IN PARTNERSHIP WITH:
CNRS

Université Nice - Sophia Antipolis

Activity Report 2012

Project-Team AOSTE

Models and methods of analysis and optimization for systems with real-time and embedding constraints

IN COLLABORATION WITH: Laboratoire informatique, signaux systèmes de Sophia Antipolis (I3S)

RESEARCH CENTERS
Sophia Antipolis - Méditerranée
Paris - Rocquencourt

THEME
Embedded and Real Time Systems

Table of contents

Project-Team AOSTE

Keywords: Embedded Systems, Synchronous Languages, Model-Driven Engineering, Real-Time, Scheduling, Concurrency, Model Of Computation

Aoste is a joint team with the University of Nice/Sophia-Antipolis (UNS) and the UMR CNRS I3S. It is also co-located between the two Inria centers of Sophia-Antipolis and Rocquencourt.

Creation of the Project-Team: July 01, 2004 .

1. Members

Research Scientists

Robert de Simone [Team Leader, Senior Researcher, Inria, Sophia Antipolis - Méditerranée, HdR]
Yves Sorel [Team Vice-leader, Senior Researcher, Inria, Paris - Rocquencourt]
Dumitru Potop Butucaru [Junior Researcher,Inria, Paris - Rocquencourt]
Laurent George [Associate Professor, UPEC, on leave at Inria until August 2012, now External Collaborator, Paris - Rocquencourt, HdR]

Faculty Members

Charles André [Professor, UNS, emeritus, Sophia Antipolis - Méditerranée, HdR]
Sid-Ahmed-Ali Touati [Professor, UNS, Sophia Antipolis - Méditerranée, HdR]
Frédéric Mallet [Associate Professor, UNS, Sophia Antipolis - Méditerranée, HdR]
Marie-Agnès Peraldi Frati [Associate Professor, UNS, Sophia Antipolis - Méditerranée]
Julien Deantoni [Associate Professor, UNS, Sophia Antipolis - Méditerranée]

Engineers

Nicolas Chleq [Inria support engineer, Sophia Antipolis - Méditerranée]
Jean-François Le Tallec [Temporary Engineer, until June 2012, Sophia Antipolis - Méditerranée]
Simon Nivault [Temporary Engineer, Paris - Rocquencourt]
Meriem Zidouni [Temporary Engineer, Paris - Rocquencourt]
Abderraouf Benyahia [Temporary Engineer, from May 2012, Paris - Rocquencourt]
Daniel de Rauglaudre [Inria support engineer, Paris - Rocquencourt]
Maxence Guesdon [Inria support Engineer, until March 2011, Paris - Rocquencourt]
Cécile Stentzel [Inria support Engineer, from April 2011, Paris - Rocquencourt]

PhD Students

Manel Djemal [CORDI/S scholarship, Paris - Rocquencourt]
Thomas Carle [Inria scholarship, since September 2011, Paris - Rocquencourt]
Falou Ndoye [Inria scholarship, Paris - Rocquencourt]
Carlos Gomez Cardenas [MENESR scholarship, Sophia Antipolis - Méditerranée]
Matias Vara Larsen [funded by ANR GeMoC, from December 2012, Sophia Antipolis - Méditerranée]
Ameni Khecharem [funded by ANR HOPE, Sophia Antipolis - Méditerranée]
Mohamed Marouf [until June 2012, joint with EPI IMARA, Paris - Rocquencourt]
Ling Yin [ECNU Shanghai, until September 2012, Sophia Antipolis - Méditerranée]

Post-Doctoral Fellows

Jean-Vivien Millo [funded by ANR HeLP, Sophia Antipolis - Méditerranée]
Arda Goknil [funded by ARTEMIS PRESTO, Sophia Antipolis - Méditerranée]
Virginia Papailiopoulou [until May 2012, Paris - Rocquencourt]
Zhen Zhang [from July 2012, Paris - Rocquencourt]

Administrative Assistants

Patricia Lachaume [TCN Inria, part-time (50%), Sophia Antipolis - Méditerranée]
Christine Anocq [TRS Inria, part-time (30%), Paris - Rocquencourt]

Magali Richir [AI CNRS, part-time (5%), Sophia Antipolis - Méditerranée]

2. Overall Objectives

2.1. Embedded System Design

Typical embedded software *applications* display a mix of multimedia signal/data processing with modal interfaces, resulting in heterogenous concurrent data-flow streaming models, and often stringent real-time constraints. Similarly, embedded *architectural platforms* are becoming increasingly parallel, with dedicated hardware accelators and manycore processors. The optimized compilation of such kinds of applications onto such execution platforms involves complex mappping issues, both in terms of spatial distribution and in terms of temporal scheduling. Currently, it is far from being a fully automatic compilation process as in the case of commodity PC applications. Models are thus needed, both as formal mathematical objects from theoretical computer science to provide foundations for embedded system design, and also as engineering models to support an effective design flow.

Our general approach is directly inspired from the theories of synchronous languages, process networks, and of real-time distributed scheduling. We insist on the introduction of *logical time* as functional design ingredient to be explicitly considered as first-class modeling element of systems. Logical time is based on logical clocks, where such a clock can be defined as any meaningful sequence of event occurrences, usually meant as activation/triggering conditions for actions and operations in the systems. So logical time can be multiform, a global partial order built from local total orders of clocks. In the course of the design flow *time refinement* takes place, as decison are made towards placement and timing of various tasks and operations. This solves in part the constraints between clocks, committing to schedule and placement decisions. The final version should be totally ordered, and then subjet to physical timing verification as to physical constraints.

The general (logical) *Time Model* has been standardized as part of the OMG profile for Modeling and Analysis of Real-Time Embedded systems (MARTE).

Work on polychronous formalisms (descending from ESTEREL), on a Clock Constraint Specification Language handling logical time, on Application-Architecture Adequation approach and real-time scheduling results has been progressed over the years, resulting in sofware environments such as SYNDEX or TimeSquare.

2.2. Highlights of the Year

Aoste underwent its periodical Inria evaluation, as part of the Real-Time Embededded theme, in its eighth year of existence. Evaluation was very positive.

3. Scientific Foundations

3.1. Models of Computation and Communication (MoCCs)

Participants: Charles André, Robert de Simone, Jean-Vivien Millo, Dumitru Potop Butucaru.

Esterel, SyncCharts, synchronous formalisms, Process Networks, Marked Graphs, Kahn networks, compilation, synthesis, formal verification, optimization, allocation, refinement, scheduling

Formal Models of Computation form the basis of our approach to Embedded System Design. Because of the growing importance of communication handling, it is now associated with the name, MoCC in short. The appeal of MoCCs comes from the fact that they combine features of mathematical models (formal analysis, transformation, and verification) with this of executable specifications (close to code level, simulation, and implementation). Examples of MoCCs in our case are mainly synchronous reactive formalisms and dataflow process networks. Various extensions or specific restrictions enforce respectively greater expressivity or more focused decidable analysis results.

DataFlow Process Networks and Synchronous Reactive Languages such as ESTEREL/SYNCCHARTS and SIGNAL/POLYCHRONY [53], [54], [48], [15], [4], [13] share one main characteristics: they are specified in a self-timed or loosely timed fashion, in the asynchronous data-flow style. But formal criteria in their semantics ensure that, under good correctness conditions, a sound synchronous interpretation can be provided, in which all treatments (computations, signaling communications) are precisely temporally mapped. This is refered to as clock calculus in synchronous reactive systems, and leads to a large body of theoretical studies and deep results in the case of DataFlow Process Networks [49], [47] (consider SDF balance equations for instance [56]).

As a result, explicit schedules become an important ingredient of design, which ultimately can be considered and handled by the designer him/herself. In practice such schedules are sought to optimize other parts of the design, mainly buffering queues: production and consumption of data can be regulated in their relative speeds. This was specially taken into account in the recent theories of Latency-Insensitive Design [50], or N-synchronous processes [51], with some of our contributions [6].

Explicit schedule patterns should be pictured in the framework of low-power distributed mapping of embedded applications onto manycore architectures, where they could play an important role as theoretical formal models on which to compute and optimize allocations and performances. We describe below two lines of research in this direction. Striking in these techniques is the fact that they include time and timing as integral parts of early functional design. But this original time is logical, multiform, and only partially ordering the various functional computations and communications. This approach was radically generalized in our team to a methodology for logical time based design, described next (see 3.2).

3.1.1. *K-periodic static scheduling and routing in Process Networks*

In the recent years we focused on the algorithm treatments of ultimately k-periodic schedule regimes, which are the class of schedules obtained by many of the theories described above. An important breakthrough occurred when realizing that the type of ultimatelly periodic binary words that were used for reporting *static scheduling* results could also be employed to record a completely distinct notion of ultimately k-periodic route switching patterns, and furthermore that commonalities of representation could ease combine them together. A new model, by the name of K-periodical Routed marked Graphs (KRG) was introduced, and extensively studied for algebraic and algorithmic properties [5].

The computations of optimized static schedules and other optimal buffering configurations in the context of latency-insensitive design led to the K-Passa software tool development 5.2.

3.1.2. *Endochrony and GALS implementation of conflict-free polychronous programs*

The possibility of exploring various schedulings for a given application comes from the fact that some behaviors are truly concurrent, and mutually *conflict-free* (so they can be executed independently, with any choice of ordering). Discovering potential asynchronous inside synchronous reactive specifications then becomes something highly desirable. It can benefit to potential distributed implementation, where signal communications are restricted to a minimum, as they usually incur loss in performance and higher power consumption. This general line of research has come to be known as Endochrony, with some of our contributions [11].

3.2. Logical Time in Model-Driven Embedded System Design

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

Starting from specific needs and opportunities for formal design of embedded systems as learned from our work on MoCCs (see 3.1), we developed a Logical Time Model as part of the official OMG UML profile MARTE for Modeling and Analysis of Real-Time Embedded systems. With this model is associated a Clock Constraint Specification Language (CCSL), which allows to provide loose or strict logical time constraints between design ingredients, be them computations, communications, or any kind of events whose repetitions can be conceived as generating a logical conceptual clock (or activation condition). The definition of CCSL is provided in [1].

Our vision is that many (if not all) of the timing constraints generally expressed as physical prescriptions in real-time embedded design (such as periodicity, sporadicity) could be expressed in a logical setting, while actually many physical timing values are still unknown or unspecified at this stage. On the other hand, our logical view may express much more, such as loosely stated timing relations based on partial orderings or partial constraints.

