IN PARTNERSHIP WITH:
CNRS

Université Nice - Sophia Antipolis

Activity Report 2019

Project-Team Kairos

Multiform Logical Time for Formal Cyber-Physical System Design

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

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Embedded and Real-time Systems

Table of contents

Project-Team Kairos

Creation of the Project-Team: 2019 July 01

Keywords:

Computer Science and Digital Science:

A1.1.1. - Multicore, Manycore
A1.1.2. - Hardware accelerators (GPGPU, FPGA, etc.)
A1.2.5. - Internet of things
A1.2.7. - Cyber-physical systems
A1.5.2. - Communicating systems
A2.2. - Compilation
A2.3. - Embedded and cyber-physical systems
A2.4. - Formal method for verification, reliability, certification
A2.5.1. - Software Architecture & Design

Other Research Topics and Application Domains:

B5.1. - Factory of the future
B5.4. - Microelectronics
B6.1. - Software industry
B6.4. - Internet of things
B6.6. - Embedded systems
B6.7. - Computer Industry (harware, equipments...)
B7.2. - Smart travel
B8.1. - Smart building/home
B8.2. - Connected city
B9.5.1. - Computer science

1. Team, Visitors, External Collaborators

Research Scientists

Robert de Simone [Team leader, Inria, Senior Researcher, HDR]
Luigi Liquori [Inria, Senior Researcher, HDR]
Eric Madelaine [Inria, Researcher, HDR]
Dumitru Potop Butucaru [Inria, Researcher, from Jul 2019, HDR]

Faculty Members

Julien Deantoni [Univ de Nice - Sophia Antipolis, Associate Professor, HDR]
Frédéric Mallet [Univ de Nice - Sophia Antipolis, Professor, HDR]
Marie-Agnès Peraldi Frati [Univ de Nice - Sophia Antipolis, Associate Professor]
Sid Touati [Univ de Nice - Sophia Antipolis, Professor, HDR]

Post-Doctoral Fellows

Stéphanie Challita [Inria, Post-Doctoral Fellow, from Apr 2019]
Jad Khatib [Inria, Post-Doctoral Fellow, from Oct 2019]

PhD Students

Joëlle Abou Faysal [Renault, PhD Student, from Mar 2019, granted by CIFRE]
Carsten Bruns [Univ Côte d'Azur, PhD Student]

Giovanni Liboni [Safran, PhD Student, granted by CIFRE]
Hugo Pompougnac [Inria, PhD Student, from Oct 2019]
Claude Stolze [Univ de Nice - Sophia Antipolis, PhD Student]
Hui Zhao [Inria, PhD Student, until Jan 2019]

Technical staff

Paul Bouche [IRT Saint Exupéry, Engineer]
Luc Hogie [CNRS, Engineer]
Amin Oueslati [IRT Saint Exupéry, Engineer, until Aug 2019]

Interns and Apprentices

Fei Gao [Inria, from Mar 2019 until Aug 2019]
Cristian Grigoriu [Inria, from Mar 2019 until Aug 2019]
Zechen Hou [Inria, until Jan 2019]

Administrative Assistant

Patricia Riveill [Inria]

Visiting Scientists

Xiaohong Chen [East China Normal University (Shanghai), from Nov 2019]
Grygoriy Zholtkevych [V.N. Karazin Kharkiv National University (Ukrain), from Oct 2019 until Nov 2019]
Matteo Sereno [University of Turin (It), Professor, Juin 2019]
Thomas Ehrhard [CNRS, University of Paris, Research Director, Sept 2019]

2. Overall Objectives

2.1. Overall Objectives

The Kairos proposal ambitions to deal with the Design of Cyber-Physical Systems (CPS), at various stages, using Model-Based techniques and Formal Methods. Design here stands for co-modeling, co-simulation, formal verification and analysis activities, with connections both ways from models to code (synthesis and instrumentation for optimization). Formal analysis, in turn, concerns both functional and extra-functional correctness properties. Our goal is to link these design stages together, both vertically along the development cycle, and horizontally by considering the interactions between cyber/digital and physical models. These physical aspects comprise both physical environments and physical execution platform representations, which may become rather heterogeneous as in the cases of the Internet of Things (IoT) and computing at the edges of the gateways. The global resulting methodology can be tagged as Model-Based, Platform-Based CPS Design (Fig.1).

CPS design must take into account all 3 aspects of application requirements, execution platform guarantees and contextual physical environment to establish both functional and temporal correctness. The general objective of Kairos is thus to contribute in the definition of a corresponding design methodology, based on formal Models of Computation for joint modeling of cyber and physical aspects, and using the important central concept of Logical Time for expressing the requirements and guarantees that define CPS constraints.

Logical Multiform Time. It may be useful to provide an introduction and motivation for the notion of Logical Multiform Time (and Logical Clocks), as they play a central role in our approach to Design. We call Logical Clock any repetitive sequence of occurrences of an event (disregarding possible values carried by the event). It can be regularly linked to physical time (periodic), but not necessarily so: fancy processors may change speeds, simulation engine change time-integration steps, or much more generally one may react with event-driven triggers of complex logical nature (do this after 3-times that unless this...). It is our belief that user specifications are generally expressed using such notions, with only partial timing correlations between distinct logical clocks, so that the process of realization (or “model-based compilation”) consists for part in establishing (by analysis or abstract simulation) the possible tighter relations between those clocks (unifying them from a partial order of local total orders to a global total order). We have defined in the past a small

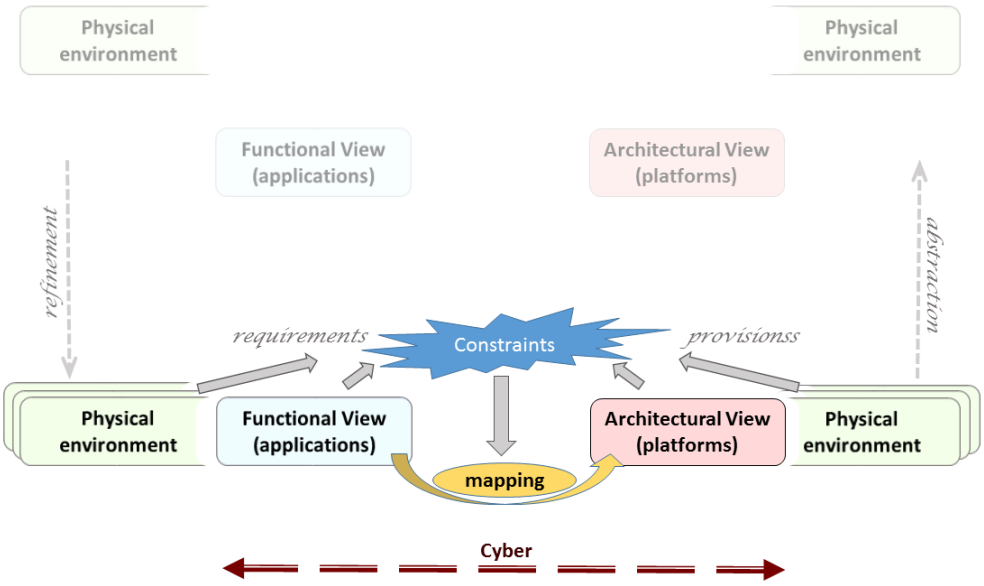

Figure 1. Cyber-Physical generic architectural features

language of primitives expressing recognized constraints structuring the relations between distinct logical clocks. This language (named CCSL for Clock Constraint Specification Language), borrows from notions of Synchronous Reactive Languages, Real-Time Scheduling Theory, and Concurrent Models of Computations and Communication (MoCCs) in Concurrency Theory altogether. Corresponding extensions of Timed Models originally based on single (discrete or continuous) time can also be considered. Logical Time is used in our approach to express relation constraints between heterogeneous models, of cyber or physical origin, and to support analysis and co-simulation. Addressing cyber-physical systems demands to revisit logical time to deal with the multi-physical and sometimes uncertain environments.

In the following sections we describe in turn the research agenda of Kairos on co-modeling, co-simulation, co-analysis and verification, and relation from models to code, respectively.

3. Research Program

3.1. Cyber-Physical co-modeling

Cyber-Physical System modeling requires joint representation of digital/cyber controllers and natural physics environments. Heterogeneous modeling must then be articulated to support accurate (co-)simulation, (co-)analysis, and (co-)verification. The picture above sketches the overall design framework. It comprises functional requirements, to be met provided surrounding platform guarantees, in a contract approach. All relevant aspects are modeled with proper Domain Specific Languages (DSL), so that constraints can be gathered globally, then analyzed to build a mapping proposal with both a structural aspect (functions allocated to platform resources), but also a behavioral ones, scheduling activities. Mapping may be computed automatically or not, provably correct or not, obtained by static analytic methods or abstract execution. Physical phenomena (in a very broad acceptance of the term) are usually modeled using continuous-time

models and differential equations. Then the "proper" discretization opportunities for numerical simulation form a large spectrum of mathematical engineering practices. This is not at all the domain of expertise of Kairos members, but it should not be a limitation as long as one can assume a number of properties from the discretized version. On the other hand, we do have a strong expertise on modeling of both embedded processing architectures and embedded software (i.e., the kind of usually concurrent, sometimes distributed software that reacts to and control the physical environment). This is important as, unlike in the "physical" areas where modeling is common-place, modeling of software and programs is far from mainstream in the Software Engineering community. These domains are also an area of computer science where modeling, and even formal modeling, of the real objects that are originally of discrete/cyber nature, takes some importance with formal Models of Computation and Communications. It seems therefore quite natural to combine physical and cyber modeling in a more global design approach (even multi-physic domains and systems of systems possibly, but always with software-intensive aspects involved). Our objective is certainly not to become experts in physical modeling and/or simulation process, but to retain from it only the essential and important aspects to include them into System-Level Engineering design, based on Model-Driven approaches allowing formal analysis.

This sets an original research agenda: Model-Based System Engineering environments exist, at various stages of maturity and specificity, in the academic and industrial worlds. Formal Methods and Verification/Certification techniques also exist, but generally in a point-wise fashion. Our approach aims at raising the level of formality describing relevant features of existing individual models, so that formal methods can have a greater general impact on usual, "industrial-level", modeling practices. Meanwhile, the relevance of formal methods is enhanced as it now covers various aspects in a uniform setting (timeliness, energy budget, dependability, safety/security...).

New research directions on formal CPS design should focus on the introduction of uncertainty (stochastic models) in our particular framework, on relations between (logical) real-time and security, on relations between common programming languages paradigms and logical time, on extending logical frameworks with logical time, on the concern with resource discovery also in presence of mobility inherent to connected objects and Internet of Things.

