, n^{o} 6541, INRIA, May 2008.

[28] F. LAGARDE, F. MALLET, C. ANDRÉ, S. GÉRARD, F. TERRIER. *An Automated Process for Designing UML Profiles*, Technical report, n^{o} 6599, INRIA, July 2008.

[29] F. MALLET, C. ANDRÉ. *UML/Marte CCSL, Signal and Petri nets*, Technical report, n^{o} 6545, INRIA, May 2008.

[30] D. POTOP-BUTUCARU, R. DE SIMONE, Y. SOREL. *Deterministic execution of synchronous programs in an asynchronous environment*, Research Report, n^{o} RR-6656, INRIA, 2008.

[31] D. POTOP-BUTUCARU, R. DE SIMONE, Y. SOREL. *Optimized static real-time scheduling of communications on a broadcast bus*, Research Report, n^{o} RR-6655, INRIA, 2008.

Scientific Popularization

[32] J. DEANTONI, JEAN-PHILIPPE. BABAU. *SAIA : une utilisation conjointe du génie logiciel et des méthodes formelles*, Journal du département IF de l'INSA de Lyon, 2009.

[33] S. DEMATHIEU, S. GÉRARD, F. MALLET. *Marte, le nouveau standard UML pour les systèmes temps réel embarqués*, Electronique, March 2008.

Other Publications

[34] C. ANDRÉ, B. FERRERO, F. MALLET. *TimeSquare: a Multiform Time Simulation Environment*, available at http://www-sop.inria.fr/oasis/SAFA/abstracts/safa-dec08-CAndre.pdf, December 2008.

[35] J. DEANTONI. *Developing Real-Time Embedded Applications Independently of an Execution Platform*, available at http://www-sop.inria.fr/oasis/SAFA/abstracts/safa-dec08-JDeAntoni.pdf, December 2008.

[36] F. MALLET, C. ANDRÉ. *On the semantics of UML/MARTE Clock Constraints*, 12th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2009), 2009.

[37] F. MALLET, MARIE-AGNÈS. PERALDI-FRATI, C. ANDRÉ. .

References in notes

[38] C. ANDRÉ. *Representation and Analysis of Reactive Behavior: a Synchronous Approach*, in "Computational Engineering in Systems Applications (CESA)", IEEE-SMC, 1996, p. 19–29.

[39] F. BACCELLI, G. COHEN, GEERT JAN. OLSDER, JEAN-PIERRE. QUADRAT. *Synchronization and Linearity: an algebra for discrete event systems*, John Wiley & Sons, 1992, http://cermics.enpc.fr/~cohen-g/SED/book-online.html.

[40] A. BENVENISTE, G. BERRY. *The Synchronous Approach to Reactive and Real-Time Systems*, in "Proceedings of the IEEE", vol. 79, n^{o} 9, September 1991, p. 1270-1282.

[41] F. BOUSSINOT, R. DE SIMONE. *The Esterel Language*, in "Proceedings of the IEEE", September 1991.

[42] J. CARLIER, P. CHRÉTIENNE. *Problèmes d'ordonnancement*, Masson, 1988.

[43] L. CARLONI, K. MCMILLAN, A. SANGIOVANNI-VINCENTELLI. *Theory of Latency-Insensitive Design*, in "IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems", 2001.

[44] J.B. DENNIS. *First Version of a Dataflow Procedure Language*, in "Lecture Notes in Computer Sci.", vol. 19, Springer-Verlag, 1975, p. 362-376.

[45] S. EDWARDS. *Languages for Digital Embedded Systems*, Kluwer, 2000.

[46] T. GRANDPIERRE, C. LAVARENNE, Y. SOREL. *Optimized Rapid Prototyping For Real-Time Embedded Heterogeneous multiprocessors*, in "Proceedings of 7th International Workshop on Hardware/Software Co-Design, CODES'99", 1999.

[47] N. HALBWACHS. *Synchronous Programming of Reactive Systems*, in "Computer Aided Verification", 1998, p. 1-16, http://citeseer.ist.psu.edu/10686.html.

[48] H. HEINECKE. *AUTOSAR, an industrywide initiative to manage the complexity of emerging Automotive E/E-Architecture*, in "Electronic Systems for Vehicles 2003, VDI Congress, Baden-Baden", 2003.

[49] EDWARD A. LEE, D. G. MESSERSCHMITT. *Static Scheduling of Synchronous Data Flow Programs for Digital Signal Processing*, 1987.

[50] C.L. LIU, J.W. LAYLAND. *Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment*, in "Journal of the ACM", 1973.

[51] Y. SOREL. *Massively Parallel Systems with Real Time Constraints, the Algorithm Architecture Adequation Methodology*, in "Proceedings of Conference on Massively Parallel Computing Systems, MPCS'94, Ischia, Italy", May 1994.