So far we have used CCSL to express important phenonema as present in several formalisms: AADL (used in avionics domain), EAST-ADL2 (proposed for the AutoSar automotive electronic design approach), IP-Xact (for System-on-Chip (*SoC*) design). The difference here comes from the fact that these formalisms were formerly describing such issues in informal terms, while CCSL provides a dedicated formal mathematical notation. Close connections with synchronous and polychronous languages, especially Signal, were also established; so was the ability of CCSL to model dataflow process network static scheduling.

In principle the MARTE profile and its Logical Time Model can be used with any UML editor supporting profiles. In practice we focused on the PAPYRUS open-source editor, mainly from CEA LIST. We developed under Eclipse the TIMESQUARE solver and emulator for CCSL constraints (see 5.1), with its own graphical interface, as a stand-alone software module, while strongly coupled with MARTE and Papyrus.

While CCSL constraints may be introduced as part of the intended functionality, some may also be extracted from requirements imposed either from real-time user demands, or from the resource limitations and features from the intended execution platform. Sophisticated detailed descriptions of platform architectures are allowed using MARTE, as well as formal allocations of application operations (computations and communications) onto platform resources (processors and interconnects). This is of course of great value at a time where embedded architectures are becoming more and more heterogeneous and parallel or distributed, so that application mapping in terms of spatial allocation and temporal scheduling becomes harder and harder. This approach is extensively supported by the MARTE profile and its various models. As such it originates from the Application-Architecture-Adequation (AAA) methodology, first proposed by Yves Sorel, member of Aoste. AAA aims at specific distributed real-time algorithmic methods, described next in 3.3.

Of course, while logical time in design is promoted here, and our works show how many current notions used in real-time and embedded systems synthesis can naturally be phrased in this model, there will be in the end a phase of validation of the logical time assumptions (as is the case in synchronous circuits and SoC design with timing closure issues). This validation is usually conducted from Worst-Case Execution Time (WCET) analysis on individual components, which are then used in further analysis techniques to establish the validity of logical time assumptions (as partial constraints) asserted during the design.

3.3. The AAA (Algorithm-Architecture Adequation) methodology and Real-Time Scheduling

Participants: Laurent George, Dumitru Potop Butucaru, Yves Sorel.

Note: The AAA methodology and the SynDEx environment are fully described at http://www.syndex.org/, together with relevant publications.

3.3.1. Algorithm-Architecture Adequation

The AAA methodology relies on distributed real-time scheduling and relevant optimization to connect an Algorithm/Application model to an Architectural one. We now describe its premises and benefits.

The Algorithm model is an extension of the well known data-flow model from Dennis [52]. 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 define 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, whose vertices are of two types: “processor” (one sequencer of operations and possibly several sequencers of communications) and “medium” (support of communications), and whose edges are directed connections.

The resulting implementation model [9] is obtained by an external compositional law, for which the architecture graph operates on the algorithm graph. Thus, that result is a set of algorithm graphs, "architecture-aware", corresponding to refinements of the initial algorithm graph, by computing spatial (distribution) and timing (scheduling) allocations of the operations according to the architecture graph resource availability. In that context "Adequation" refers to some search amongst the solution space of resulting algorithm graphs, labelled by timing characteristics, for one which verifies timing constraints and optimizes some criteria, usually the total execution time and the number of computing resources (but other criteria may exist). The next section describes distributed real-time schedulability analysis and optimization techniques for that purpose.

3.3.2. Distributed Real-Time Scheduling and Optimization

We address 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. 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 [57], 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 [10]. 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 leads to solving 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-optimal for realistic problems corresponding to industrial needs. Long time ago we proposed a very fast “greedy” heuristics [8] whose results were regularly improved, and extended with local neighborhood heuristics, or used as initial solutions for metaheuristics such as variants of “simulated annealing”.

In addition to the spatial dimension (distributed) of the real-time scheduling problem, other important dimensions are the type of communication mechanisms (shared memory vs. message passing), or the source of control and synchronization (event-driven vs. time-triggered). We explore real-time scheduling on architectures corresponding to all combinations of the above dimensions. This is of particular impact in application domains such as automotive and avionics (see 4.2).

Since real-time distributed systems are often safety-critical we address dependability issues, to tolerate faults in processors and communication interconnects. We maily focus on software redondancy, rather than hardware, to ensure real-time behaviour preservation in presence of faulty processors and/or communication media (where possible failures are predictively specified by the designer). We investigate fail silent, transient, intermittent, and Byzantine faults.

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 specification of 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 design 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 compilations/syntheses 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 in general well accepted in the transportation domains, where design of digital software and electronic parts in usually tighly coupled with larger apsects of system design, where models from physics are being used already. The formalisms AADL (for avionics) and AutoSar [55] (for automotive) are providing support for this, unfortunately not always with a clean and formal semantics. Thus there is a strong need here for approaches that bring closer together formal methods and tools on the one hand, engineering best practices on the other hand.

From a structural point of view AUTOSAR succeeded in establishing a framework that provides significant confidence in the proper integration of software components from a variety of distinct suppliers. But beyond those structural (interface) aspects, dynamic and temporal views are becoming more of a concern, so that AUTOSAR has introduced the AUTOSAR Specification of Timing Extension. AUTOSAR (discrete) timing models consist of timing descriptions, expressed by events and event chains, and timing constraints that are imposed on these events and event chains.

An important issue in all such formalisms is to mix in a single design framework heterogeneous time models and tasks: based on different timebases, with different triggering policy (event-triggered and time-triggered), and periodic and/or aperiodic tasks, with distinct periodicity if ever. Adequate modeling is a prerequisite to the process of scheduling and allocating such tasks onto complex embedded architectural platforms (see AAA approach in foundation section 3.3). Only then can one devise powerful synthesis/analysis/verification techniques to guide designers towards optimized solutions.

Traceability is also an important concern, to close the gap between early requirements and constraints modelling on the one hand, verification and correct implementation of these constraints at the different levels of the development on the other hand.

5. Software

5.1. TimeSquare

Participants: Charles André, Nicolas Chleq, Julien Deantoni, Frédéric Mallet [correspondant].

TimeSquare is a software environment for the modeling and analysis of timing constraints in embedded systems. It relies specifically on the Time Model of the MARTE UML profile (see section 3.2), and more accurately on the associated Clock Constraint Specification Language (CCSL) for the expression of timing constraints.

TimeSquare offers four main functionalities:

1. graphical and/or textual interactive specification of logical clocks and relative constraints between them;
2. definition and handling of user-defined clock constraint libraries;
3. automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction;
4. call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...).

In practice TimeSquare is a plug-in developed with Eclipse modeling tools. The software is registered by the *Agence pour la Protection des Programmes*, under number IDDN.FR.001.170007.000.S.P.2009.001.10600. It can be downloaded from the site http://timesquare.inria.fr/. It has been integrated in the OpenEmbeDD ANR RNTL platform, and other such actions are under way.

5.2. K-Passa

Participants: Jean-Vivien Millo [correspondant], Robert de Simone.

This software is dedicated to the simulation, analysis, and static scheduling scheduling of Event/Marked Graphs, SDF and KRG extensions. A graphical interface allows to edit the Process Networks and their time annotations (*latency, ...*). Symbolic simulation and graph-theoretic analysis methods allow to compute and optimize static schedules, with best throughputs and minimal buffer sizes. In the case of KRG the (ultimately k-periodic) routing patterns can also be provided and transformed for optimal combination of switching and scheduling when channels are shared. KPASSA also allows for import/export of specific description formats such as UML-MARTE, to and from our other TimeSquare tool.

The tool was originally developed mainly as support for experimentations following our research results on the topic of Latency-Insensitive Design. This research was conducted and funded in part in the context of the CIM PACA initiative, with initial support from ST Microelectronics and Texas Instruments.

KPASSA is registered by the Agence pour la Protection des Programmes, under the number IDDN.FR.001.310003.000.S.P.2009.000.20700. it can be downloaded from the site http://www-sop.inria.fr/aoste/index.php?page=software/kpassa.

5.3. SynDEx

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

SynDEx is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. Developed in OCaML it can be downloaded free of charge, under Inria copyright, from the general SynDEx site http://www.syndex.org.

The AAA methodology is described in section 3.3. Accordingly, SynDEx explores the space of possible allocations (spatial distribution and temporal scheduling), from application elements to architecture resources and services, in order to match real-time requirements; it does so by using schedulability analyses and heuristic techniques. Ultimately it generates automatically distributed real-time code running on real embedded platforms. The last major release of SynDEx (V7) allows the specification of multi-periodic applications.

Application algorithms can be edited graphically as directed acyclic task graphs (DAG) where each edge represents a data dependence between tasks, or they may be obtained by translations from several formalisms such as Scicos (http://www.scicos.org), Signal/Polychrony (http://www.irisa.fr/espresso/Polychrony), or UML2/MARTE models (http://www.omg.org/technology/documents/profile_catalog.htm).

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). In order to deal with heterogeneous architectures it may feature several components of the same kind but with different characteristics.

Two types of non-functional properties can be specified for each task of the algorithm graph. First, a period that does not depend on the hardware architecture. Second, real-time features that depend on the different types of hardware components, ranging amongst *execution and data transfer time, memory, etc.*. Requirements are generally constraints on deadline equal to period, latency between any pair of tasks in the algorithm graph, dependence between tasks, etc.

Exploration of alternative allocations of the algorithm onto the architecture may be performed manually and/or automatically. The latter is achieved by performing real-time multiprocessor schedulability analyses and optimization heuristics based on the minimization of temporal or resource criteria. For example while satisfying deadline and latency constraints they can minimize the total execution time (makespan) of the application onto the given architecture, as well as the amount of memory. The results of each exploration is visualized as timing diagrams simulating the distributed real-time implementation.

Finally, real-time distributed embedded code can be automatically generated for dedicated distributed real-time executives, possibly calling services of resident 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. To this date, executives kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be achieved at reasonable cost following these examples as patterns.

5.4. SAS

Participants: Daniel de Rauglaudre [correspondant], 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 main contribution of SAS, when compared to other commercial and academic softwares of the same kind, is that it takes into account the exact preemption cost between tasks during the schedulability analysis. Beside usual real-time constraints (precedence, strict periodicity, latency, etc.) and fixed-priority scheduling policies (Rate Monotonic, Deadline Monotonic, Audsley^{++}, User priorities), SAS additionaly allows to select dynamic scheduling policy algorithms such as Earliest Deadline First (EDF). The resulting schedule is displayed as a typical Gantt chart with a transient and a permanent phase, or as a disk shape called "dameid", which clearly highlights the idle slots of the processor in the permanent phase.