3.2. Cyber-Physical co-simulation

The FMI standard (Functional Mock-Up Interface) has been proposed for "purely physical" (i.e., based on persistent signals) co-simulation, and then adopted in over 100 industrial tools including frameworks such as Matlab/Simulink and Ansys, to mention two famous model editors. With the recent use of co-simulation to cyber-physical systems, dealing with the discrete and transient nature of cyber systems became mandatory. Together with other people from the community, we shown that FMI and other frameworks for co-simulation badly support co-simulation of cyber-physical systems; leading to bad accuracy and performances. More precisely, the way to interact with the different parts of the co-simulation require a specific knowledge about its internal semantics and the kind of data exposed (e.g., continuous, piecewise-constant). Towards a better co-simulation of cyber-physical systems, we are looking for conservative abstractions of the parts and formalisms that aim to describe the functional and temporal constraints that are required to bind several simulation models together.

3.3. Formal analysis and verification

Because the nature of our constraints is specific, we want to adjust verification methods to the goals and expressiveness of our modeling approach. Quantitative (interval) timing conditions on physical models combined with (discrete) cyber modes suggest the use of SMT (Satisfiability Modulo Theories) automatic solvers, but the natural expressiveness requested (as for instance in our CCSL constructs) shows this is not always feasible. Either interactive proofs, or suboptimal solutions (essentially resulting of abstract run-time simulations) should be considered. Complementarily to these approaches, we are experimenting with new variants of symbolic behavioural semantics, allowing to construct finite representations of the behaviour of CPS systems with explicit handling of data, time, or other non-functional aspects.

3.4. Relation between Model and Code

While models considered in Kairos can also be considered as executable specifications (through abstract simulation schemes), they can also lead to code synthesis and deployment. Conversely, code execution of smaller, elementary software components can lead to performance estimation enriching the models before global mapping optimization. CPS introduce new challenging problems for code performance stability. Indeed, two additional factors for performance variability appear, which were not present in classical embedded systems: 1) variable and continuous data input from the physical world and 2) variable underlying hardware platform. For the first factor, CPS software must be analysed in conjunction with its data input coming from the physics, so the variability of the performance may come from the various data. For the second factor, the underlying hardware of the CPS may change during the time (new computing actors appear or disappear, some actors can be reconfigured during execution). The new challenge is to understand how these factors influence performance variability exactly, and how to provide solutions to reduce it or to model it. The modeling of performance variability becomes a new input.

3.5. Code generation and optimization

A significant part CPS design happens at model level, through activities such as model construction, analysis, or verification. However, in most cases the objective of the design process is implementation. We mostly consider the implementation problem in the context of embedded, real-time, or edge computing applications, which are subject to stringent performance, embedding, and safety *non-functional requirements*.

The implementation of such systems usually involves a mix of synthesis—(real-time) scheduling, code generation, compilation—and performance (*e.g.* timing) analysis. One key difficulty here is that synthesis and performance analysis depend on each other. As enumerating the various solutions is not possible for complexity reasons, heuristic implementation methods are needed in all cases. One popular solution here is to build the system first using unsafe performance estimations for its components, and then check system *schedulability* through a global analysis. Another solution is to use safe, over-approximated performance estimations and perform their mapping in a way that ensures by construction the schedulability of the system.

In both cases, the specification of the design space—functional specification, execution platform model, non-functional requirements, implementation model—is a key problem. Another problem is the definition of scalable and efficient mapping methods based on both "exact" approaches (ILP/SMT/CP solving) and compilation-like heuristics.

3.6. Extending logical frameworks with logical time

The Curry-Howard isomorphism (*proposition-as-types and proofs-as-typed-λ-terms*) represent the logical and computational basis to interactive theorem provers: our challenge is to investigate and design time constraints within a Dependent Type Theory (e.g. if event A happened-before event B, then the timestamp/type of A is less (i.e. a subtype) than the timestamp/type of B). We are currently extending the Edinburgh Logical Framework (LF) of Harper-Honsell-Plotkin with relevant constructs expressing logical time and synchronization between processes. Also, union and intersection types with their subtyping constraints theories could capture some constraints expressions *à la* CCSL needed to formalize logical clocks (in particular CCSL expressions like subclock, clock union, intersection and concatenation) and provide opportunities for an *ad hoc* polymorphic timed Type Theory. Logical time constraints seen as property types can beneficially be handled by logical frameworks. The new challenge here is to demonstrate the relevance of Type Theory to work on logical and multiform timing constraint resolution.

3.7. Object-oriented programming and logical time

We formalize in the past object-oriented programming features and safe static type systems featuring delegation-based or trait inheritance: well typed program will never produce into the `message-not-found` infamous run-time error. We view the logical time as a means to enhance the description of timing constraints

and properties on top of existing language semantics. When considering general purpose object-oriented languages, like Java, Type Theory is a natural way to provide such properties. Currently, few languages have special types to manage instants, time structures and instant relations like subclocking, precedence, causality, equality, coincidence, exclusion, independence, etc. CCSL provides ad-hoc constructors to specify clock constraints and logical time: enriching object-oriented type theories with CCSL expressions could constitute an interesting research perspective towards a wider usage of CCSL. The new challenge is to consider logical time constraints as behavioral type properties, and the design of programming language constructs and *ad-hoc* type systems. Advances of typed-calculi featuring those static time features will be applied to our extension [42] of the lambda-calculus of objects of Fisher-Honsell-Mitchell.

3.8. Extensions for spatio-temporal modeling and mobile systems

While Time is clearly a primary ingredient in the proper design of CPS systems, in some cases Space, and related notions of local proximity or conversely long distance, play also a key role for correct modeling, often in part because of the constraints this puts on interactions and time for communications. Once space is taken into account, one has to recognize also that many systems will request to consider mobility, originated as change of location through time. Mobile CPS (or mCPS) systems occur casually, e.g., in the case of Intelligent Transportation Systems, or in roaming connected objects of the IoT. Spatio-temporal and mobility modeling may each lead to dynamicity in the representation of constraints, with the creation/deletion/discovering of new components in the system. This opportunity for new expressiveness will certainly cause new needs in handling constraint systems and topological graph locations. The new challenge is to provide an algebraic support with a constraint description language that could be as simple and expressive as possible, and of use in the semantic annotations for mobile CPS design. We also aims to provide fully distributed routing protocols to manage Semantic Resource Discovery in IoT.

4. Application Domains

4.1. Cyber-Physical and Embedded Systems

We have historical contacts with industrial and academic partners in the domains of avionics and embedded electronics (Airbus, Thales, Safran). We have new collaborations in the fields of satellites (Thales Alenia Space) and connected cars (Renault Software Labs). These provide for use case and new issues in CPS co-modeling and co-design (Digital Twins) further described in New Results section.

4.2. Connected Objects in the Internet Of Things

Due to increasing collaborations with local partners, we have recently considered Smart Contracts (as popularized in Blockchain frameworks), as a way to formally established specification of behavioral system traces, applied to connected objects in a IoT environment. The new ANR project SIM is based on this.

5. Highlights of the Year

5.1. Highlights of the Year

- New Collaboration with Renault Software Labs - CIFRE starting in April 2019
- ANR Project on the verification of smart contracts on the use of multi-modalities transportations in the smart city - AAPG 2019 PRCE

5.1.1. Awards

Frederic Mallet is Laureate of the program ‘Jeune Talent France Chine 2019’ from French Embassy in China.

6. New Software and Platforms

6.1. VerCors

VERification of models for distributed communicating COmponants, with safety and Security

KEYWORDS: Software Verification - Specification language - Model Checking

FUNCTIONAL DESCRIPTION: The VerCors tools include front-ends for specifying the architecture and behaviour of components in the form of UML diagrams. We translate these high-level specifications, into behavioural models in various formats, and we also transform these models using abstractions. In a final step, abstract models are translated into the input format for various verification toolsets. Currently we mainly use the various analysis modules of the CADP toolset.

RELEASE FUNCTIONAL DESCRIPTION: It includes integrated graphical editors for GCM component architecture descriptions, UML classes, interfaces, and state-machines. The user diagrams can be checked using the recently published validation rules from, then the corresponding GCM components can be executed using an automatic generation of the application ADL, and skeletons of Java files.

The experimental version (2019) also includes algorithms for computing the symbolic semantics of Open Systems, using symbolic methods based on the Z3 SMT engine.

NEWS OF THE YEAR: The experimental version (2019) also includes: - algorithms for computing the symbolic semantics of Open Systems, using symbolic methods based on the Z3 SMT engine. - a stand alone textual editor for (open) pNet systems, that generates API code to construct their internal representation in the platform.

- Participants: Antonio Cansado, Bartlomiej Szejna, Eric Madelaine, Ludovic Henrio, Marcela Rivera, Nassim Jibai, Oleksandra Kulankhina, Siqi Li, Xudong Qin and Zechen Hou
- Partner: East China Normal University Shanghai (ECNU)
- Contact: Eric Madelaine
- URL: https://team.inria.fr/scale/software/vercors/

6.2. TimeSquare

KEYWORDS: Profil MARTE - Embedded systems - UML - IDM

SCIENTIFIC DESCRIPTION: TimeSquare offers six main functionalities:

* graphical and/or textual interactive specification of logical clocks and relative constraints between them,

* definition and handling of user-defined clock constraint libraries,

* automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction,

* call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...).

* compilation to pure java code to enable embedding in non eclipse applications or to be integrated as a time and concurrency solver within an existing tool.

* a generation of the whole state space of a specification (if finite of course) in order to enable model checking of temporal properties on it

FUNCTIONAL DESCRIPTION: 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, and more accurately on the associated Clock Constraint Specification Language (CCSL) for the expression of timing constraints.

- Participants: Benoît Ferrero, Charles André, Frédéric Mallet, Julien DeAntoni and Nicolas Chleq
- Contact: Julien DeAntoni
- URL: http://timesquare.inria.fr

6.3. GEMOC Studio

KEYWORDS: DSL - Language workbench - Model debugging

SCIENTIFIC DESCRIPTION: The language workbench put together the following tools seamlessly integrated to the Eclipse Modeling Framework (EMF):

- Melange, a tool-supported meta-language to modularly define executable modeling languages with execution functions and data, and to extend (EMF-based) existing modeling languages.
- MoCCML, a tool-supported meta-language dedicated to the specification of a Model of Concurrency and Communication (MoCC) and its mapping to a specific abstract syntax and associated execution functions of a modeling language.
- GEL, a tool-supported meta-language dedicated to the specification of the protocol between the execution functions and the MoCC to support the feedback of the data as well as the callback of other expected execution functions.
- BCOoL, a tool-supported meta-language dedicated to the specification of language coordination patterns to automatically coordinates the execution of, possibly heterogeneous, models.
- Sirius Animator, an extension to the model editor designer Sirius to create graphical animators for executable modeling languages.

FUNCTIONAL DESCRIPTION: The GEMOC Studio is an eclipse package that contains components supporting the GEMOC methodology for building and composing executable Domain-Specific Modeling Languages (DSMLs). It includes the two workbenches: The GEMOC Language Workbench: intended to be used by language designers (aka domain experts), it allows to build and compose new executable DSMLs. The GEMOC Modeling Workbench: intended to be used by domain designersto create, execute and coordinate models conforming to executable DSMLs. The different concerns of a DSML, as defined with the tools of the language workbench, are automatically deployed into the modeling workbench. They parametrize a generic execution framework that provide various generic services such as graphical animation, debugging tools, trace and event managers, timeline, etc.

- Participants: Didier Vojtisek, Dorian Leroy, Erwan Bousse, Fabien Coulon and Julien DeAntoni
- Partners: IRIT - ENSTA - I3S - OBEO - Thales TRT
- Contact: Benoît Combemale
- URL: http://gemoc.org/studio.html

6.4. BCOol

BCOoL

KEYWORDS: DSL - Language workbench - Behavior modeling - Model debugging - Model animation

FUNCTIONAL DESCRIPTION: BCOoL is a tool-supported meta-language dedicated to the specification of language coordination patterns to automatically coordinates the execution of, possibly heterogeneous, models.

- Participants: Julien DeAntoni, Matias Vara Larsen, Benoît Combemale and Didier Vojtisek
- Contact: Julien DeAntoni
- URL: http://www.gemoc.org

6.5. myMed

A geo-localised Framework for building Publish-Subscribe applications in a fixed and mobile environment

KEYWORDS: Framework - Peer-to-peer. - NoSQL - Mobile application - Social network - Publish-subscribe - Iot - Peer-to-peer

SCIENTIFIC DESCRIPTION: myMed : an ad-hoc framework to design, develop, host, and execute Publish-Subscribe based fully distributed applications running in a static or mobile network. Application examples can be found in Online Social Networks or in Resource Discovery for the IoT. In a nutshell myMed is composed by:

• A myMed Software Development Kit (SDK) to develop fixed and mobile web applications, but also to build native applications on Smartphones equipped with Android or iOS. Every module can be freely used without interfering with other applications, in a true Lego fashion.

• A myMed cloud to execute the mobile applications: the cloud is composed of a backbone of 50PCs, distributed through the "AlpMed" EuroRegion and following some precise network criteria (4G, optical Fiber, ..). The operating system running on those PC is a customised and partitioned version of Ubuntu to allow to use the PC as a myMed server as well as a ordinary desktops. As in Peer-to-Peer technology, we do not require that all machines belonging to the backbone are constantly running.

• A myMed backbone, based on a well-tested noSQL database, Cassandra, which can accommodate any number of users without any code changes. Machines can be classically concentrated on a data-center or – more interestingly – fully decentralized (modulo a decent internet connection). Failures of one or many machines do not affect the running of the system, thanks to replication of the data on several servers. A little collection of proof of concept applications to validate, experiment, and testing the development kit and the execution cloud have been implemented.

FUNCTIONAL DESCRIPTION: myMed is an experimental framework for implementing, hosting and deploying, on the top of a built-in cloud platform, many applications using intensively the Publish-Subscribe (PUB/SUB) paradigm, like e.g. Open Social Networks or Resource Discovery in a distributed data-base. Those applications could take advantage of sharing common software modules, hardware resources, making inter-communication and inter-interaction simpler and improving rapid development and deployement.

- Participants: Luigi Liquori, Claudio Casetti, Mariangiola Dezani and Mino Anglano
- Partners: Politecnico di Torino - Université de Nice Sophia Antipolis (UNS) - Università di Torino - Università del Piemonte Orientale
- Contact: Luigi Liquori
- URL: http://www.mymed.fr

6.6. JMaxGraph

KEYWORDS: Java - HPC - Graph algorithmics

FUNCTIONAL DESCRIPTION: JMaxGraph is a collection of techniques for the computation of large graphs on one single computer. The motivation for such a centralized computing platform originates in the constantly increasing efficiency of computers which now come with hundred gigabytes of RAM, tens of cores and fast drives. JMaxGraph implements a compact adjacency-table for the representation of the graph in memory. This data structure is designed to 1) be fed page by page, à-la GraphChi, 2) enable fast iteration, avoiding memory jumps as much as possible in order to benefit from hardware caches, 3) be tackled in parallel by multiple-threads. Also, JMaxGraph comes with a flexible and resilient batch-oriented middleware, which is suited to executing long computations on shared clusters. The first use-case of JMaxGraph allowed F. Giroire, T. Trolliet and S. Pérennes to count K2,2s, and various types of directed triangles in the Twitter graph of users (23G arcs, 400M vertices). The computation campaign took 4 days, using up to 400 cores in the NEF Inria cluster.

- Contact: Luc Hogie
- URL: http://www.i3s.unice.fr/~hogie/software/?name=jmaxgraph

6.7. Lopht

Logical to Physical Time Compiler

KEYWORDS: Real time - Compilation

SCIENTIFIC DESCRIPTION: The Lopht (Logical to Physical Time Compiler) has been designed as an implementation of the AAA methodology. Like SynDEx, Lopht relies on off-line allocation and scheduling techniques to allow real-time implementation of dataflow synchronous specifications onto multiprocessor systems. But there are several originality points: a stronger focus on efficiency, which results in the use of a compilation-like approach, a focus on novel target architectures (many-core chips and time-triggered embedded systems), and the possibility to handle multiple, complex non-functional requirements covering real-time (release dates and deadlines possibly different from period, major time frame, end-to-end flow constraints), ARINC 653 partitioning, the possibility to preempt or not each task, and finally SynDEx-like allocation.

FUNCTIONAL DESCRIPTION: Compilation of high-level embedded systems specifications into executable code for IMA/ARINC 653 avionics platforms. It ensures the functional and non-functional correctness of the generated code.

- Participants: Dumitru Potop-Butucaru, Manel Djemal, Thomas Carle and Zhen Zhang
- Contact: Dumitru Potop-Butucaru

6.8. LoPhT-manycore

Logical to Physical Time compiler for many cores

KEYWORDS: Real time - Compilation - Task scheduling - Automatic parallelization

SCIENTIFIC DESCRIPTION: Lopht is a system-level compiler for embedded systems, whose objective is to fully automate the implementation process for certain classes of embedded systems. Like in a classical compiler (e.g. gcc), its input is formed of two objects. The first is a program providing a platform-indepedent description of the functionality to implement and of the non-functional requirements it must satisfy (e.g. real-time, partitioning). This is provided under the form of a data-flow synchronous program annotated with non-functional requirements. The second is a description of the implementation platform, defining the topology of the platform, the capacity of its elements, and possibly platform-dependent requirements (e.g. allocation).

From these inputs, Lopht produces all the C code and configuration information needed to allow compilation and execution on the physical target platform. Implementations are correct by construction Resulting implementations are functionally correct and satisfy the non-functional requirements. Lopht-manycore is a version of Lopht targeting shared-memory many-core architectures.

The algorithmic core of Lopht-manycore is formed of timing analysis, allocation, scheduling, and code generation heuristics which rely on four fundamental choices. 1) A static (off-line) real-time scheduling approach where allocation and scheduling are represented using time tables (also known as scheduling or reservation tables). 2) Scalability, attained through the use of low-complexity heuristics for all synthesis and associated analysis steps. 3) Efficiency (of generated implementations) is attained through the use of precise representations of both functionality and the platform, which allow for fine-grain allocation of resources such as CPU, memory, and communication devices such as network-on-chip multiplexers. 4) Full automation, including that of the timing analysis phase.

The last point is characteristic to Lopht-manycore. Existing methods for schedulability analysis and real-time software synthesis assume the existence of a high-level timing characterization that hides much of the hardware complexity. For instance, a common hypothesis is that synchronization and interference costs are accounted for in the duration of computations. However, the high-level timing characterization is seldom (if ever) soundly derived from the properties of the platform and the program. In practice, large margins (e.g. 100%) with little formal justification are added to computation durations to account for hidden hardware complexity. Lopht-manycore overcomes this limitation. Starting from the worst-case execution time (WCET) estimations of computation operations and from a precise and safe timing model of the platform, it maintains a precise timing accounting throughout the mapping process. To do this, timing accounting must take into account all details of allocation, scheduling, and code generation, which in turn must satisfy specific hypotheses.

FUNCTIONAL DESCRIPTION: Accepted input languages for functional specifications include dialects of Lustre such as Heptagon and Scade v4. To ensure the respect of real-time requirements, Lopht-manycore pilots the use of the worst-case execution time (WCET) analysis tool (ait from AbsInt). By doing this, and by using a precise timing model for the platform, Lopht-manycore eliminates the need to adjust the WCET values through the addition of margins to the WCET values that are usually both large and without formal safety guarantees. The output of Lopht-manycore is formed of all the multi-threaded C code and configuration information needed to allow compilation, linking/loading, and real-time execution on the target platform.

NEWS OF THE YEAR: In the framework of the ITEA3 ASSUME project we have extended the Lopht-manycore to allow multiple cores to access the same memory bank at the same time. To do this, the timing accounting of Lopht has been extended to take into account memory access interferences during the allocation and scheduling process. Lopht now also pilots the aiT static WCET analysis tool from AbsInt by generating the analysis scripts, thus ensuring the consistency between the hypotheses made by Lopht and the way timing analysis is performed by aiT. As a result, we are now able to synthesize code for the computing clusters of the Kalray MPPA256 platform. Lopht-manycore is evaluated on avionics case studies in the perspective of increasing its technology readiness level for this application class.

- Participants: Dumitru Potop-Butucaru and Keryan Didier
- Contact: Dumitru Potop-Butucaru

7. New Results

7.1. Spatio-temporal constraints for mobile systems, with automotive driving assistance illustrations

Participants: Frédéric Mallet, Joëlle Abou Faysal, Robert de Simone, Xiaohong Chen.

The objective here is to extend constraint specifications to encompass spatial aspects in addition to logical multiform time. Spatio-temporal logics and requirement formalisms are thus an inspiration here. But mobility requests additionally that these spatio-temporal relations evolve in time. We are investigating in several directions:

- a target methodological approach is to consider these spatio-temporal relations to express safe driving rules as requirements or guarantees, meant to (in)validate trajectory proposals computed by a lower-level algorithmic system (itself operating on more direct neighborhood information). A realistic size case study is handled in collaboration with Renault Software Labs, as part of the CIFRE PhD contract of Joëlle Abou-Faysal, to define the precise needs in expressiveness and formal validation.
- Preliminary definitions of a spatio-temporal requirement specification languages, borrowing ideas from spatio-temporal logics and formal mobile process modeling (none of which being sufficient to our aim), is being progressed in collaboration with fellow researchers from ECNU Shanghai [20].

7.2. System Engineering for Performance and Availability in satellite embedded COTS

Participants: Robert de Simone, Julien Deantoni, Amin Oueslati, Paul Bouche.

In the context of the IRT ATIPPIC project, which provided engineer position funding for Paul Bouche and Amin Oueslati, we investigated the application of a realistic formal design methodology applied on a real case study under construction by the ATIPPIC partners, in this case a prototype satellite based on general-purpose electronic Components-on-the-Shelf (COTS), not radiation-hardened. We focused on the one hand on the Model-Based Design of local interconnects, to provide analysis techniques regarding bandwidth and possible congestion of inter-process communications; on the other hand, we considered formal analysis of availability in case of fault (solar radiations), to study impact of alternative mitigation techniques for fault tolerance. Results were delivered in the form of Capella viewpoints and analysis tools to the IRT Saint-Exupéry, as free software. They were also published in [18], [23].