For a schedulable task system under EDF, when the exact preemption cost is considered, the period of the permanent phase may be much longer than the least commun multiple (LCM) of the periods of all tasks, as often found in traditional scheduling theory. Specific effort has been made to improve display in this case. The classical utilization factor, the permanent exact utilization factor, the preemption cost in the permanent phase, and the worst response time for each task are all displayed when the system is schedulable. Response times of each task relative time can also be displayed (separately).

SAS is written in OCaML, using CAMLP5 (syntactic preprocessor) and OLIBRT (a graphic toolkit under X). Both are written by Daniel de Rauglaudre. It can be downloaded from the site http://pauillac.inria.fr/~ddr/sas-dameid/.

6. New Results

6.1. Logical time in Model-Driven Engineering embedded design

Participants: Charles André, Frédéric Mallet, Julien Deantoni, Marie-Agnès Peraldi Frati, Arda Goknil, Nicolas Chleq.

6.1.1. TimeSquare

We progressed our work on the foundations of logical time modeling as present in MARTE Time Model and our CCSL clock constraint specification language, while continuing the development of the TimeSquare tool environment which supports this in practice. A technical position paper was presented to the international TOOLS conference [22].

6.1.2. ECL (Event Constraint Language

Our contributions on CCSL and Time Model to the MARTE profile are part of the standard, but so far expressed in a syntax that is clearly distinct of the former UML notations. On the other hand, UML provides a textual language, named OCL, to express well-formedness constraints on diagram models and metamodels. While the original objectives were quite different, it seemed tempting to extend or adapt the general OCL philosophy, and to apply it then to timing and performance constraints as targeted by CCSL. The goal is to able the description of MoCs in an appropriate syntax, at metamodeling level. The result was a new syntax, called ECL for event contraint language, endowed with the well-established, sound timing interpretation as in CCSL. This work was reported in [40].

6.1.3. Logical time clocks to schedule data-flow models

Data-flow models can be used to capture data dependencies from applications, execution platforms and allocations. Most of the time such data dependencies impose only a partial order on the execution of application elements onto the execution platform and allow several allocation schemes. In [38], we have shown how to use logical time and CCSL constraints to capture explicitly the partial order imposed by the data-dependencies without imposing a total order. This work of representation expressivity then paved the way for analysis studies on time refinement, described in 6.3.

6.1.4. Timing requirement modeling

One of the weak points of UML regarding a complete system design flow is its poor treatment of requirement capture (although this is partly corrected in the SysML profile). When requirements are made on timing aspects and logical time (as in our advocated approach), the relevant syntactic expressivity must be provided. We worked on the definition of a Domain-Specific Language (DSL for Timing Requirements engineering. The results were presented in [24], then applied to system specification in the context of the work described in section 6.6.

6.2. Semantic translation of CCSL constraints into appropriate Büchi automata for trace recognition

Participants: Frédéric Mallet, Julien Deantoni, Robert de Simone, Ling Yin.

Our CCSL language expresses timing and scheduling constraints for a system, based on the notion of abstract logical clocks providing time events, and contraints linking them with relations of "asynchronous" nature (precedence, faster than) or of "synchronous" origin (subclocking, included in). Of course in a large system design both types coexist, and functional definitions also live next to declarative specifications to allow several timing solutions. Such a solution, called a schedule, must enforce that each logical clock either ticks endlessly, or terminates properly, in a way that globally respects the constraints. In previous works we have shown how a large variety of semantic scheduling constraints from the literature could soundly be represented in CCSL.

This year we focused on the semantic foundation of our CCSL language, by defining a structral operational semantic translation into a specific type of transition systems. Because we deal with infinite traces we had to consider acceptance mechanisms such as Büchi repeated states (as already used for translation of LTL temporal logic formulae in classical model-checking). Next we found out that, while state-labeled acceptance conditions were fine to obtain a direct and intuitive translation of individual constraints, building the composition of such models when dealing with multiple constraints was much easier in the case of *transition-labeled* Büchi automata (with repeated acceptance criteria now on transitions); the theory carries over to such case quite naturally, and has already been studied in the past. Finally, because traces must include infinite occurrences for *each* clock, we had to move to so-called *extended* Büchi automata, again a model already studied previously.

We provided a complete semantic translation for all CCSL kernel constructs. Most importantly, we provided an efficient and simple fix-point algorithm to check the existance of a valid schedule, based on the type of automata just defined. This is (we believe) a genuine improvement on existing results, with potential applications outside our direct scope. These results are presented in a technical report, submitted for publication [46].

6.3. Timing refinement for multidimensional dataflow models using MARTE Time Model

Participants: Frédéric Mallet, Julien Deantoni, Jean-Vivien Millo.

Extensions of dataflow process networks have been proposed (as multidimensional SDF) to combine task parallelism (as in traditional process networks) with intensive data parallelism (as proposed in the Array-OL/Gaspard2 formalism developed in the DaRT EPI, for instance). The prospect of scheduling (seen as precise time cycle allocation) is here more complex, because of possible trade-offs between the granularity of treatments at task level *vs.* the size of data arrays that are handled uniformally in parallel inside each task. We considered how these phenomena could be represented (if not solved) inside the framework of MARTE Time Model and logical clocks, so as to handle such design issues in a well-defined MDE approach. Additionally, we used the MARTE platform description to specify how the previous models are refined through mapping allocation. The resulting modeling framework was presented in a journal article [19]. This work was conducted jointly with P. Boulet, from DaRT EPI, and C. Glitia, former DaRT PhD and Aoste postdoc student.

6.4. Process Network analysis

Participants: Robert de Simone, Jean-Vivien Millo.

6.4.1. K-periodic routing schemes for Network-on-Chip data traffic

This year we considered more specifically the issue of exploiting the predictable routing schemes of our KRG models, expressed as infinite binary words to indicate the sucessive branching directions at merge/select switch nodes, in order to encode data traffic patterns expanded at compile time, when mapping applications expressed under the form of dataflow process networks onto processor arrays in manycore architectures based on network-on-chip interconnects. To show the potential impact of such predicatble compile-time routing patterns, we stdudied as a typical example a fulll (all-to-all) broadcast algorithm on a mesh topology, connecting mode-less computation nodes as in the theory of cellular automata. This resulted in a precise recursive definition of routing patterns, which achieve an optimal data propagation (broadcast implemented as multicast), given the availability of actual links in the NoC topology. This result was presented at the Autamata'2012 conference [30], and an expanded version is available as technical report [44].

A wider view of the approach, and its potential benefits, are described in a technical report [43], submitted for publication.

6.4.2. Optimal data placement for process network scheduling

The topic of efficient scheduling of dataflow process network traffic to optimize both throughput and buffer queue sizing has given rise to a huge literature starting with seminal works in [49], [47], [56]. It has recently been given new impulse due to the advent of manycore architectures (see above). We conducted a number of theoretical works, to establish how such optimal computation scheduling can be best achieved in configurations where data are evenly distributed and streched in time across the (process) network. While this result is intuitively obvious, we formalized precisely what evenly distributed technically means, with the notion of balanced/mechanical words going a long way back in formal language theory, and we demonstrated that under such assumptions optimal schedules could be constructed *in a fully analytical way*, without any symbolic simulation steps or behavior expansion. The result was accepted for publication in a journal article [20].

6.5. Transformation from MARTE Time Model and CCSL to formal analysis models

Participants: Frédéric Mallet, Ling Yin.

This work was conducted in the context of an on-going collaboration with the Software Engineering Institute (SEI) of East Normal China University (ECNU) at Shanghai, which led altogether in part to the DAESD Associated-team, followed by a LIAMA joint project proposal recently submitted (HADES), and the co-supervision by Frédéric Mallet (together with Professor Jing Liu from ECNU) of the PhD thesis of Yin Ling. Yin Ling spent a one-year visit in our team, funded on a chinese governemental grant.

We studied the efficient and sound formal translation of a subset of CCSL contraints into the PROMELA/SPIN formalism, to benefit from model-checking formal analysis features in this environment. The translation is not completely direct, as synchronous simultaneity is not a native notion of PROMELA, and has to be encoded as atomicity. The motivating principles and translation details are provided in [42]. A similar attempt could be considered in the future, this time with the synchronous model-checker SMV, which allows coumpound instantaneous atomic behaviors.

Another line of research was initiated at ECNU to consider *logical continuous time*, while most of our current work considers only discrete time (while MARTE Time Model considers both). Considerations on *hybrid state diagrams* , inviting the expressive power of formal hierarchical hybrid automata models into the MDE design space of UML MARTE, were investigated in [27].

6.6. Use of MARTE Time Model and Logical Time in automotive design and AUTOSAR/TADL

Participants: Marie-Agnès Peraldi Frati, Julien Deantoni, Arda Goknil.

Precise timing constraint modeling and analysis [26], [33] is a key point for the correct development of automotive electronics. EAST-ADL and AUTOSAR has been adopted as standards in automotive industry. The timing model (TADL :Time augmented Description Language) of these standards raises different issues, mainly concerning the precise modeling of the multi clock characteristics of distributed systems together with parameterized timing expressions. In the ITEA TIMMO-2-USE project [35] 8.3.2.1, we conducted a work [34], [35], on extending TADL with an explicit notion of multiple time bases for modeling the various temporal referentials used in an automotive design (clocks from different ECUs, motor position, etc). Additionally, timing constraints are augmented with parameters, which can be free at the highest abstraction level and then progressively defined during the design process. As a result, a symbolic timing expression in TADL2 is possibly made of a suitable set of arithmetic operators mixing symbolic identifiers (not necessarily set variables) and referring to different time bases. One typical use of this feature is to capture unknown configuration parameters for time budgeting; another one is to relate constraints in different time-bases to each other. Inherent to this work is also the study of the allowable ranges for symbolic values that are dictated by a set of constraints.

6.7. Multiview modeling and power intent in Systems-on-chip

Participants: Carlos Gomez Cardenas, Ameni Khecharem, Jean-François Le Tallec, Frédéric Mallet, Julien Deantoni, Robert de Simone.

6.7.1. High-level power management modeling

One of the concern of the UML MARTE profile is to allow non-functional property modeling, so that the same system bare description can be annotated in a number of views. In our case, combined with our logical time framework, such properties can be made as time-depending, inside potentially distinct views. We examplified this approach by dealing to a large extent with the example of low-power design and energy modeling in the case of Systems-on-Chip (SoC) in the mobile phone domain. Pure power/thermal modeling can be realized, based on the system global architecture, then made operational with the use of logical time controllers triggering power management functionalities.

Thermal/power simulation models are usually relying on continuous time. Therefore we considered the issue of *logical continuous* time, in an early attempt at combining simulation of continuous time power/thermal models with intrinsically discrete functional aspects. A prototype was realized in Scicos, as part of Ameni Khecharem master internship.