7.3. Efficient solvers and provers for CCSL

Participants: Frédéric Mallet, Xiaohong Chen.

One of the goal of the team is to promote the use of logical time in various application domains. This requires to have efficient solvers for CCSL. We have made considerable progresses on this part along two lines. One by relying on SMT solvers (like Z3), the other by building a dynamic logic amenable to building semi-automatic proofs for logical time properties of reactive systems. Then for some classes of problems we can efficient solving tools.

- The first step is to have an efficient Z3 library for solving CCSL specifications. We have improved a lot the performances over last year by getting rid of some of the existential quantifiers in our properties [35].
- Second, we use this solver to help requirement engineers elicite the requirements. We use execution traces to help generate valid satisfied CCSL specifications [28].
- Third, we have built a dynamic logics based on CCSL, where the formulae are derived from CCSL relational operators and programs include some of CCSL expressions and some imperative reactive constructs akin to Esterel programs. Then we have an interactive proof system, that helps prove that some reactive program satisfies a set of formulas at all time. As we use only a subset of CCSL then, we can restrict to a decidable subset of the logics and the SMT solver is always efficient. The SMT helps guide the semi-automatic proof [34] by identifying the next proof rules that can be used (or not).

7.4. Formal temporal Smart Contracts

Participants: Frédéric Mallet, Marie-Agnès Peraldi Frati, Robert de Simone.

"Smart Contracts", as a way to define legal ledger evolution in Blockchain environments, can be seen as rule constraints to be satisfied by the set of their participants. Such contracts are often reflecting requirements or guarantees extracted from a legal or financial corpus of rules, while this can be carried to other technical fields of expertise. Our view is that Smart Contracts are often relying on logically timed events, thus welcoming the specification style of our formalisms (such as CCSL). The specialization of multiform logical time constraints to this domain is under study, in collaboration with local academic partners at UCA UMR LEAT and Gredeg, and industrial partners, such as Symag and Renault Software Labs. Local funding was obtained from UCA DS4H EUR Academy 1, which allowed preparation of the ANR project SIM that was accepted in 2019. One goal is to get acceptance from the lawyers while still preserving strong semantics for verification. This builds on our previous expertise [16].

7.5. CCSL extension to Stochastic logical time

Participants: Frédéric Mallet, Robert de Simone.

CCSL specifications allows distinct clocks with unfixed inter-relations. In settings such as cyber-physical modeling, probabilistic rates of relative occurrences may be provided as bounds. The objective is to provide construct to introduce such relations for the inclusion and precedence partial orders, but also to consider also constructs that associate them. Preliminary results have been obtained by Frédéric Mallet in collaboration with fellow researchers from ECNU Shanghai.

7.6. Semantic Resource Discovery in Internet

Participant: Luigi Liquori.

Results [30] are obtained in close collaboration with professors Matteo Sereno and Rossano Gaeta from the University of Turin. Internet in recent years has become a huge set of channels for content distribution highlighting limits and inefficiencies of the current protocol suite originally designed for host-to-host communication. We propose a Content Name System Service (CNS) that provides a new network aware Content Discovery Service. The CNS behavior and architecture uses the BGP inter-domain routing information. In particular, the service registers and discovers resource names in each Internet Autonomous System (AS): contents are discovered by searching through the augmented AS graph representation classifying ASes into customer, provider, and peering, as the BGP protocol does. An interesting extension of this Internet Service could be to scale up to Internet of Things and to Cyber Physical Systems inter-networked with networks different than Internet.

7.7. Raising Semantic Resource Discovery in IoT

Participants: Luigi Liquori, Marie-Agnès Peraldi Frati.

Within the standards for M2M and the Internet of Things, managed by ETSI, oneM2M, we are looking for suitable mechanisms and protocols to perform a Semantic Resource Discovery as described in the previous Subsection. More precisely, we are extending the (actually weak) Semantic Discovery mechanism of the IoT oneM2M standard. The goal is to enable an easy and efficient discovery of information and a proper internetworking with external source/consumers of information (e.g. a data bases in a smart city or in a firm), or to directly search information in the oneM2M system for big data purposes. oneM2M ETSI standard has currently a rather weak native discovery capabilities that work properly only if the search is related to specific known sources of information (e.g. searching for the values of a known set of containers) or if the discovery is very well scoped and designed (e.g. the lights in a house). We submitted our vision in ETSI project submission "Semantic Discovery and Query in oneM2M" (currently under evaluation by ETSI) for extending oneM2M with a powerful Semantic Resource Discovery Service, taking into account additional constraints, such as topology, mobility (in space), intermittence (in time), scoping, routing ...

7.8. Empirical study of Amdahl's law on multicore processors

Participants: Carsten Bruns, Sid Touati.

Since many years, we observe a shift from classical multiprocessor systems to multicores, which tightly integrate multiple CPU cores on a single die or package. This shift does not modify the fundamentals of parallel programming, but makes harder the understanding and the tuning of the performances of parallel applications. Multicores technology leads to sharing of microarchitectural resources between the individual cores, which Abel et al. classified in storage and bandwidth resources. In this research report [39], we empirically analyze the effects of such sharing on program performance, through repeatable experiments. We show that they can dominate scaling behavior, besides the effects described by Amdahl's law and synchronization or communication considerations. In addition to the classification of Abel et al., we view the physical temperature and power budget also as a shared resource. It is a very important factor for performance nowadays, since DVFS over a wide range is needed to meet these constraints in multicores. Furthermore, we demonstrate that resource sharing not just leads a flat speedup curve with increasing thread count but can even cause slowdowns. Last, we propose a formal modeling of the performances to allow deeper analysis. Our work aims to gain a better understanding of performance limiting factors in high performance multicores, it shall serve as basis to avoid them and to find solutions to tune the parallel applications.

7.9. Communicating Networks of Data-Flow (sub)networks with limited memory

Participant: Robert de Simone.

Process Networks have been proposed a long time ago as models of concurrent, embedded streaming computations and communications, both amenable to formal analysis as models and executable as parallel program abstractions. As part of a larger effort at identifying precise connections between these models, programming models, and embedded parallel architectures altogether, we worked this year on the following problem: given a network of concurrent processes (Kahn-style) where each process is in turn a data-flow process network (SDF-style), can we decide in an efficient fashion (not NP-hard) whether a given assignment of communications to bounded local memories is schedulable (in a way that two simultaneous communications cannot require more than the available memory). A technical report is in preparation.

7.10. Behavioral Equivalence of Open Systems

Participants: Eric Madelaine, Cristian Grigoriu, Zechen Hou.

We consider Open (concurrent) Systems where the environment is represented as a number of processes which behavior is unspecified. Defining their behavioral semantics and equivalences from a Model-Based Design perspective naturally implies model transformations. To be proven correct, they require equivalence of "Open" terms, in which some individual component models may be omitted. Such models take into account various kind of data parameters, including, but not limited to, time. The middle term goal is to build a formal framework, but also an effective tool set, for the compositional analysis of such programs. In collaboration with ENS Lyon and Inria Lille, we studied an application of this approach to the verification of BIP architectures; this work extends previous dedicated approaches for compositional verification of BIP systems to data-dependent synchronizations [22]. Following last year results we have devised dedicated algorithms for checking equivalence of such systems [27], [41], currently under implementation in collaboration with ECNU Shanghai.

In order to facilitate the usage of our tools, we have also defined a language for defining open systems in terms of parameterized networks of synchronized automata (pNets, [4]), and implemented this language as an Eclipse-based editor in the VerCors tool (see Software section), together with interfaces to the semantic construction and equivalence checking algorithms.

7.11. Calculi with Union and Intersection types

Participants: Luigi Liquori, Claude Stolze.

Union and intersection types are interesting to improve actual programming languages static disciplines with alternative form of polymorphism. Since type inference is undecidable, our research vein focus on finding suitable "type decorations" in term syntax permitting to make type checking decidable, *i.e.* $\lambda x.x : (\sigma \to \sigma) \cap (\tau \to \tau)$ becomes $\langle \lambda x : \sigma.x, \lambda x : \tau.x \rangle : (\sigma \to \sigma) \cap (\tau \to \tau)$ in a fully-typed syntax. Those type systems uses intensively a subtyping relation stating *e.g.* that $\sigma \cap \tau \leq \sigma$ or $\sigma \leq \sigma \cup \tau$. Deciding whether $\sigma \leq \tau$ can be extremely difficult in complexity (space and time): actually, there are few algorithms in the literature dealing with union and intersection types. Recently [45] we have proved and certified in Coq a subtype algorithm of a type theory with union and intersection types; we have also extracted a running functional code. Subtyping constraints could be easily interpreted as temporal constraints in a suitable temporal algebra, like those that could be specified in CCSL. Advances of typed-calculi featuring those type disciplines are presented in [42], [31] and [14].

7.12. Bull, an Interactive Type Checker with Union and Intersection Types

Participants: Luigi Liquori, Claude Stolze.

Starting from our theoretical researches on Intersection and Union Types and related Subtype Theories, we have designed and implemented a prototype of an Interactive Typechecker based on the 2018 work on the Δ-framework [43], on the 2017 work on decidable subtyping logic for Intersection and Union types [45], and on our recent advances on the Δ-calculus [42] and [14]. The prototype is called *Bull*; Bull has a command-line interface where the user can declare axioms, terms, and perform computations. These terms can be incomplete, therefore the type checking algorithm uses unification to try to construct the missing subterms. A Read-Eval-Print-Loop allows to define axioms and definitions, and performs some basic terminal-style features like error pretty-printing, subexpressions highlighting, and file loading. Moreover, it can typecheck a proof and normalize it. We use the syntax of *Pure Type Systems* of Berardi to improve the compactness and the modularity of the kernel. Abstract and concrete syntax are mostly aligned: the concrete syntax is similar to the concrete syntax of the ITP Coq. We have also designed and implemented a *higher-order unification algorithm à la* Huet for terms, while typechecking and partial type inference are done by our *bidirectional refinement algorithm*. The refinement can be split into two parts: the essence refinement and the typing refinement. The bidirectional refinement algorithm aims to have partial type inference, and to give as much information as possible to the unifier. For instance, if we want to find a $?y$ such that $\vdash_\Sigma \langle \lambda x : \sigma.x, \lambda x : \tau.?y\rangle : (\sigma \to \sigma) \cap (\tau \to \tau)$, we can infer that $x : \tau \vdash ?y : \tau$ and that $\wr\, ?y \,\wr =_\beta x$. We are experimenting with classical examples in Bull, like the ones formalized by Pfenning with his Refinement Types in LF, and we are looking for examples taking into account preorders, constraints and operators (like *e.g.* $<, \leq, >, \geq, \cup, \cap$...) that could be interpreted as timed algebras expressions *à la* CCSL. This would be a little step toward the formal and certified definition of a simple timed type systems for the λ-calculus and a Timed Logical Framework.

The software can be actually retrieved on the GitHub repository Bull (registration to the BIL Inria data base is in progress).

7.13. Co-Modeling for Better Co-Simulations

Participants: Julien Deantoni, Giovanni Liboni.

A Collaborative simulation consists in coordinating the execution of heterogeneous models executed by different tools. In most of the approaches from the state of the art, the coordination is unaware of the behavioral semantics of the different models under execution; *i.e.*, each model and the tool to execute it is seen as a black box. We highlighted that it introduces performance and accuracy problems [44].

In order to improve the performance and correctness of co-simulations, we proposed a language to defined model behavioral interfaces, *i.e.*, to expose some information about the model behavioral semantics. We also proposed another language to make explicit the way to coordinate the different models by using dedicated connectors. The goal is to provide few information about the models to avoid intellectual property violations, but enough to allow an expert to make relevant choices concerning their coordination. The resulting models can then be exploited to generate a dedicated coordination, aware of the specificity of each model [29]. Future work mainly consists in experimenting a new co-simulation interface taking advantage of the model behavioral interface and proposed as a generalization of co-simulation interfaces from the state of the art.

This work is realized in the context of the GLOSE project (see Section 1) in collaboration with Safran and other Inria teams (namely HyCOMES and DiVerSE).

7.14. CCSL for Models Behavioral Composition

Participants: Julien Deantoni, Frédéric Mallet, Hui Zhao.

The growing use of models for separating concerns in complex systems has lead to a proliferation of model composition operators. These composition operators have traditionally been defined from scratch following various approaches differing in formality, level of detail, chosen paradigm, and styles. Due to the lack of proper foundations for defining model composition (concepts, abstractions, or frameworks), it is difficult to compare or reuse composition operators. In [17], we proposed research directions towards a unifying framework that reduces all structural composition operators to structural merging, and all composition operators acting

on discrete behaviors to event scheduling. Our belief is that CCSL, embedding both synchronous and asynchronous relations, is a good candidate to specify the event scheduling corresponding to the coordination of the different behaviors. However, as already stated in previous sections, to achieve such a status, some extensions to CCSL must be proposed. One of them was the possibility to prioritize events in the presence of synchronous relations. This was formally defined in [26] and implemented in the TimeSquare tool.Other interesting extensions are under study in the context of heterogeneous models, see Section 7.13.

As part of Zhao Hui's PhD work, we have proposed a language to bring together subsets of existing predefined languages in a bid to combine their expressiveness. Rather than trying to build the ultimate unified language, sum of all languages, we would rather select meaningful features in existing languages and build a new language based on those features. As an example of application, we have shown how to combine the functional models of Capella with the security models of SysML-sec in an ad-hoc security-aware language for functional analysis [36].

7.15. Expressing IoT security constraints

Participants: Stéphanie Challita, Robert de Simone.

In the framework of Inria Project Lab SPAI, we are considering extensions of the logical time constraint style of CCSL, in order to encompass locality information as well as the duality between (dynamic) agents and (static) resources. Once an appropriate framework has been defined to express occupancy of resources by agents through (logical) time, notions of access rights, enclaves, privileges and priorities may be encoded straightforwardly, and rules governing their proper secure use can be expressed as properties. Results will be presented at the completion of Stephanie Challita postdoctoral period.

7.16. Real-Time Systems Compilation

Participants: Dumitru Potop Butucaru, Hugo Pompougnac, Jad Khatib.

This work took place in the framework of the PIA ES3CAP project (see section 9.2.5) and in close collaboration with Inria PARKAS, Airbus, Safran Aircraft Engines, Kalray, and the IRT Saint-Exupéry. It funded the last year of Keryan Didier PhD thesis (before the Paris Kairos subteam was created).

The key difficulty of real-time scheduling is that timing analysis and resource allocation depend on each other. An exhaustive search for the optimal solution not being possible for complexity reasons, heuristic approaches are used to break this dependency cycle. Two such approaches are typical in real-time systems design. The first one uses unsafe timing characterizations for the tasks (*e.g.* measurements) to build the system, and then checks the respect of real-time requirements through a global timing analysis. The second approach uses a formal model of the hardware platform enabling timing characterizations that are safe for all possible resource allocations (worst-case bounds). So far, the practicality of the second approach had never been established. Automated real-time parallelization flows still relied on simplified hypotheses ignoring much of the timing behavior of concurrent tasks, communication and synchronization code. And even with such unsafe hypotheses, few studies and tools considered the (harmonic) multi-periodic task graphs of real-world control applications, and the problem of statically managing all their computational, memory, synchronization and communication resources.

Our work has provided the first demonstration of the feasibility of the second approach, showing good practical results for classes of real-world applications and multiprocessor execution platforms whose timing predictability allows keeping pessimism under control. This requires something that is missing in previous work:the tight orchestration of all implementation phases: WCET analysis, resource allocation, generation of glue code ensuring the sequencing of tasks on cores and the synchronization and memory coherency between the cores, compilation and linking of the resulting C code. This orchestration is conducted on a very detailed timing model that considers both the tasks and the generated glue code, and which includes resource access interferences due to multi-core execution. Orchestration is not a mere combination of existing tools and algorithms. Enabling predictable execution and keeping pessimism under control requires the formal and algorithmic integration of all design phases, which in turn required the definition of an application

normalization phase that facilitates timing analysis, of an original code generation algorithm designed to provide mapping-independent worst-case execution time bounds, and of new real-time scheduling algorithms capable of orchestrating memory allocation and scheduling.

Extensive results on the application of this method to real-file avionics case studies (>5000 unique nodes) mapped on the Kalray MPPA256 Bostan many-core have been presented in [15], [21] and in the PhD thesis of Keryan Didier, defended in September.

The Kalray MPPA platform provides excellent support for safety-critical real-time implementation, by allowing the computation of static WCET bounds. This is no longer true on more classical multi-cores such as those with ARM and POWER micro-architecture. We are currently aiming at extending our method to allow mapping on such multi-cores. Full schedulability guarantees cannot be provided on such platforms. Instead, our aim is to allow the synthesis of implementations that are functionally correct, efficient, and where impredictability is reduced to a minimum by eliminating controllable sources of timing variability. This line of work has been pursued in the context of the collaboration contracts with Airbus and IRT Saint-Exupéry. First results are promising.

Further extensions of our method are under way, most notably to cover timing predictable architectures different from the Kalray MPPA 256.

7.17. Formal Modeling of Concurrent Implementations

Participant: Dumitru Potop Butucaru.

Concurrent programming is notoriously difficult, especially in constrained embedded contexts. Threads, in particular, are wildly non-deterministic as a model of computation, and difficult to analyze in the general case. Fortunately, it is often the case that multi-threaded, semaphore-synchronized embedded software implements high-level functional specifications written in a deterministic data-flow language such as Scade or (safe subsets of) Simulink. We claim that in this case the implementation process should build not just the multi-threaded C code, but (first and foremost) a richer model exposing the data-flow organization of the computations performed by the implementation. From this model, the C code is extracted through selective pretty-printing, while knowledge of the data-flow organization facilitates analysis.

This year, we have proposes a language for describing such implementation models that expose the data-flow behavior hiding under the form of a multi-threaded program. The language allows the representation of efficient implementations featuring pipelined scheduling and optimized memory allocation and synchronization. We showed applicability on a large-scale industrial avionics case study and on a commercial many-core [24].

7.18. Scalability of Constraint Programming for Real-Time Scheduling

Participants: Dumitru Potop Butucaru, Robert de Simone.

Given two abstract modeling descriptions, one of a dataflow process network for the application, one of a block diagram structure for the computing platform and its interconnects, together with cost functions for the elementary computations and communications, one is bound to seek optimal mappings pairing the two. Amongst all the possible techniques, an obvious one consists in using general constraint solvers (real, integer, or boolean constraint programming, SMT solvers, CP solvers, etc.). Given the NP-hard nature of the problem, the issue here is to experimentally determine the empyrical complexity of various scheduling problems, and thus help in determining when solvers can be used for the resolution of scheduling problems.

In previous years we addressed this issue for ILP and SMT solvers. This year, we considered a Constraint Programming solver with dedicated support for modeling and solving real-time scheduling problems (IBM ILOG CPLEX CP Optimizer). The work was conducted in the framework of Bimael Iosif's student internship, and the writing of a paper is under way.

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

Safran : Desir/Glose We participate to the bilateral collaborative program Desir, put up by Safran to work with selected academic partners. We share the Glose project started in this program with two other Inria teams : HyComes, and DiverSE. The aim of the project is to improve early stages of system engineering by allowing early execution and co-simulation of heterogeneous models. The technical content of our contributions is described in section 7.13. A CIFRE PhD is funded by Renault on related topics.

IRT Saint-Exupery ATIPPIC This cooperative project aims at building a computing digital electronic structure of micro-satellites on ordinary, "COTS" processors. The project was accepted for 30 months and will reach completion by the end of 2019. It funds two temporary research engineers working under our own supervision, while exchanging extensively with the rest of the ATIPPIC project, which is actually physically hosted by Inria. The technical content of our contributions is described in section 7.2.

Airbus In the continuation of the ITEA3 ASSUME project, Airbus has provided funding for the extension of the Real-Time Systems Compilation method to allow parallelization onto multi-cores with classical ARM or POWER architecture. The technical content of our contributions is described in section 7.16. The technical content of our contributions is described in section 7.2.

IRT Saint-Exupery The CAPHCA project of IRT Saint-Exupéry has provided funding for the extension of the Real-Time Systems compilation method to allow parallelization onto timing predictable multi-cores different from the Kalray MPPA 256. The targets of this work are Infineon TC27x and FlexPRET.

Renault Software Lab We have started, at the end of 2018, a collaboration with Renault Software Labs on the definition of rules for ensuring safe maneuvers in autonomous vehicles. The rules express conditions from the environments, safety rules to preserve the integrity of the vehicles, driving legislation rules, local rules from the authorities. The rules must be updated dynamically when the vehicle evolves and are used to monitor at run-time the behavior of the ADAS. While the ADAS contains several algorithms relying on machine learning, the monitoring system must be predictive and rules must guarantee formally that the system does not cause any accident. So it can be seen as a way to build trustworthy monitoring of learning algorithms. A CIFRE PhD is funded by Renault on this topic and has started in April 2019.

Accenture Labs We have continued discussions with Accenture Labs, started in 2018, on Smart Contract languages for permissioned blockchains. A CIFRE funding is under way.
In recent years, various platform developments focused on so-called *private* (or *permissioned*) blockchain(s) and digital ledgers. Almost all private blockchains present their own implementation of Smart Contract. Between public and private blockchains we are observing a wide variety of different languages with different capabilities and limitations. Inspired by our researches in object-oriented languages [40], we aim at designing a language which might extend an object instance upon receiving a message, an ability referred to by Cardelli as *self-inflicted* operation. Public and private blockchains would take advantage of this novel capability in building safe and flexible intelligent smart contracts.