This work was conducted in the context of Carlos Gomez PhD thesis, and in collaboration with several partners inside the ANR HeLP project. It should be continued in the forthcoming PhD thesis of Ameni Khecharem, just started in the context of the follow-up ANR HOPE project, which will consider specific issues of hierachical power modeling and compositional power management (as an example of incremental multiview aspects).

6.7.2. IP-XACT

In this context of high-level power modeling and multiview concerns, we considered the emerging Accelera standard IP-XACT, made to provide easy-to-plug interfaces and Architecture Description Language (ADL) to allow simple assembly of hardware IP components into well-behaved SoCs. More specifically we provided means to annotate such interface with extra informations, directly borrowed from UML MARTE NFP properties, to handle power and thermal aspects. A number of model transformations back and forth between MARTE and (extended) IP-XACT were realized, and extraction of IP-XACT compliant interfaces from proprietary SystemC code describing the elementary IP component tehmselves has been defined and implemented as well.

This work was initiated as part of a project with STMicroelectronics, inside the nano2012 programme (ended 2011), and continued as part of the ANR HeLP collaboration. It resulted in the PhD thesis of Jean-François Le Tallec (who remained in the team for a couple of months later to complete the prototype implementation) [16].

6.8. Correct and efficient implementation of polychronous formalisms

Participants: Thomas Carle, Manel Djemal, Dumitru Potop Butucaru, Robert de Simone, Yves Sorel.

We extended our work on extending the AAA methodology for polychronous processes, by providing a better integration of clock analysis in the various phases of the implementation process (allocation, scheduling, pipelining, etc.). We also considered a wider range of implementation targets (time-triggered, MPSoC) and non-functional constraints (partitioning).

6.8.1. Time-Triggered Platform targets

Our first result this year concerns the automatic scheduling and code generation for time-triggered platforms. We extended our previous results in two significant ways. First, we designed a novel approach for specification of real-time features of time-triggered systems, with deadlines longer than periods; this allows a faithful representation of complex end-to-end flow requirements. Second, we provided new algorithms for off-line pipelined scheduling of these specifications onto partitioned time-triggered architectures *à la* ARINC 653; allocation of time slots/windows to partitions can be either complete or partially provided, or synthesized by our tool. Automatic allocation and scheduling onto multi-processor (distributed) systems with a global time base becomes feasible, taking into account communication costs. For single processors, we allow the generation of fully compliant ARINC653/APEX implementation code.

This work was mainly carried out inside the FUI Parsec 8.2.2.2 (which funds the PhD thesis of T. Carle) and P 8.2.2.1 projects, as well as a collaboration with ASTRIUM Space Transportation. First results are presented in a technical report, submitted for publication [39].

6.8.2. Multi-Processor System-on-Chip (MP-SoC) targets

Our second contribution concerns the automatic allocation and real-time scheduling over MPSoC (multi-processor on chip) architectures with NoC (network-on-chip) interconnect. One must take into account the specific 2D mesh network-on-chip topology, and synthesize the NoC routing patterns. This work provides operational execution support for the contributions described in 6.9.

6.8.3. The LoPhT tool

Our recent work on extending the AAA methodology with better handling of execution conditions, with pipelining and pipelined scheduling, and with specific real-time scheduling and code generation techniques for time-triggered/partitioned and MPSoC platforms resulted in the development of a new scheduling and code generation toolbox, called LoPhT (for Logical to Physical Time Compiler).

6.9. Programmable On-Chip Networks

Participants: Thomas Carle, Manel Djemal, Dumitru Potop Butucaru, Robert de Simone, Zhen Zhang.

Modern computer architectures are increasingly relying on multi-processor systems-on-chip (MPSoCs), with data transfers between cores and memories managed by on-chip networks (NoC). This reflects in part a convergence between embedded, general-purpose PC, and high-performance computing (HPC) architecture designs.

Efficient compilation of applications onto MPSoCs remains largely an open problem, with the issue of best mapping of computation parts (threads, tasks,...) onto processing resources amply recognized, while the issue of best use of the interconnect NoC to route and transfer data still less commonly tackled. In the most general case, dynamic allocation of applications and channel virtualization can be guided by user-provided information under various forms, as in OpenMP, CUDA, OpenCL and so on. But then there is no clear guarantee of optimality, and first attempts by non-experts often show poor performances in the use of available computing power. Conversely there are consistent efforts, in the domains of embedded and HPC computing, aiming at automatic parallelization, compile-time mapping and scheduling optimization. They rely on the fact that applications are often known in advance, and deployed without disturbance from foreign applications, and without uncontrolled dynamic creation of tasks. Our contribution follows this “static application mapping” approach.

An optimal use of the NoC bandwidth should authorize data transfers to be realized according to (virtual) channels that are temporarily patterned to route data “just-in-time”. Previous works have identified the need for Quality of Service (QoS) in “some” data connections across the network (therefore borrowing notions from macroscopic networks, say internet and its protocols). But our experience with the AAA methodology strongly suggests that optimal NoC usage should result from a global optimization principle (embodied in a form of the AAA methodology), as opposed to a collection of local optimizations of individual connections. Indeed, various data flows with distinct sources and targets will nevertheless be highly concerted, both in time and space, like in a classical pipelined CPU, where the use of registers (replaced in our case with a complex NoC) is strongly synchronized with that of the functional units.

One main problem in applying such a global optimization approach is to provide the proper hardware infrastructures allowing the implementation of optimal computation and communication mappings and schedules. Our thesis is that optimal data transfer patterns should be encoded using simple programs configuring the router nodes (each router being then programmed to act its part in the global concerted computation and communication scheme).

We addressed this problem in the framework of our collaboration with the "Embedded Systems- on-Chips" department of the LIP6 laboratory, one of the main site of expertise for SoC/NoC design and Hardware/software codesign. This collaboration first materialized with the co-supervision of M. Djemal’s PhD thesis. We concretely supported our proposed approach by extending the DSPIN 2D mesh network-on-chip (NoC) developed at UPMC- LIP6. In this NoC, we replace the fair arbitration modules of the NoC routers with static, micro-programmable modules that can enforce a given packet routing sequence, as specified by small programs. The design of such simple routing schemes can, for instance, be extracted from our results in section 6.4.
We advocate the desired level of expressiveness/complexity for such simple configuration programs, and provide experimental data (cycle-accurate simulations) supporting our choices. We also wrote an architecture synthesis tool that allows simple architectural exploration of MPSoCs using the new DSPINPro NoC. First results in this direction have been presented in the DASIP 2012 conference, where our paper [23] has been short-listed for best paper award.

6.10. Uniprocessor Real-Time Scheduling

Participants: Laurent George, Mohamed Marouf, Daniel De Rauglaudre, Yves Sorel.

6.10.1. Combination of Non-Preemptive and Preemptive Tasks

We focused on fixed priority scheduling for a combination of non-preemptive strict periodic tasks in conjunction with preemptive sporadic tasks, that we extended to software fault tolerance [29]. We first investigated the transient phase for non-preemptive strict periodic tasks and we proved that its length is smaller than the transient phase for preemptive periodic tasks. Then, we determined the worst case scenario for preemptive sporadic tasks where the Worst Case Response Time (WCRT) can be obtained in the presence of strict periodic tasks. We proved that these release times belong only to the permanent phase of strict periodic tasks, and thus that the schedulability analysis for sporadic tasks can be restricted to the permanent phase. For preemptive sporadic tasks, we extended the classical necessary and sufficient schedulability condition based on the worst case response time computation to take into account non-preemptive strict periodic tasks. Finally, we considered software fault tolerance in the particular case where each primary strict periodic task has an alternate sporadic task which is released when the primary task fails. The schedulability analysis guarantees that even if all strict periodic tasks fail then all their respective alternate tasks will meet their deadlines.

6.10.2. Formal Proofs of Real-Time Scheduling Theorems

We completed two formal proofs of theorems in Coq on scheduling of fixed priority real-time preemptive tasks: one dealing with the sizes of busy periods (about 3500 lines of Coq), and another one dealing with response time (about 5200 lines of Coq). A monograph about these proofs, together with the formal check in Coq of scheduling conditions of strict periodicity, presented in the conference JFLA 2012 [37], have been started (currently about 70 pages).

6.11. Multiprocessor Real-Time Scheduling

Participants: Abderraouf Benyahia, Laurent George, Mohamed Marouf, Falou Ndoye, Simon Nivault, Yves Sorel, Cécile Stentzel, Meriem Zidouni.

6.11.1. Non-Preemptive Partitioned Fault Tolerant Scheduling

We addressed partitioned multiprocessor scheduling of non-preemptive strict periodic tasks which is extended thereafter to hardware fault tolerance [17].

In order to schedule a task set of non-preemptive strict periodic tasks on a multiprocessor platform, we partitioned this task set into subsets of tasks, each one is scheduled on a single processor using our proposed uniprocessor scheduling algorithm. The partition is carried out according to an enhanced “First Fit” algorithm that balances the load of the tasks on all the processors. However, inter-processors communications can lead to delay task execution. Thus, we determined the start time of each task taking into account the communication delay between this latter task and its predecessor tasks. Also, as inter-processor communications may generate a transient phase, we computed the length of the transient phase.

We proposed a fault tolerant real-time scheduling algorithm which allows hardware processors and/or buses faults, and conserves the strict periodicity of each task. We also proposed a graph transformation algorithm, applied on the task graph, which generates redundancies of tasks as well as dependencies. The transformation adds also selector tasks which choose data coming from the non failing processors and buses. That algorithm is based on exclusion relations to assign redundant tasks (resp. dependencies) to different processors (resp. busses). Then, we extended the previous partitioned multiprocessor scheduling algorithm to manage fault tolerance taking into account these exclusion relations.

This approach was successfully implemented on a CyCabs electric vehicle in a real-time fault tolerant tracking application where some processor or some bus could fail without any consequence on the proper execution of the application, i.e. same functional behaviour and real-time constraints satisfied.

6.11.2. Partitioned Scheduling with Exact Preemption Cost

Preemption allows a better scheduling success ratio but has a cost that must not be neglected in safety critical applications of domains such as avionic, automotive, etc. We focused on partitioned multiprocessor scheduling of independent preemptive periodic real-time tasks, while taking into account the exact preemption cost with the $\oplus$ operation formerly proposed by Meumeu and Sorel [10]. We improved the “greedy” heuristic proposed last year and compared it with the “Best-Fit” (BF) and “Worst-Fit” (WF) heuristics classically used in partitioned multiprocessor scheduling, but extended to take into account the exact preemption cost. We also compared our heuristic with an exact “Branch and Bound” algorithm with the same extension. The first comparison shows that the task allocation found by our heuristic gives a better response time than those found by WF and BF. This is due to the fact that the execution of the tasks is better parallelized. On the other hand, BF and WF heuristics execute a bit faster than our heuristic because they do not use all the available processors contrary to our heuristic which has the advantage to improve the load balancing of the tasks on all the processors.

Then, we addressed the scheduling of preemptive periodic real-time tasks with dependence constraints involving task precedences and data dependences. We considered harmonic tasks, i.e. periods of tasks are multiple or equal, to avoid loss of data. In order to satisfy data dependence constraints, we modified the release dates and deadlines of the dependent tasks according to the reception date of the data. In addition, data dependences between tasks mean to share data between dependent tasks which can cause deadlock and priority inversion problems. In order to solve these problems while taking into account the preemption cost, we proposed a new schedulability condition based on an extension of the $\oplus$ operation. We plan to propose a mutiprocessor scheduling heuristic based on that condition applied on tasks with modified release dates and deadlines.

6.11.3. Semi-partitioned Scheduling

Semi-partitioned multiprocessor scheduling stands between partitioned and global scheduling, the latter allowing migrations. We mainly addressed the semi-partitioned scheduling approach where the Worst Case Execution Time (WCET) of a job can be portioned, each portion being executed on a dedicated processor, according to a static pattern of migration. A job is migrated at its local deadline, computed from the deadline of the task it belongs to. We have studied this approach in the context of a fork/join task model with thread parallelism. A task is composed of a sequence of segments that can be parallelized in threads, if needed. The local deadlines depends on the number of parallel threads assigned to each segment.

6.11.4. Code Generation for Multicore

This work was carried out in the OPENPROD ITEA project 8.3.2.2. xMod developed by IFPEN (IFP Energies Nouvelles), is an heterogeneous model integration environment that allows model importation from specific tools such as Simulink, AMSIM, etc. It also provides as a virtual instrumentation laboratory. In order to make xMod being able to run simulations with hardware-in-the-loop environment, we developed a new SynDEx executive kernel based on the kernel, dedicated to Windows/RTX, developed last year. That executive kernel is used with the macro-code generated by SynDEx to produce a real-time executable code that can drive the execution (real-time multi-core distribution and synchronized execution) of the models imported by xMod and simulated in the virtual instrumentation laboratory. This prototype as well as the report describing the corresponding achieved works, are the final deliverable of the OPENPROD project.

Furthermore, a French and English SynDEx code generation reference manual has been written to help future SynDEx users and maintainers to generate real-time code for already supported architectures or new ones.

6.11.5. Gateway with Modeling Languages for Certified Code Generation

This work was carried out in the P FUI project 8.2.2.1. We provide inside the project expertise mainly on schedulability analysis and automatic generation of distributed real-time code. In this context, we developed a gateway between UML/MARTE and SynDEx. From a model specified with UML (Activity Diagram to specify algorithms and Composite Structure Diagram to specify multicomponent architectures) and refined with the UML profile MARTE (Modeling and Analysis of Real-Time Embedded Systems), we use the gateway to generate automatically distributed real-time application specified in the SynDEx format. Currently, we intend to provide a gateway between the GeneAuto language and SynDEx. The GeneAuto language is a subset of the future pivot P language. We presently deal with the part of the GeneAuto language corresponding to Simulink for data-flow modeling and we plan to deal soon with the part corresponding to Stateflow for control-flow modeling (composition of automata).

6.11.6. SynDEx Updates

We continued the software developments for the future version 8 of SynDEx which will feature a new software architecture to allow better functionality evolutions and maintenance. On the other hand in the COTROS ADT ("Génération de code temps réel distribué optimisé et sûr"), we completed the tests on the new automatic code generator for the current version 7 of SynDEx. This new generator produces code for mono-periodic and multi-periodic applications with condition and repetitive control structures, for the different hardware architectures supported by SynDEx. We developed a checker for the generated code that was integrated in the new generator. This checker verifies the correct use of semaphores and consequently the absence of deadlocks in the real-time code. Deadlocks are the most difficult part when dealing with distributed architectures. We achieved also a maintenance report describing the structure and the main features of code generator, as well as the technical choices we did.

6.12. Variability of program execution times on multicore processors

Participants: Sid-Ahmed-Ali Touati, Matias Vara Larsen, Abdelhafid Mazouz.

The activity described here represents the finalization of previous efforts conducted by Sid Touati and members of his groups, initiated before he joined the AOSTE EPI, and which are progressively merged with our own objectives, for results to be reported hopefully next year).

With the massive introduction of multicore platforms on embedded systems, parallel applications gained in performance. However, we showed in previous studies that the performance gain comes with high instability: program execution times vary in important way. We investigated the reasons for this variations and tried to understand the factors that influence program performance variability, that we decompose intro multiple families: factors from the application itself (implemented algorithms, coding technique, synchronization barriers, etc.), factors from the execution environment (OS effects, thread scheduling, Input/Ouput operations) and factors from the underlying hardware (micro-architecture, memory hierarchy, speculative execution, hardware data prefectching, etc.). Now, we have better understanding to these factors thanks to the work of two students:

1. Mr. Abdelhafid Mazouz who defended his PhD under the direction of Sid Touati at the university of Versailles in 11th of December 2012. The title of his PhD is "An Empirical Study of Program Performance of OpenMP Applications on Multicore Platforms".
2. Mr. Matias Vara Larsen, intern under the supervision of Sid Touati from February to June 2012, inside the Aoste EPI in Sophia-Antipolis, co-funded under a grant from Inria international internship program). The topic of his internship was to study the influence of he Linux kernels (multiple versions) on the stability of parallel applications.

Last, we published a rigorous statistical protocol in [21] called the Speedup-Test. It is used to analyze valid speedups (performance gain) in presence of performance instability: The Speedup-Test protocol is implemented and distributed as an open source tool based on R software. Our statistical methodology defines a consistent improvement compared with the usual performance analysis method in high-performance computing.

7. Bilateral Contracts and Grants with Industry

7.1. Thales ARCADIA/Melody

Participants: Frédéric Mallet, Robert de Simone.

In the remote context of ARTEMIS CESAR 8.3.1.1 we conducted a specific study of the functional expressiveness of the ARCADIA/Melody environment, developed and deployed internally inside several Thales divisions. A questionnaire was designed by us, according to the various semantic variation points that we identified into this Model-Driven Engineering (MDE) environment. It was then sent to potential users for feedback, and reporting was done together with colleagues from Thales TRT (R&D division) to their management. As a result a number of non-trivial redesign decisions were taken. Our findings were presented through a number of focused meetings held at Thales in the Saclay technopark. While most work was performed at this stage on purely data-flow functional description diagrams, there is an interest inside the company to extend this type of critical survey analysis to extended description models, including event-based control and modes.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. CIM PACA

Participants: Robert de Simone, Ameni Khecharem, Carlos Gomez Cardenas.

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 so far actively participating in the Design Platform (one of the three platforms launched in this context), of which Inria is a founding member.

This year our ANR proposal HOPE was labeled by the regional SCS Cluster, through its ARCSIS/CIM PACA branch for microelectronics design. The project was consequently accepted, and will benefit from support from CIM PACA Design platform to host prototype and commercial software from project members (Synopsys, Docea Power, and Magillem, see 8.2.1.3).

8.2. National Initiatives

8.2.1. ANR

8.2.1.1. RT-Simex

Participants: Julien deAntoni, Frédéric Mallet.

The RT-Simex 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 OBEO, a software publishing company based in Nantes. The project ended in April 2012.

8.2.1.2. HeLP

Participants: Carlos Gomez Cardenas, Ameni Khecharem, Robert de Simone, Jean-Vivien Millo.

The ANR HeLP project deals with joint modeling of functional behavior and energy consumption for the design of low-power heterogeneous SoCs. Partners are ST Microelectronics and Docea Power (SME) as industrial; Inria, UNS (UMR LEAT), and VERIMAG (coordinator) as academics. Our goal in this project is twofold: first, combine SoC modeling with temporal behavior and logical time with energy/power modeling as extra annotations on MARTE models; second, compare the capacities of high-level SystemC TLM abstraction with that of Esterel seen as a multiclock formalism based on logical abstract time.

The PhD thesis of Carlos Gomez, while not formerly funded by this project, is closely linked to its results (by providing a MDE metamodel with non-functional multiview aspects, such as performance, power and temperature. Several transformation links were realized, towards AcePlorer tool by DOCEA POWER, partner of the project, or also (as part of Ameni Khecharem internship) towards Scilab for simulation execution. Some of this work will be continued in the forthcoming ANR HOPE project.

8.2.1.3. HOPE

Participants: Carlos Gomez Cardenas, Ameni Khecharem, Robert de Simone.

This project was only recently started, with a kick-off meeting in November. Original proponents were UMR LEAT, Texas Instruments, Synopsys, Docea Power, Magillem, and ourselves. It seems that, due to internal reorganisation, TI might withdraw from the project. Other major semiconductor industrial partners in PACA are being approched for replacement (mainly Intel). The purpose of the HOPE project is to focus on high-level modeling and early estimation of hierarchical power management techniques, with potential synthesis in the end if feasible.

8.2.1.4. GeMoC

Participants: Matias Vara Larsen, Julien deAntoni, Frédéric Mallet.

This project was only recently started, with a kick-off meeting in December. It is admistratively handled by CNRS for our joint team, on the UMR I3S side. Partners are Inria (Triskell EPI), ENSTA-Bretagne, IRIT, Obeo, Thales TRT.
The project focuses on the modeling of heterogeneous systems using Models of Computation and Communication for embedded and real-time systems, described using generic means of MDE techniques (and in our case the MARTE profile, and most specifically its Time Model, which allows to specify precise timely constraints for operational semantic definition).

8.2.2. FUI

8.2.2.1. FUI P

Participants: Abderraouf Benyahia, Dumitru Potop Butucaru, Yves Sorel.

The goal of project P is to support the model-driven engineering of high-integrity embedded real-time systems by providing an open code generation framework able to verify the semantic consistency of systems described using safe subsets of heterogeneous modeling languages, then to generate optimized source code for multiple programming (Ada, C/C++) and synthesis (VHDL, SystemC) languages, and finally to support a multi-domain (avionics, space, and automotive) certification process by providing open qualification material. Modeling languages range from behavioural to architectural languages and present a synchronous and asynchronous semantics (Simulink/Matlab, Scicos, Xcos, SysML, MARTE, UML),
See also: http://www.open-do.org/projects/p/
Partners of the project are: industrial partners (Airbus, Astrium, Continental, Rockwell Collins, Safran, Thales), SMEs (AdaCore, Altair, Scilab Enterprise, STI), service companies (ACG, Aboard Engineering, Atos Origins) and research centers (CNRS, ENPC, Inria, ONERA).