9. Partnerships and Cooperations

9.1. Regional Initiatives

9.1.1. Université Côte d'Azur Academy 1 and EUR DS4H

In the context of the local UCA-Jedi IDEX program and its RISE Academy, we were afforded a three years funding, including a postdoctoral position, for the "Smart IoT for Mobility" project. This project, lead by the LEAT UMR and Kairos, aims at building a formal language for the design of smart contracts in the context of a mobility project, in collaboration with Renault Software Labs and Symag, a subsidiary of BNP Paribas. This agreement was operational in preparing the larger ANR project SIM, that was accepted this year, while an even larger European project is under proposal.

9.1.2. PSPC-Region project ADAVEC

This project was recently accepted, and not yet started in practice. It associates Renault Software Labs with UCA (represented by our team), together with Avisto Telecom and EPICnPOC companies. The focus is on requirements and specification for Automated Driving Assistance, and more specially the transitions that need to be properly handled when control needs to be held back to the human driver.

9.2. National Initiatives

9.2.1. ANR Project SIM

The ANR SIM (Smart IoT for Mobility) is a PRCE project co-funded by ANR (AAPG 2019) and DGA for 42 months. The national coordinator is the LEAT (UMR CNRS) and the other partners are Renault Software Labs and Symag. The goal is to provide a formal meta-language to describe smart contracts that can be used in the context of an autonomous vehicles to provide services to the users. The services are related to the combined use of multi-model transportation systems by having a single smart contracts that can enforce all the intermediate transactions with all the actors involved (car manufacturing, parking lease, highway toll companies, insurances, bike rental companies).

9.2.2. Competitivity Clusters

The Kairos team is involved in the actions of the cluster SCS (Systèmes Communicants Sécurisés) and Frédéric MALLET is elected in the steering committee of SCS. One of the more prominent action is to build, in partnership with University Aix-Marseille, a Digital Innovation Hub, to open the access (with actions of transfer and valorization) to Digital Innovations for companies that would benefit from it, like public institutions (hospitals, human resources, employment institutions) or private companies that could use IoT for agriculture, tourism, smart infrastructures (harbours, buildings, cities).

9.2.3. CNRS GDRs

We are registered members of three GDR funded by CNRS : **SoC**^{2}, on topics of Hardware-software codesign and Non-Functional Property modeling for co-simulation; **LTP**, on verification and language design for reactive CPS systems; **GPL**, on software engineering and Domain-Specific Languages.

9.2.4. Inria Project Lab SPAI

This collaborative action, targeting *Security by Program Analysis for the IoT (SPAI)*, is headed by the Indes Project, and associated the Antique, Privatics and Celtique EPIs. See 7.15 for our contribution.

9.2.5. PAI ES3CAP

ES3CAP (Embedded Smart Safe Secure Computing Autonomous Platform) is a PIA (Programme d'Investissements d'Avenir) project. Its budget is of 22.2MEuros, over 36 months. The national coordinator is Kalray, and other partners include Safran, Renault, and MBDA. The objectives of the project are to:

- Build a hardware and software industry-grade solution for the development of computation-intensive critical application. The solution should cover the needs of industrial end users, and target multi/many-core hardware platforms. The solution will come with 3 to 6 usage profiles specific to various industries (automotive, aerospace, defence)

- Improve the technology readiness level of the proposed development flow from TRL4-5 (technology development) to TRL6-7, thus approaching as much as possible commercialization.
- Build an alternate, perennial ecosystem for critical real-time OSs and development tools for computer vision, data fusion and neural networks. The tools and components must be available on a prototyping and demonstration platform that is safe and secure.
- Capitalize on the convergence between the automotive and aerospace markets on subjects such as security, safety, decision making, and big data.

Our technical contributions to this project are described in 7.16. This project partially finances Hugo Pompugnac's PhD and Jad Khatib's post-doc.

9.3. International Initiatives

9.3.1. Inria International Partners

9.3.1.1. IIP TuMuLT

Title: Trustworthy Modeling using Logical Time

International Partner (Institution - Laboratory - Researcher):

E.C.N.U. (Shanghai, China) - Departement of Software Engineering and Computer Science - Zhang Min

Duration: 2018 - 2022

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

- Modeling the Uncertain Environments of Cyber-Physical Systems: Logical Time is one of the main scientific foundation of the KAIROS Team. From the background in theory of concurrency, we are used to consider mainly discrete control systems that can guarantee a functional determinism independently of any implementation-specific timing variation. Addressing Cyber-Physical Systems and the Internet of Things means widening those assumptions to consider the external environment, typically involving uncertainty, as part of the design. This task explores the definition of sound extensions to logical time to capture both the physical continuous behavior and make an abstract characterization as a statistical approximation.
- SMT For Logical Time: While synchronous systems usually focus on finite state-based control systems, our abstraction of logical time relies on both Boolean algebra (for synchronous operations) and integer arithmetic (for synchronizing mechanisms), In that context, SMT is a promising solution to solve systems that combine several theories. We had first results on this aspect [SCP'17] but we still need to increase the subset of constraints that can be addressed efficiently as well as the performances of the solving tools.
- Spatio-Temporal Specification for Trustworthy Intelligent Transportation Systems: Focusing on Intelligent Transportation Systems as a subset of Cyber-Physical Systems, we encounter specific problems. This task would focus on extensions of our framework for a spatio-temporal logics based on logical time. This means a description of the location of infrastructures as well as the ability to build constraints that depend both on time (logic or physical) and locations (logical or physical).
- Symbolic approaches for models and analysis of Open systems: Methods for analyzing and guaranteeing the properties of critical and complex systems, including their data and time depend aspects, have strongly evolved with the emergence of efficient SAT and SMT engines. We are working on novel methods combining classical verification paradigms with SMT approaches to create symbolic and compositional verification methods and tool platforms [22], [27].

  Collaboration will come in the form of scientific short or middle term visits, student exchanges (master and PhD), and organization of events (workshops and conferences).

9.3.1.2. Informal International Partners

- Luigi Liquori has a steady collaboration with researchers from University of Udine and Turin, Italy.
- We keep close informal relations with the Universities of Kiel and Bamberg Germany, in the context of the Synchronous Reactive academic community. We all attended the yearly Synchron seminar, held this year in Aussois (together with researchers from Verimag and the Parkas and Spades Inria teams).
- Frédéric Mallet has a collaboration with Peter Olvecsky from University of Oslo. He was funded in 2019 by a program of the French Embassy in Norway called Asgard.

9.3.2. Participation in Other International Programs

- PHC Cai Yuan Pei: The partnership is a joint funding from Campus France and Chinese Scholarship Council (CSC) to fund short exchanges of permanent staffs and long exchanges of PhD students. A 2-week visit was carried out by Frédéric Mallet in 2019, while Xiaohong Chen is visiting France during 3 months starting in mid-November. The program is funded for three years and a PhD student (Zhang Juan) will visit our team during 16 months in 2020.

9.4. International Research Visitors

9.4.1. Visits of International Scientists

- Xiaohong Chen, Assistant Professor at East China Normal University (Shanghai), from Nov 2019 to Feb 2020.
- Grygoriy Zholtkevych, Professor at V.N. Karazin Kharkiv National University (Ukraine), from Oct 2019 until Nov 2019.
- Peter Olvescky, Professor at University of Oslo, from November 24th to November 29th, 2019.
- Matteo Sereno, Professor, University of Turin, Italy, in May 2019.
- Thomas Ehrhard, University of Paris, in September 2019.

9.4.2. Visits to International Teams

9.4.2.1. Research Stays Abroad

- E. Madelaine spent 4 weeks visiting the Software Engineering an Computer Science department at ECNU Shanghai (2 weeks in May, 2 weeks in September), founded by the foreign expert program of ECNU; and 1 week visiting the Institute of Software of the Chinese Academy of Science (ISCAS, Beijing), funded by ISCAS.
- Marie-Agnès Peraldi Frati spent 10 days at Danang University in May 2019 in the context of the joined UCA/UD international DNIIT laboratory for student supervision and scientific meetings. The visit was funded by Mobility Contract Erasmus Mundus.
- Frédéric Mallet stayed three weeks in Shanghai in August 2019. He stayed one week in Hangzhou in September as part of a Chinese competition for oversea professors. He also stayed two weeks in Shanghai in November 2019 through the PHC Cai Yuan Pei program.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events: Organisation

10.1.1.1. General Chair, Scientific Chair

- F. Mallet was Track Scientific Chair for DATE2019 organized in Firenze in March 2019.
- E. Madelaine is Chair of the Steering Committee of the Formal Aspects of Component Systems (FACS) conference.

- J.Deantoni was Chair of the first international workshops on “Multi-Paradigm Modeling for Cyber-Physical Systems” and “Modeling Language Engineering and Execution”.
- J.Deantoni was Chair of “Computational Science” track of the RIVF 2019 conference.

10.1.2. Scientific Events: Selection

10.1.2.1. Chair of Conference Program Committees

- F. Mallet was Program Co-Chair of the 10th Workshop on Formal Techniques for Safety Critical Systems (FTSCS), organized as a satellite event of ICFEM 2019, in Shenzhen, Chine, in November 2019.
- F. Mallet was Program Co-Chair of ICTERI 2019 organized in Kherson, Ukraine, in May 2019.

10.1.2.2. Member of the Conference Program Committees

- J.Deantoni was in the program committee of: CoSim-CPS’19 and DSD’19
- F. Mallet was in the program committees of SEFM’19, DATE’19, FDL’19, DSD’19, Modelsward’19.
- D. Potop-Butucaru was in the program committee of FDL’19 and ACSD’19

10.1.2.3. Reviewer

- E. Madelaine is reviewer for the Formal Aspects of Component Systems (FACS), and industrial Formal Methods (iFM) conferences.
- L. Liquori and Claude Stolze reviewed for the conference FOSSACS/ETAPS.

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

- F. Mallet is now in the editorial board of Springer journal Software-Intensive Cyber-Physical Systems (SICS) dedicated to all topics around software in embedded and cyber-physical systems.
- F. Mallet was guest editor for a special issue of Elsevier Science of Computer Programming dedicated to the Theoretical Aspects of Software Engineering.

10.1.3.2. Reviewer - Reviewing Activities

- E. Madelaine was reviewer for the Journal of Systems and Software (JSS), and for Science of Computer Programming (SCP).
- L. Liquori and Claude Stolze reviewed for Fundamenta Informaticae.
- F. Mallet was reviewer for IEEE Transactions on Software Engineering (TSE) and Elsevier Journal on Microprocessors and Microsystems (MICPRO).

10.1.4. Invited Talks

- J.Deantoni was invited for a talk in the Danang University during the RIVF conference.
- J.Deantoni was invited for a talk during the “Computer Automated Multi-Paradigm Modelling 2019” meeting in Bellairs.
- F. Mallet was invited as a keynote speaker to the 11th IEEE TASE 2019 organized in Guilin, Chine (July 2019).
- L. Liquori is invited for a talk at the Bamberg University, Jan 2020.