8.2.2.2. FUI PARSEC

Participants: Dumitru Potop Butucaru, Thomas Carle, Zhen Zhang, Yves Sorel.

The PARSEC Project aims at providing development tools for critical real-time distributed systems requiring certification according to the most stringent standards such as DO-178B (avionics), IEC 61508 (transportation) or Common Criteria for Information Technology Security Evaluation. The approach proposed by PARSEC provides an integrated toolset that helps software engineers to meet the requirements associated to the certification of critical embedded software. Partners of the project are: Alstom, Thales, Ellidiss, OpenWide, Systerel, CEA, InriaS, Telecom ParisTech.
See also: http://www.systematic-paris-region.org/sites/default/files/exports/projets/fichiers/ProjetPARSEC_BookSystematic2012.pdf.

8.3. European Initiatives

8.3.1. ARTEMIS Projects

8.3.1.1. CESAR

Participant: Robert de Simone.

Title: CESAR

Duration: February 2009 - June 2012

Coordinator: AVL - GmbH (Austria)

Others partners: AIRBUS Operations GbmH (Germany), AIRBUS Operations SAS (France), ABB AS (Norway), ABB AB (Sweden), AbsInt Angewandte Informatik GmbH (Germany), ACCIONA Infraestructuras S.A. (Spain), Ansaldo STS S.p.A. (Italy), ASTRIUM SAS (France), AIRBUS Operations Limited (United Kingdom), Aristotle University of Thessaloniki (Greece), Commissariat à l´Energie Atomique et aux Energies Alternatives (France), CNRS (France), Centro Ricerche Fiat S.C.p.A. (Italy), Critical Software S.A. (Poland), Danieli Automation S.p.A. (Italy), Delphi France SAS (France), Deutsches Zentrum für Luft- und Raumfahrt e.V. (Germany), Dassault Systemes (France), EADS Deutschland GmbH (Germany), Fondacion Tecnalia Research & Innovation (Italy), ESTEREL Technologies SA (France), Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung e.V. (Germany)

See also: http://www.cesarproject.eu/

Abstract: CESAR stands for Cost-efficient methods and processes for safety relevant embedded systems and is a European funded project from ARTEMIS JOINT UNDERTAKING (JU). The three transportation domains automotive, aerospace, and rail, as well as the automation domain share the need to develop ultra-reliable embedded systems to meet societal demands for increased mobility and ensuring safety in a highly competitive global market. To maintain the European leading edge position in the transportation as well as automation market, CESAR aims to boost cost efficiency of embedded systems development and safety and certification processes by an order of magnitude. CESAR pursuits a multi-domain approach integrating large enterprises, suppliers, SME's and vendors of cross sectoral domains and cooperating with leading research organizations and innovative SME's.

Upon completion, CESAR was awarded an ARTEMIS honorary mention for achievement.

8.3.1.2. PRESTO

Participants: Frédéric Mallet, Arda Goknil, Julien Deantoni, Marie-Agnès Peraldi Frati, Robert de Simone.

Title: PRESTO

Duration: April 2011 - March 2014

Coordinator: Miltech (Greece)

Others partners: TELETEL S.A. (Greece), THALES Communications (France), Rapita Systems Ltd. (United Kingdom), VTT (Finland), Softeam (France), THALES (Italy), MetaCase (Finland), Inria (France), University of LâAquila (Italy), MILTECH HELLAS S.A (Greece), PragmaDev (France), Prismtech (United Kingdom), Sarokal Solutions (Finland).

See also: http://www.cesarproject.eu/

Abstract: The PRESTO project aims at improving test-based embedded systems development and validation, while considering the constraints of industrial development processes. This project is based on the integration of test traces exploitation, along with platform models and design space exploration techniques. Such traces are obtained by execution of test patterns, during the software integration design phase, meant to validate system requirements. The expected result of the project is to establish functional and performance analysis and platform optimisation at early stage of the design development. The approach of PRESTO is to model the software/hardware allocation, by the use of modelling frameworks, such as the UML profile for model-driven development of Real Time and Embedded Systems (MARTE). The analysis tools, among them timing analysis including Worst Case Execution Time (WCET) analysis, scheduling analysis and possibly more abstract system-level timing analysis techniques will receive as inputs on the one hand information from the performance modelling of the HW/SW-platform, and on the other hand behavioural information of the software design from tests results of the integration test execution.

8.3.2. Collaborations in European Programs, except FP7

8.3.2.1. ITEA2 Timmo2Use

Participants: Marie-Agnès Peraldi Frati, Julien DeAntoni, Arda Goknil, Jean-Vivien Millo, Yves Sorel.

Program: ITEA2

Project acronym: Timmo2Use

Project title: TIMing MOdel, TOols, algorithms, languages, methodology, and USE cases

Duration: October 2010 - October 2012

Coordinator: Volvo Technology AB (Sweden)

Other partners: AbsInt Angewandte Informatik GmbH (Germany), Arcticus Systems AB (Sweden), Chalmers University of Technology (Sweden), Continental Automotive GmbH (Germany), Delphi France SAS (France), dSPACE GmbH (Germany), INCHRON GmbH (Germany), Institut National de Recherche en Informatique et Automatique (France), Mälardalen University (Sweden), Rapita Systems Ltd. (United Kingdom), RealTime-at-Work (France), Robert Bosch GmbH (Germany), Symtavision GmbH (Germany), Technische Universität Braunschweig (Germany), Time Critical Networks (Sweden), Universität Paderborn (Germany).

See also: http://timmo-2-use.org/

Abstract: TIMMO develops different types of timing constraints and dynamic behaviour formalisms, to be used inside the supply chain and the complex development process in distributed real-time automotive system design. TIMMO-2-USE stands for TIMing MOdel - TOols, algorithms, languages, methodology, and USE cases which summarizes the main objectives of the project, i.e., the development of novel tools, algorithms, languages, and a methodology validated by use cases.

The project provides partial funding for the postdoctoral positions of Jean-Vivien Millo and Arda Goknil.

8.3.2.2. ITEA2 OPENPROD

Participants: Simon Nivault, Yves Sorel.

Program: ITEA2

Project acronym: OpenProd

Project title: Open Model-Driven Whole-Product Development and Simulation Environment

Duration: June 2009 - May 2012

Coordinator: Siemens Industrial TurboMachinery AB (Sweden)

Other partners: Appedge (France), Bosch Rexroth AG (Sweden), CEA LIST (France), EADS Innovation Works (France), Electricité De France (France), Equa Simulation AB (Sweden), ETH Zürich (Switzerland), Fachhochschule Bielefeld (Germany), Fraunhofer FIRST (Germany), IFP (France), Inria Rocquencourt (France), INSA Lyon (France), Linköping University (Sweden), LMS Imagine (France), MathCore Engineering AB (Sweden), Metso Automation (France), Nokia (Finland), Plexim GmbH (Germany), Pöyry Forest Industry (Finland), PSA Peugeot Citroen (France), Siemens AG, Sector Energy (Germany), SKF Sverige AB (Sweden), Technische Universität Braunschweig (Germany), TLK Thermo GmbH (Germany), VTT Technical Research Centre (Finland), XRG Simulation GmbH (Germany).

See also: http://www.ida.liu.se/~pelab/OpenProd/

Abstract: The OPENPROD project is developing an open whole-product, model-driven systems development, modelling and simulation (M&S) environment that integrates the leading open industrial software development platform Eclipse with open-source modelling and simulation tools such as OpenModelica and industrial M&S tools and applications. The project will enable a more formalised validation of production to cut time to market and ensure higher quality, using open solutions which will have a high impact, based on easy uptake and wide dissemination.

8.4. International Initiatives

8.4.1. Inria Associated Teams

8.4.1.1. DAESD

Title: Distributed/Asynchronous and Embedded/synchronous Systems Development

Inria principal investigator: Robert de Simone

International Partner (Institution - Laboratory - Researcher):

East China Normal University (China) - SEI-Shone - Yixiang Chen

Duration: 2012 - 2014

See also: https://team.inria.fr/DAESD/

The development of concurrent and parallel systems has traditionally been clearly split in two different families: distributed and asynchronous systems on one hand, now growing very fast with the recent progress of the Internet towards large scale services and clouds; embedded, reactive, or hybrid systems on the other hand, mostly of synchronous behaviour. The frontier between these families has attracted less attention, but recent trends, e.g. in industrial systems, in *Cyber-Physical systems*, or in the emerging *Internet of Things*, give a new importance to research combining them. The aim of the DAESD associate team is to combine the expertise of the Oasis and Aoste teams at Inria, the SEI-Shone team at ECNU-Shanghai, and to build models, methods, and prototype software tools inheriting from synchronous and asynchronous models. We plan to address modelling formalisms and tools, for this combined model; to establish a method to analyze temporal and spatial consistency of embedded distributed real-time systems; to develop scheduling strategies for multiple tasks in embedded and distributed systems with mixed constraints. In parallel with our research collaboration this Associate Team, the SEI-Shone lab is organizing a workshop in Shanghai, with a first edition in Nov. 2011, on "Distributed - Asynchronous and Embedded - synchronous Systems Development".

8.4.2. Participation In International Programs

8.4.2.1. LIAMA

Following the DAESD associated-team, a proposal for a LIAMA project with ECNU Shanghai, named HADES, has been presented recently at the LIAMA steering committee in December 2012. It is a joint proposal with the OASIS EPI.

8.5. International Research Visitors

8.5.1. Visits of International Scientists

Jagadish Suryadevara (IDT, Mälardalen University, Sweden) visited us for two months in May/June 2012.

8.5.1.1. Internships

Matias Ezequiel VARA LARSEN (from Mar 2012 until Jun 2012)

Subject: Study of the influence of Linux operating system on OpenMP applications performances on multicore processors

Institution: National University of La Plata (Argentina)

9. Dissemination

9.1. Scientific Animation

Robert de Simone

Technical Program Committee: FDL2012, SIES2012, SSMBSE2012, MEMOCODE2012, ESLsyn2012.

Board of Administrators: CIM PACA Design Platform association

Scientific co-animator: DAS SE (Specific Action Domain on Embedded Systems), Aerospace Valley Competitiveness Cluster

Yves Sorel

Technical Program Committee: RTNS 2012, DASIP 2012.

Editorial Board: Traitement du Signal Journal

Steering Committee: OCDS/SYSTEM@TIC Paris-Region Cluster

Frédéric Mallet

Technical Program Committee: ESLsyn2012.

Laurent George

Technical Program Committee: ECRTS 2012, RTAS 2012, WFCS 2012.

Co-editor: special issue of Real-Time Systems Journal made of best papers from RTNS 2011 conference (with Alan Burns, Univ. of York, UK).

Scientific Co-chair: ACTRISS group, supported by GDR ASR (CNRS, France) (http://www.actriss.org/).

Julien Deantoni

Technical Program Committee: CIEL2012.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence: Julien Deantoni, Computer Environnement, 30 h, L2 level, Polytech engineering school of University Nice/Sophia-Antipolis (UNS EPU) France.

Master:Julien Deantoni, Model Driven Engineering, 22 h, M2, UNS EPU.

Master : Julien Deantoni, C++ and Object Oriented Programming, 54 h, M1, UNS EPU.

Master: Julien Deantoni, Embedded Software and systems, 7 h, M2, UNS EPU.

Master: Julien Deantoni, VHDL, 40 h, M1, UNS EPU.

Licence: Sid-Ahmed-Ali Touati, Assembleurs et jeux d'instructions, 52h, L3, UNS Licence info.

Licence: Sid-Ahmed-Ali Touati, Systèmes informatiques, 30h, L1, UNS Licence info.

Master: Sid-Ahmed-Ali Touati, Programmation efficaces pour programmes embarqués et hautes performances, 16h, M1 Master ISI.

Master: Sid-Ahmed-Ali Touati, Systèmes d'exploitation avancés, 39h, M1, UNS Master ISI.

Master: Sid-Ahmed-Ali Touati, Programmation efficace et Optimisation de code, 16h, M1, UNS Master ISI.

Master: Sid-Ahmed-Ali Touati, Architecture des Processeurs, 15h, M1, UNS EPU.

Licence : Frédéric Mallet, Introduction à la Programmation Objet, 45h, L1, UNS.

Licence: Frédéric Mallet, Architecture des ordinateurs, 45h, L3, UNS.

Master: Frédéric Mallet, Programmation Avancée et Design Patterns, 93h, M1, UNS.

Master: Frédéric Mallet, Java pour l'Informatique Industrielle, 24h, M1, UNS.

Master: Frédéric Mallet, Architectures des ordinateurs, 12h, M1, UNS.

Master: Frédéric Mallet, Formal Models for Network-On-Chips, 3h, M2, UNS.

Licence : Marie-Agnes Peraldi-Frati, Algorithms and programming 60h,L1, UNS Institute of technology.

Licence : Marie-Agnes Peraldi-Frati, System and Networks administration 80h, L2, UNS Institute of technology .

Licence :Marie-Agnes Peraldi-Frati, Web Programming 50 h , L2, UNS Institute of technology.

Master : Robert de Simone, Formal Models for Networks-on-Chip, 24h , M2, UNS.

Master : Robert de Simone, Semantics of Embedded and Distributed Systems, 24 h, M1, UNS.

Master: Yves Sorel, Optimization of distributed real-time embedded systems, 24H, M2, University Paris Sud.

Master: Yves Sorel, Distributed real-time systems, 26H, M2, University Paris Est

Master: Yves Sorel, Specification and formal models for embedded systems, 28H, M2, ENSTA Engineering School Paris

Master: Yves Sorel, Correct by construction design of reactive systems, 18H, M2, ESIEE Engineering School Noisy-Le-Grand

Master: Dumitru Potop, Programmation synchrone des systèmes temps-réel, 8h, M1, EPITA Engineering School Paris

Licence: Laurent George, Java and Shell programming 48h, L1, IUT RT UPEC

Master: Laurent George, Distributed Real-Time Systems, 24h, M2, UPEC

9.2.2. Supervision

PhD: Mohamed Marouf, *Ordonnancement temps réel dur multiprocesseur tolérant aux fautes appliqué à la robotique mobile*, Ecole des Mines de Paris, defended on June 1st 2012, supervised by Yves Sorel, co-supervised by Brigitte D'Andrea-Novel (Institut Mines-Telecom).

PhD in progress: Carlos Ernesto Gomez-Cardenas,*Environnement multi-vues pour la métamodélisation sémantique formelle de systèmes embarqués*, UNS, started September 2010, supervised by Frédéric Mallet.

PhD in progress: Matias Vara-Larsen, *Toward a formal and hierarchical timed model for concurrent heterogeneous model*, ANR/CNRS, started November 2012, supervised by Frédéric Mallet, co-supervised by Julien Deantoni.

PhD in progress: Ameni Khecharem, *High-Level modeling of hierarchical power management policies in SoCs*, UNS, started October 2012, supervised by Robert de Simone.

PhD in progress: Ying Lin, *Formal Analysis of polychronous models with MARTE/CCSL*, East China Normal University, started September 2011, supervised by Jing Liu (ECNU), co-supervised by Frédéric Mallet.

PhD in progress: Falou Ndoye, *Multiprocessor real-time scheduling taking into account preemption cost*, started January 2011, supervised by Yves Sorel.

PhD in progress: Manel Djemal, *Distributed real-time scheduling onto NoC architectures*, EDITE/UPMC, started Nov. 2010, co-supervised by Alix Munier (UPMC/Lip6) and D. Potop-Butucaru.

PhD in progress: Thomas Carle, *Real-time implementation of embedded control applications with conditional control onto time-triggered architectures*, EDITE/UPMC, started Sep. 2011, supervised by D. Potop-Butucaru.

PhD in progress: Pierre Courbin (ECE/UPEC), *Ordonnancement temps réel parallèle multiprocesseur*, started September 2009, supervised by Laurent George.

PhD in progress: Philippe Thierry (CIFRE Thales/UPEC), *Ordonnancement temps réel multiprocesseur sous contraintes de sécurité*, started April 2010, supervised by Laurent George.

PhD in progress: Vincent Sciandra, (CIFRE Veolia/UPEC), *Ordonnancement temps réel à criticité mixte dans les systèmes de transports intelligents (ITS)*, started Oct. 2010, supervised by Laurent George.

PhD in progress: Manar Qamhieh, (UPEMLV), *Ordonnancement temps réel de graphes de tâches parallèles à criticité mixte*, started Sept. 2010, co-supervised by Serge Midonnet (UPEMLV) and Laurent George.

PhD: Abdelhafid Mazouz, *An Empirical Study of Program Performance of OpenMP Applications on Multicore Platforms*, defended December 11 2012, UVSQ, supervised by Sid Touati (as part of his activities prior to joining Aoste), co-supervisd by D. Bartou (LABRI).

9.2.3. Juries

Robert de Simone

PhD reviewer: Sébastien Guillet (Université Bretagne-Sud), Cédric Auger (Université Orsay Paris-Sud),

HDR examiner: Ludovic Apvrille (Institut Mines-Telecom) (President of Jury).

Frédéric Mallet

PhD reviewer: Boutheina Bannour (Ecole Centrale Paris), Daniel Knorreck (Télécom ParisTech), Michel Lauer (INP Toulouse), Sabir Idrees (Télécom ParisTech)

PhD examiner: Cyril Faucher (Université de La Rochelle)

Sid Touati

PhD reviewer: Mohamed Benazouz (U. Paris 6), Jingyan Jourdan-Lu (ENS Lyon), Jean-Noël Quintin (INP Grenoble)

PhD examiner: Julien Jaegger (UVSQ)

Laurent George

PhD reviewer: Aurélien Monot (Université de Lorraine, Nancy)

Yves Sorel

PhD examiner: Mohamed Benazouz (University Pierre et Marie Curie)

Julien Deantoni

PhD examiner: Abderraouf Benyahia (Supelec)

10. Bibliography

Major publications by the team in recent years

[1] C. André. *Syntax and Semantics of the Clock Constraint Specification Language (CCSL)*, Inria, 05 2009, 37, RR-6925, http://hal.inria.fr/inria-00384077/en/.

[2] C. André, J. Deantoni, F. Mallet, R. de Simone. *The Time Model of Logical Clocks available in the OMG MARTE profile*, in "Synthesis of Embedded Software: Frameworks and Methodologies for Correctness by Construction", S. K. Shukla, J.-P. Talpin (editors), Springer Science+Business Media, LLC 2010, Jul 2010, p. 201–227, Chapter 7, http://hal.inria.fr/inria-00495664.

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

[4] 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.

[5] J. Boucaron, A. Coadou, R. de Simone. *Formal Modeling of Embedded Systems with Explicit Schedules and Routes*, in "Synthesis of Embedded Software: Frameworks and Methodologies for Correctness by Construction", S. K. Shukla, J.-P. Talpin (editors), Springer Science+Business Media, LLC 2010, Jul 2010, p. 41–78, Chapter 2, http://hal.inria.fr/inria-00495667.

[6] J. Boucaron, R. de Simone, J.-V. Millo. *Latency-insensitive design and central repetitive scheduling*, in "MEMOCODE", 2006, p. 175-183, http://dx.doi.org/10.1109/MEMCOD.2006.1695923.

[7] 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/publications/pubs/aor07/aor07.pdf.

[8] 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.

[9] 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", Mont Saint-Michel, France, June 2003, http://www-rocq.inria.fr/syndex/publications/pubs/memocode03/memocode03.pdf.

[10] P. MEUMEU YOMSI, Y. SOREL. *Extending Rate Monotonic Analysis with Exact Cost of Preemptions for Hard Real-Time Systems*, in "Proceedings of 19th Euromicro Conference on Real-Time Systems, ECRTS'07", Pisa, Italy, July 2007, http://www-rocq.inria.fr/syndex/publications/pubs/ecrts07/ecrts07.pdf.

[11] D. POTOP-BUTUCARU, BENOÎT. CAILLAUD. *Correct-by-Construction Asynchronous Implementation of Modular Synchronous Specifications*, in "Fundam. Inf.", January 2007, vol. 78, p. 131–159, http://portal.acm.org/citation.cfm?id=1366007.1366013.

[12] D. POTOP-BUTUCARU, R. DE SIMONE, Y. SOREL. *From Synchronous Specifications to Statically-Scheduled Hard Real-Time Implementations*, in "Synthesis of Embedded Software: Frameworks and Methodologies for Correctness by Construction", S. K. SHUKLA, J.-P. TALPIN (editors), Springer Science+Business Media, LLC 2010, Jul 2010, p. 229–262, Chapter 8, http://hal.inria.fr/inria-00495666.

[13] D. POTOP-BUTUCARU, S. EDWARDS, G. BERRY. *Compiling Esterel*, Springer, 2007.

[14] D. POTOP-BUTUCARU, R. DE SIMONE. *Optimizations for Faster Execution of Esterel Programs*, in "MEMOCODE'03", 2003.