10.1.5. Scientific Expertise

- F. Mallet is an elected member of the Conseil National des Universités (CNU) for section 27.
- F. Mallet has made an expertise of projects for the Belgium Agency of Research (FNRS).

10.1.6. Research Administration

- F. Mallet is Deputy Director of UMR I3S Laboratory and as such, member of its "comité de direction" and "conseil de laboratoire", together with the steering committee of the graduate school (EUR) DS4H.
- Sid Touati is member of the direction committee of I3S laboratory.
- M.A Peraldi-Frati is member of the I3S Laboratory council. She has been recently appointed member of UCA Academic Council.
- Luigi Liquori is member of the IFIP working group WG1.6 on Rewriting.

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

Licence : Sid TOUATI, Fondement machine, 75 heures eq TD, L1 informatique, Université Côte d'Azur.

Licence : Sid TOUATI, Architecture machine, 45 heures eq TD, L3 informatique, Université Côte d'Azur.

Licence : Sid TOUATI, Compilation, 33 heures eq TD, L3 informatique, Université Côte d'Azur.

Master: Sid TOUATI, Architectures et logiciels hautes performances, 81 heures eq TD, Master 1 informatique, Université Côte d'Azur.

Master international: Sid TOUATI, Advanced operating systems, 30 heures eq TD, Master 1 informatique, Université Côte d'Azur.

International Master: Frédéric Mallet, Safety-Critical Systems, 32h.

Master: Frédéric Mallet, Software Engineering, 32h.

Master : Robert de Simone, Formal Methods for NoC-based design, 36 heures eq TD, M2 International Ubinet, Université Côte d'Azur.

License: Marie-Agnès Peraldi-Frati teaches Web security (20h eq TD), Security of connected objects (20h eq TD), IoT Infrastructure deployment (20 H) and Large scale plateform for IoT (20h eq TD) in a licence cursus dedicated to Internet of Objects, Infrastructure and Applications.

Master:Marie-Agnès Paraldi Frati, Web Security and IoT Platform, 20h eq TD, Master 2 SICOM, Univ Avignon.

Master : Luigi Liquori, Peer-to-peer systems, 32 eq TD, Université Côte d'Azur.

Master: Luigi Liquori, Rewriting Systems and Pattern Matching, 12 eq TD, Université de Lorraine.

Ph.D.: Winter School on Theoretical Foundations of Computer Science, 4-9 February 2019, Georgia. Luigi Liquori. Peer-to-peer and related systems, International Black Sea University and Shota Rustaveli National Science Foundation of Georgia.

Ph.D.: 12th International School on Rewriting (ISR) July 2020, Spain. Luigi Liquori. Pattern matching λ-calculi.

Master: Julien Deantoni, Finite State Machine, 54h eq TD, Polytech'Nice.

Master: Julien Deantoni, Multi Paradigm Programming in C++, 54h eq TD, Polytech'Nice.

Master: Julien Deantoni, Domain Specific Languages, 32h eq TD, Polytech'Nice.

Master: Julien Deantoni, Language Interpreter, 32h eq TD, Polytech'Nice.

Master: Julien Deantoni, Micro-controller programming, 8h eq TD, Polytech'Nice.

Master: Dumitru Potop-Butucaru, A synchronous approach to the design of embedded real-time systems, 30h, EPITA Engineering School, Paris.

Master: Dumitru Potop-Butucaru, Real-time embedded systems, 42h, EIDD (École d'Ingenieur Denis Diderot), Paris

10.2.2. Supervision

- HDR : Julien Deantoni, *Towards Formal System Modeling: Making Explicit and Formal the Concurrent and Timed Operational Semantics to Better Understand Heterogeneous Models*, Université Côte d'Azur, Juillet 2019 [13]
- PhD : Claude Stolze, Combining union, intersection and dependent types in an explicitly typed lambda-calculus, Université Côte d'Azur, Dec 16th 2019, Luigi Liquori. [14]
- PhD in progress : Carsten BRUNS, Performance analysis and optimisation of C++ applications, Université Côte d'Azur, 2021, Sid TOUATI.
- PhD in progress : Giovanni Liboni, Coordination of discrete (Cyber) Models, Université Cote d'Azur, end 2021, Frédéric Mallet, Julien DeAntoni.
- PhD in progress : Joelle Abou Faysal, A Formal Language for Ensuring Safety Scenarios in Autonomous Vehicules, Université Cote d'Azur, 2022, Frédéric Mallet.
- PhD: Keryan Didier, Contributions to the safe and efficient parallelisation of hard real-time systems. Sorbonne University, September 19, 2019, Dumitru Potop-Butucaru.
- PhD in progress : Hugo Pompougnac, Sorbonne University, 2022, Dumitru Potop-Butucaru.

10.2.3. Juries

- Frederic Mallet was president of the PhD Jury of Ines Array, Université Cote d'Azur, March 2019.
- Eric Madelaine was member of the PhD jury of Frédéric Lemoine (CNAM, Paris), July 2019.
- Frederic Mallet was reviewer for the PhD Jury of Frédéric Giroudot (ISAE, Toulouse), December 13th 2019.
- Frederic Mallet was reviewer for the PhD Jury of Ngo Minh Thang NGUYEN (Centrale Supélec/CEA), December 16th 2019.
- Dumitru Potop-Butucaru was a member of the PhD jury of Lina Marsso (Univ. Grenoble-Alpes, Inria, CNRS, GrenobleINP), December 10, 2019.
- Dumitru Potop-Butucaru was a member of the PhD jury of Pierre Donat-Bouilloud (Sorbonne University), December 6, 2019.
- Luigi Liquori was member of the PhD jury of Claude Stolze, Université Côte d'Azur, December 2019.

10.3. Popularization

10.3.1. Interventions

Luigi Liquori. Table ronde: "Journée Internationale du m-Tourisme 2019. Blockchain for Tourism", M-Tourism Day, Telecom Valley, Panel, Cannes, 2019.

11. Bibliography

Major publications by the team in recent years

[1] 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, July 2010, 28 p. , Chapter 7, https://hal.inria.fr/inria-00495664

[2] Y. BAO, M. CHEN, Q. ZHU, T. WEI, F. MALLET, T. ZHOU. *Quantitative Performance Evaluation of Uncertainty-Aware Hybrid AADL Designs Using Statistical Model Checking*, in "IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems", December 2017, vol. 36, n^{o} 12, pp. 1989–2002 [*DOI :* 10.1109/TCAD.2017.2681076], https://hal.inria.fr/hal-01644285

[3] T. CARLE, D. POTOP-BUTUCARU, Y. SOREL, D. LESENS. *From Dataflow Specification to Multiprocessor Partitioned Time-triggered Real-time Implementation* *, in "Leibniz Transactions on Embedded Systems", November 2015 [*DOI :* 10.4230/LITES-V002-I002-A001], https://hal.inria.fr/hal-01263994

[4] L. HENRIO, E. MADELAINE, M. ZHANG. *A Theory for the Composition of Concurrent Processes*, in "36th International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE)", Heraklion, Greece, E. ALBERT, I. LANESE (editors), Formal Techniques for Distributed Objects, Components, and Systems, 2016, vol. LNCS-9688, pp. 175-194 [*DOI :* 10.1007/978-3-319-39570-8_12], https://hal.inria.fr/hal-01432917

[5] F. HONSELL, L. LIQUORI, P. MAKSIMOVIC, I. SCAGNETTO. *LLFP : A Logical Framework for modeling External Evidence, Side Conditions, and Proof Irrelevance using Monads*, in "Logical Methods in Computer Science", February 2017, https://hal.inria.fr/hal-01146059

[6] F. JEBALI, D. POTOP BUTUCARU. *Ensuring Consistency between Cycle-Accurate and Instruction Set Simulators*, in "18th International Conference on Application of Concurrency to System Design, ACSD 2018, Bratislava, Slovakia, June 25-29, 2018", 2018, pp. 105–114, https://doi.ieeecomputersociety.org/10.1109/ACSD.2018.00019

[7] L. LIQUORI, C. TEDESCHI, L. VANNI, F. BONGIOVANNI, V. CIANCAGLINI, B. MARINKOVIC. *Synapse: A Scalable Protocol for Interconnecting Heterogeneous Overlay Networks*, in "NETWORKING 2010 9th International IFIP TC 6 Networking Conference, Chennai, India, May 11-15, 2010. Proceedings", Chennai, India, M. CROVELLA, L. M. FEENEY, D. RUBENSTEIN, S. V. RAGHAVAN (editors), Lecture Notes in Computer Science, Springer Verlag, May 2010, vol. 6091, pp. 67–82 [*DOI :* 10.1007/978-3-642-12963-6_6], https://hal.inria.fr/hal-00909544

[8] F. MALLET, R. DE SIMONE. *Correctness issues on MARTE/CCSL constraints*, in "Science of Computer Programming", August 2015, vol. 106, pp. 78–92 [*DOI :* 10.1016/J.SCICO.2015.03.001], https://hal.inria.fr/hal-01257978

[9] 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, pp. 113-130 [*DOI :* 10.1016/J.TCS.2012.08.012], https://hal.inria.fr/hal-00764076

[10] G. NGO HOANG, L. LIQUORI, H. NGUYEN CHAN. *Backward-Compatible Cooperation of Heterogeneous P2P Systems*, in "15th International Conference on Distributed Computing and Networking - ICDCN 2014, Coimbatore, India, January 4-7, 2014", Coimbatore, India, Lecture Notes in Computer Science, Springer Verlag, January 2014, vol. 8314, pp. 287-301 [*DOI :* 10.1007/978-3-642-45249-9_19], https://hal.inria.fr/hal-00906798

[11] D. POTOP-BUTUCARU, R. DE SIMONE, J.-P. TALPIN. *Synchronous hypothesis and polychronous languages*, in "Embedded Systems Design and Verification", R. ZURAWSKI (editor), CRC Press, 2009, pp. 6-1-6-27 [*DOI :* 10.1201/9781439807637.CH6], https://hal.inria.fr/hal-00788473

[12] M. ZHANG, F. DAI, F. MALLET. *Periodic scheduling for MARTE/CCSL: Theory and practice*, in "Science of Computer Programming", March 2018, vol. 154, pp. 42–60 [*DOI :* 10.1016/J.SCICO.2017.08.015], https://hal.inria.fr/hal-01670450

Publications of the year

Doctoral Dissertations and Habilitation Theses

[13] J. DEANTONI. *Towards Formal System Modeling: Making Explicit and Formal the Concurrent and Timed Operational Semantics to Better Understand Heterogeneous Models*, Université Côte d'Azur, CNRS, I3S, France, July 2019, Habilitation à diriger des recherches, https://hal.inria.fr/tel-02427962