[15] R. DE SIMONE, D. POTOP-BUTUCARU, JEAN-PIERRE. TALPIN. *The Synchronous Hypothesis and Synchronous Languages*, in "Embedded Systems Handbook", CRC Press, 2005, chap. 8.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[16] J.-F. LE TALLEC. *Extraction de modèles pour la conception de systèmes sur puce*, Université de Nice Sophia Antipolis, January 2012.

[17] M. MAROUF. *Ordonnancement temps réel dur multiprocesseur tolérant aux fautes appliqué à la robotique mobile*, Ecole des Mines de Paris, Spécialité Informatique temps réel, robotique et automatique, 2012, http://www-rocq.inria.fr/syndex/publications/pubs/theses/THMM.pdf.

Articles in International Peer-Reviewed Journals

[18] M. BACHIR, S. TOUATI, F. BRAULT, D. GREGG, A. COHEN. *Minimal Unroll Factor for Code Generation of Software Pipelining*, in "International Journal of Parallel Programming", 2012, p. 1-58, http://dx.doi.org/10.1007/s10766-012-0203-z.

[19] C. GLITIA, J. DEANTONI, F. MALLET, J.-V. MILLO, P. BOULET, A. GAMATIÉ. *Progressive and explicit refinement of scheduling for multidimensional data-flow applications using uml marte*, in "Design Automation for Embedded Systems", 2012, vol. 16, n^{o} 2, p. 137-169 [*DOI :* 10.1007/S10617-012-9093-Y], http://hal.inria.fr/hal-00727239.

[20] J.-V. MILLO, R. DE SIMONE. *Periodic scheduling of marked graphs using balanced binary words*, in "Theoretical Computer Science", November 2012, vol. 458, n^{o} 2, p. 113–130 [*DOI :* 10.1016/J.TCS.2012.08.012], http://hal.inria.fr/hal-00764076.

[21] S. TOUATI, J. WORMS, S. BRIAIS. *The Speedup-Test: a statistical methodology for program speedup analysis and computation*, in "Concurrency and Computation: Practice and Experience", 2012, http://dx.doi.org/10.1002/cpe.2939.

International Conferences with Proceedings

[22] J. Deantoni, F. Mallet. *TimeSquare: Treat your Models with Logical Time*, in "TOOLS - 50th International Conference on Objects, Models, Components, Patterns - 2012", Prague, Czech Republic, C. A. Furia, S. Nanz (editors), Lecture Notes in Computer Science - LNCS, Springer, May 2012, vol. 7304, p. 34-41 [*DOI :* 10.1007/978-3-642-30561-0_4], http://hal.inria.fr/hal-00688590.

[23] M. Djemal, F. Pêcheux, D. Potop-Butucaru, R. de Simone, F. Wajsbürt, Z. Zhang. *Programmable Routers for Efficient Mapping of Applications onto NoC-based MPSoCs*, in "DASIP International Conference on Design and Architecture for Signal and Image Processing", IEEE, October 2012.

[24] A. Goknil, M.-A. Peraldi-Frati. *A DSL for Specifying Timing Requirements*, in "MoDRE- 2sd International Model-Driven Requirements Engineering (MoDRE) Workshop", Chicago, United States, IEEE Digital Library, September 2012, http://hal.inria.fr/hal-00757168.

[25] C. Gomez, J. Deantoni, F. Mallet. *Multi-View Power Modeling based on UML, MARTE and SysML*, in "SEAA - 38th Euromicro Conference on Software Engineering and Advanced Applications", Cesme, Turkey, October 2012, p. 17-20 [*DOI :* 10.1109/SEAA.2012.66], http://hal.inria.fr/hal-00720735.

[26] S. Kuntz, M.-A. Peraldi-Frati, H. Blom, D. Karlsson. *Timing Modeling with AUTOSAR. Current State and Future Directions*, in "DATE 2012. Design Automation & Test in Europe", Dresden, Germany, March 2012, http://hal.inria.fr/hal-00667070.

[27] Z. Liu, J. Liu, J. He, F. Mallet, Z. Miaomiao. *Formal Specification of Hybrid MARTE Statecharts*, in "Sixth International Symposium on Theoretical Aspects of Software Engineering", Beijing, China, IEEE, July 2012 [*DOI :* 10.1109/TASE.2012.26], http://hal.inria.fr/hal-00764044.

[28] F. Mallet. *Automatic Generation of Observers from MARTE/CCSL*, in "International Symposium on Rapid System Prototyping", Tampere, Finland, IEEE, 2013, http://hal.inria.fr/hal-00764066.

[29] M. Marouf, L. George, Y. Sorel. *Schedulability analysis for a combination of non-preemptive strict periodic tasks and preemptive sporadic tasks*, in "ETFA'12 - 17th IEEE International Conference on Emerging Technologies and Factory Automation", Kraków, Poland, IEEE, September 2012, http://hal.inria.fr/hal-00737917.

[30] J.-V. Millo, R. de Simone. *Refining cellular automata with routing constraints*, in "Automata & JAC (Exploratory track)", E. Formenti (editor), September 2012, vol. 2.

[31] J.-V. Millo, S. Ramesh. *Relating Requirement and Design Variabilities*, in "Proceedings of the international workshop on Software Quality and Management (SQAM'12)", December 2012.

[32] S. Mohalik, S. Ramesh, J.-V. Millo, Krishna Shankara. Narayanan, Ganesh Khandu. Narwane. *Tracing SPLs precisely and efficiently*, in "Proceedings of the 16th International Software Product Line Conference - Volume 1", New York, NY, USA, SPLC '12, ACM, 2012, p. 186–195, http://doi.acm.org/10.1145/2362536.2362562.

[33] M.-A. Peraldi-Frati, A. Goknil, M. Adedjouma, P.-Y. Gueguen. *Modeling a BSG-E Automotive System with the Timing Augmented Description language.*, in "ISOLA 2012 -5th International Symposium On

Leveraging Applications of Formal Methods, Verification and Validation", Amirandes, Héraklion, Greece, TU Dortmund - EASST, October 2012, p. 111-125, http://hal.inria.fr/hal-00757185.

[34] M.-A. Peraldi-Frati, A. Goknil, J. Deantoni, J. Nordlander. *A timing model for specifying multi clock automotive systems. The Timing Augmented Description Language V2*, in "ICECCS 2012 : International Conference on Engineering of Complex Computer Systems", Paris, France, IEEE, July 2012, 10 pages, http://hal.inria.fr/hal-00687562.

[35] M.-A. Peraldi-Frati, D. Karlsson, A. Hamann, S. Kuntz, J. Nordlander. *The TIMMO-2-USE project: Time modeling and analysis to use*, in "ERTS2012 International Congres on Embedded Real Time Software and Systems", Toulouse, France, February 2012, http://hal.inria.fr/hal-00649781.

[36] M. Qamhieh, S. Midonnet, L. George. *A Parallelizing Algorithm for Real-Time Tasks of Directed Acyclic Graphs Model*, in "RTAS'12 : The 18th IEEE Real-Time and Embedded Technology and Applications Symposium. Work-In-Progress Session", Beijing, China, April 2012, p. 45-48, http://hal.inria.fr/hal-00695818.

National Conferences with Proceeding

[37] D. De Rauglaudre. *Vérification formelle de conditions d'ordonnancabilité de tâches temps réel périodiques strictes*, in "JFLA - Journées Francophones des Langages Applicatifs - 2012", Carnac, France, February 2012, http://hal.inria.fr/hal-00665929.

Scientific Books (or Scientific Book chapters)

[38] C. Glitia, J. Deantoni, F. Mallet. *Logical Time @ Work: Capturing Data Dependencies and Platform Constraints*, in "System Specification and Design Languages", Lecture Notes in Electrical Engineering, Springer New York, 2012, vol. 106, p. 223–238 [*DOI :* 10.1007/978-1-4614-1427-8_14], http://hal.inria.fr/hal-00651864.

Research Reports

[39] T. Carle, D. Potop-Butucaru, Y. Sorel, D. Lesens. *From dataflow specification to multiprocessor partitioned time-triggered real-time implementation*, Inria, October 2012, n^{o} RR-8109, http://hal.inria.fr/hal-00742908.

[40] J. Deantoni, F. Mallet. *ECL: the Event Constraint Language, an Extension of OCL with Events*, Inria, July 2012, n^{o} RR-8031, 24, http://hal.inria.fr/hal-00721169.

[41] C. Gomez, J. Deantoni, F. Mallet. *Multi-View Power Modeling based on UML MARTE and SysML*, Inria, April 2012, n^{o} RR-7934, 19, http://hal.inria.fr/hal-00688853.

[42] F. Mallet, L. Yin. *Correct Transformation from CCSL to Promela for verification*, Inria, January 2012, n^{o} RR-7491, 33, http://hal.inria.fr/hal-00667849.

[43] J.-V. Millo, R. de Simone. *Periodic scheduling of marked graphs using balanced binary words*, Inria, February 2012, n^{o} RR-7891, 33, http://hal.inria.fr/hal-00672606.

[44] J.-V. Millo, R. de Simone. *Refining cellular automata with routing constraints*, Inria, August 2012, n^{o} RR-8051, 15, http://hal.inria.fr/hal-00725878.

[45] J.-V. Millo, S. Ramesh, Krishna Shankara. Narayanan, Ganesh Khandu. Narwane. *Compositional Verification of Evolving SPL*, Inria, October 2012, n^{o} RR-8125, 34, http://hal.inria.fr/hal-00747533.

[46] L. Yin, J. Deantoni, F. Mallet, R. de Simone. *Schedulability analysis by exhaustive state space construction: translating CCSL to transition-based Generalized Buchi Automata*, Inria, October 2012, n^{o} RR-8102, 22, http://hal.inria.fr/hal-00743874.

References in notes

[47] 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.

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

[49] J. Carlier, P. Chrétienne. *Problèmes d'ordonnancement*, Masson, 1988.

[50] 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.

[51] A. Cohen, M. Duranton, C. Eisenbeis, C. Pagetti, F. Plateau, M. Pouzet. *N-Synchronous Kahn Networks: a Relaxed Model of Synchrony for Real-Time Systems*, in "ACM International Conference on Principles of Programming Languages (POPL'06)", Charleston, South Carolina, USA, January 2006.

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

[53] S. Edwards. *Languages for Digital Embedded Systems*, Kluwer, 2000.

[54] N. Halbwachs. *Synchronous Programming of Reactive Systems*, in "Computer Aided Verification", 1998, p. 1-16, http://www-verimag.imag.fr/~halbwach/newbook.pdf.

[55] 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.

[56] Edward A. Lee, D. G. Messerschmitt. *Static Scheduling of Synchronous Data Flow Programs for Digital Signal Processing*, in "IEEE Trans. Computers", 1987.

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