[14] C. STOLZE. *Combining union, intersection and dependent types in an explicitly typed lambda-calculus*, Université Côte d'Azur, December 2019, https://hal.archives-ouvertes.fr/tel-02406953

Articles in International Peer-Reviewed Journals

[15] K. DIDIER, D. POTOP-BUTUCARU, G. IOOSS, A. COHEN, J. SOUYRIS, P. BAUFRETON, A. GRAILLAT. *Correct-by-Construction Parallelization of Hard Real-Time Avionics Applications on Off-the-Shelf Predictable Hardware*, in "ACM Transactions on Architecture and Code Optimization", August 2019, vol. 16, n° 3, pp. 1-27 [*DOI :* 10.1145/3328799], https://hal.inria.fr/hal-02422789

[16] A. KHAN, F. MALLET, M. RASHID. *A Framework to Specify System Requirements using Natural interpretation of UML/MARTE diagrams*, in "Software and Systems Modeling", February 2019, vol. 18, n° 1, pp. 11-37 [*DOI :* 10.1007/S10270-017-0588-7], https://hal.inria.fr/hal-01670423

[17] J. KIENZLE, G. MUSSBACHER, B. COMBEMALE, J. DEANTONI. *A Unifying Framework for Homogeneous Model Composition*, in "Software and Systems Modeling", January 2019, pp. 1-19 [*DOI :* 10.1007/S10270-018-00707-8], https://hal.inria.fr/hal-01949050

[18] A. OUESLATI, P. CUENOT, J. DEANTONI, C. MORENO. *System Based Interference Analysis in Capella*, in "The Journal of Object Technology", 2019, vol. 18, n° 2, 14:1 p. [*DOI :* 10.5381/JOT.2019.18.2.A14], https://hal.inria.fr/hal-02182902

[19] D. YUE, V. JOLOBOFF, F. MALLET. *TRAP: trace runtime analysis of properties*, in "Frontiers of Computer Science", June 2020, vol. 14, n° 3, pp. 1-15 [*DOI :* 10.1007/S11704-018-7217-7], https://hal.inria.fr/hal-02402957

[20] Y. ZHANG, F. MALLET, Y. CHEN. *A verification framework for spatio-temporal consistency language with CCSL as a specification language*, in "Frontiers of Computer Science", 2019, vol. 14, n° 1, pp. 105–129 [*DOI :* 10.1007/S11704-018-7054-8], https://hal.inria.fr/hal-01924463

International Conferences with Proceedings

[21] P. BAUFRETON, V. BREGEON, K. DIDIER, G. IOOSS, D. POTOP-BUTUCARU, J. SOUYRIS. *Efficient fine-grain parallelism in shared memory for real-time avionics*, in "ERTS 2020 - 10th European Congress Embedded Real Time Systems", Toulouse, France, January 2020, https://hal.inria.fr/hal-02431187

[22] S. BLIUDZE, L. HENRIO, E. MADELAINE. *Verification of concurrent design patterns with data*, in "COORDINATION 2019 - 21st International Conference on Coordination Models and Languages", Kongens Lyngby,

Denmark, H. R. NIELSON, E. TUOSTO (editors), Coordination Models and Languages, Springer International Publishing, 2019, vol. LNCS-11533, pp. 161-181, Part 4: Coordination Patterns [*DOI :* 10.1007/978-3-030-22397-7_10], https://hal.archives-ouvertes.fr/hal-02143782

[23] P. CUENOT, P. BOUCHE, R. DE SIMONE, J. DEANTONI, A. OUESLATI. *Early validation of satellite COTS-on-board computing systems*, in "ERTS 2020 -10th European Congress on Embedded Real-Time Software and Systems", Toulouse, France, January 2020, https://hal.inria.fr/hal-02413867

[24] K. DIDIER, A. COHEN, D. POTOP-BUTUCARU, A. GAUFFRIAU. *Sheep in wolf's Clothing: Implementation Models for Dataflow Multi-Threaded Software*, in "ACSD 2019 - 19th International Conference on Application of Concurrency to System Design", Aachen, Germany, IEEE, June 2019, pp. 43-52 [*DOI :* 10.1109/ACSD.2019.00009], https://hal.inria.fr/hal-02422787

[25] F. GAO, F. MALLET, M. ZHANG, M. CHEN. *Modeling and Verifying Uncertainty-Aware Timing Behaviors using Parametric Logical Time Constraint*, in "DATE 2020 - Design, Automation and Test in Europe Conference", Grenoble, France, March 2020, https://hal.archives-ouvertes.fr/hal-02429533

[26] R. GASCON, J. DEANTONI, J.-F. LE TALLEC. *Priority in Logical Time Partial Orders with Synchronous Relations*, in "IEEE RIVF 2019 - Research, Innovation and Vision for the Future", Danang, Vietnam, March 2019, https://hal.inria.fr/hal-02078493

[27] Z. HOU, E. MADELAINE. *Symbolic Bisimulation for Open and Parameterized Systems*, in "PEPM 2020 - ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation", New-Orleans, United States, January 2020 [*DOI :* 10.1145/3372884.3373161], https://hal.inria.fr/hal-02406098

[28] M. HU, T. WEI, M. ZHANG, F. MALLET, M. CHEN. *Sample-Guided Automated Synthesis for CCSL Specifications*, in "DAC 2019 - 56th Annual Design Automation Conference 2019", Las Vegas, United States, ACM Press, June 2019, pp. 1-6 [*DOI :* 10.1145/3316781.3317904], https://hal.inria.fr/hal-02402971

[29] G. LIBONI, J. DEANTONI. *WIP on a Coordination Language to Automate the Generation of Co-Simulations*, in "FDL 2019 - Forum on specification & Design Languages", Southampton, United Kingdom, September 2019, https://hal.inria.fr/hal-02292048

[30] L. LIQUORI, R. GAETA, M. SERENO. *A Network Aware Resource Discovery Service*, in "EPEW 2019 - 16th European Performance Engineering Workshop", Milano, Italy, 2019, https://hal.inria.fr/hal-01895452

[31] L. LIQUORI, C. STOLZE. *The* $\Delta - calculus : SyntaxandTypes$, in "FSCD 2019 - 4th International Conference on Formal Structures for Computation and Deduction", Dortmund, Germany, June 2019, https://hal.archives-ouvertes.fr/hal-02190691

[32] S. V. MIERLO, J. DEANTONI, L. BURGUEÑO, C. VERBRUGGE, H. VANGHELUWE. *Towards Sketching Interfaces for Multi-Paradigm Modeling*, in "MPM4CPS - First International Workshop on Multi-Paradigm Modelling for Cyber-Physical Systems", Munich, Germany, September 2019, https://hal.inria.fr/hal-02336809

[33] S. VAN MIERLO, E. SYRIANI, D. BLOUIN, M. AMRANI, J. DEANTONI, M. WIMMER. *Preface to the 1st Multi-Paradigm Modeling for Cyber-Physical Systems (MPM4CPS 2019)*, in "MODELS Conference 2019 - ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems", Munich, Germany, IEEE, September 2019, 2 p. [*DOI :* 10.1109/MODELS-C.2019.00066], https://hal.inria.fr/hal-02428017

[34] Y. ZHANG, F. MALLET, H. ZHU, Y. CHEN. *A Logical Approach for the Schedulability Analysis of CCSL*, in "TASE 2019 - 13th International Symposium on Theoretical Aspects of Software Engineering", Guilin, China, IEEE, July 2019, pp. 25-32 [*DOI :* 10.1109/TASE.2019.00-23], https://hal.inria.fr/hal-02402976

[35] M. ZHANG, F. SONG, F. MALLET, C. XIAOHONG. *SMT-Based Bounded Schedulability Analysis of the Clock Constraint Specification Language*, in "FASE 2019 - Fundamental Approaches to Software Engineering", Prague, Czech Republic, April 2019, https://hal.inria.fr/hal-02080763

[36] H. ZHAO, L. APVRILLE, F. MALLET. *Meta-models Combination for Reusing Verification Techniques*, in "MODELSWARD 2019 - 7th International Conference on Model-Driven Engineering and Software Development", Prague, Czech Republic, SCITEPRESS - Science and Technology Publications, February 2019, pp. 39-50 [*DOI :* 10.5220/0007261000390050], https://hal.inria.fr/hal-02080768

Scientific Books (or Scientific Book chapters)

[37] A. SCHULZ-ROSENGARTEN, R. VON HANXLEDEN, F. MALLET, R. DE SIMONE, J. DEANTONI. *Time in SCCharts*, in "Language, Design Methods, and Tools for Electronic System Design", Springer, December 2019, pp. 1-25 [*DOI :* 10.1007/978-3-030-31585-6_1], https://hal.inria.fr/hal-02434885

[38] H. ZHAO, L. APVRILLE, F. MALLET. *A Model-Based Combination Language for Scheduling Verification*, in "Model-Driven Engineering and Software Development", Springer International Publishing, 2020, https://hal.telecom-paristech.fr/hal-02430903

Research Reports

[39] C. BRUNS, S. TOUATI. *Empirical study of Amdahl's law on multicore processors*, Inria Sophia-Antipolis Méditerranée ; Université Côte d'Azur, CNRS, I3S, France, December 2019, n^{o} RR-9311, https://hal.inria.fr/hal-02404346

[40] A. CIAFFAGLIONE, P. D. GIANANTONIO, F. HONSELL, L. LIQUORI. *A protoype-based approach to object reclassification*, Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France, 2019, https://hal.inria.fr/hal-01646168

[41] Z. HOU, E. MADELAINE, J. LIU, Y. DENG. *Symbolic Bisimulation for Open and Parameterized Systems - Extended version*, Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France ; East China Normal University (Shanghai), November 2019, n^{o} RR-9304, 47 p. , https://hal.inria.fr/hal-02376147

Scientific Popularization

[42] L. LIQUORI, C. STOLZE. *The Delta-calculus: syntax and types*, in "FSCD 2019 - 4th International Conference on Formal Structures for Computation and Deduction", Dortmund, Germany, 2019-06-24 and 2019-06-24, 2019, https://arxiv.org/abs/1803.09660 , https://hal.archives-ouvertes.fr/hal-01963662

References in notes

[43] F. HONSELL, L. LIQUORI, C. STOLZE, I. SCAGNETTO. *The Δ-framework*, in "38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, (FSTTCS)", 2018, vol. 122, pp. 37:1–37:21

[44] G. LIBONI, J. DEANTONI, A. PORTALURI, D. QUAGLIA, R. DE SIMONE. *Beyond Time-Triggered Co-simulation of Cyber-Physical Systems for Performance and Accuracy Improvements*, in "10th Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools", Manchester, United Kingdom, January 2018, https://hal.inria.fr/hal-01675396

[45] L. LIQUORI, C. STOLZE. *A Decidable Subtyping Logic for Intersection and Union Types*, in "TTCS 2017 - 2nd International Conference on Topics in Theoretical Computer Science", Topics in Theoretical Computer Science, Springer International Publishing, 2017, vol. LNCS-10608, pp. 74-90