Activity Report 2018

Team KAIROS

Logical Time for Formal Embedded System Design

Inria teams are typically groups of researchers working on the definition of a common project, and objectives, with the goal to arrive at the creation of a project-team. Such project-teams may include other partners (universities or research institutions).

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Embedded and Real-time Systems

Table of contents

Team KAIROS

Creation of the Team: 2017 January 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]

Faculty Members

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

PhD Students

Carsten Bruns [Univ Côte d'Azur, from Oct 2018]
Giovanni Liboni [Safran Tech, from Apr 2018]
Claude Stolze [Univ. Nice - Sophia Antipolis]
Hui Zhao [Inria, Labex]
Dongdong An [East China Normal University, until Apr 2018]
Marwa Hami [Univ. Sousse, until Jul 2018]

Technical staff

Luc Hogie [Ingénieur de recherche, CNRS]

Interns

Tony Gentilini [Inria, from May 2018 until Jul 2018]
Nader Haddad [Inria, from Mar 2018 until Aug 2018]
Zechen Hou [Inria, from Nov 2018]
Tetiana Kuziaieva [Univ. Cote d'Azur, from Mar 2018 until Aug 2018]
Pierre Tassel [Univ. Nice - Sophia Antipolis, from Jun 2018 until Aug 2018]
Elena Zambon [Inria, from Jun 2018 until Aug 2018]

Administrative Assistant

Patricia Riveill [Inria]

Visiting Scientists

Tengfei Li [East China Normal University, from Mar 2018 until Oct 2018]
Jing Liu [East China Normal University, until Jan 2018]

External Collaborators

Paul Bouche [Institut de recherche technologique Saint-Exupery, from Jun 2018]
Amin Oueslati [Institut de recherche technologique Saint-Exupery]

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 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

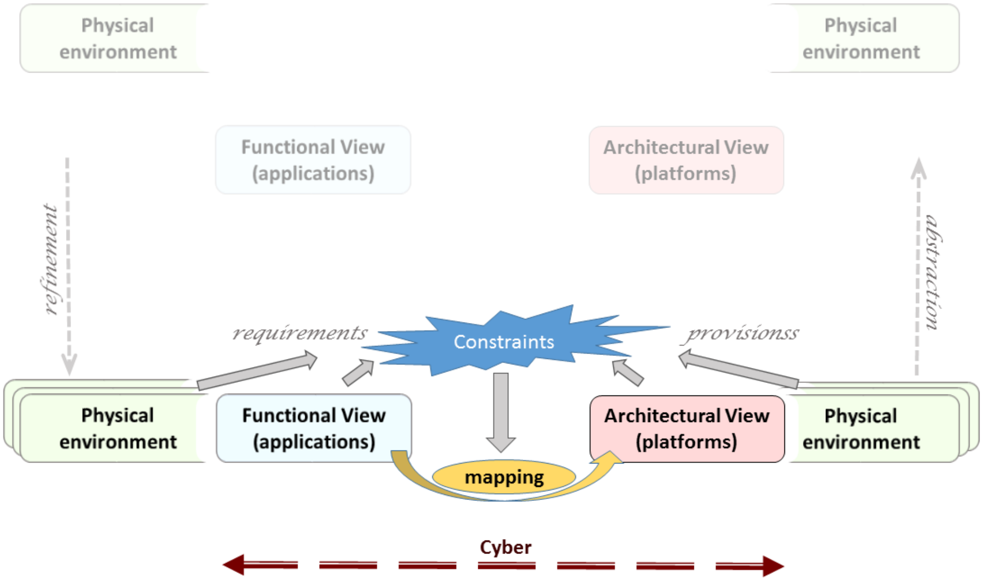

Figure 1. Cyber-Physical generic architectural features

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 discovery and 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 to Code and Optimization

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. 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 of A is less than the timestamp of B). We hope to extend 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 CCSL constraints needed to formalize logical clocks (in particular CCSL expressions like subclock, clock union, intersection and concatenation) and provide opportunities for an *ad hoc* polymorphic type theory. Logical time constraints seen as property types can be beneficially 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.

3.6. Object-oriented programming and logical time

We formalize in the past object-oriented programming features, like e.g. delegation-based and trait inheritance. We view our logical time model as a mean to enhance the description of timing constraints and properties on top of existing specification formalism. When considering general purpose object-oriented languages like Java, type-theory is a natural way to provide such properties. Currently, such languages do not have constructs nor special types to manage instants, time structures and instant relations like subclocking, precedence, causality, equality, coincidence, exclusion, independence, etc. CCSL provide 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 consider logical time constraints as behavioral type properties, and the design of programming language constructs and *ad hoc* type systems.

3.7. 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.

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

Our local collaborations on handheld, smartphone-like appliances have come to a close with the disappearance of most industrial partners at Sophia Antipolis (Texas Instruments mostly) and the end of the CIM PACA Design platform association. We are renewing collaborations with other local partners, with a focus on Smart Contract applied to connected objects in a IoT environment, and special concern for cloud/fog/edge allocation of computations, expressed with logical time modeling constraints. A speculative european consortium is put up under coordination by Easy Global Market (Sophia-based), and other initiatives with companies such as Symag, Accenture Labs Sophia, and Renault are also being developed.

5. Highlights of the Year

5.1. Highlights of the Year

5.1.1. Awards

BEST PAPER AWARD:

[16]
A. SCHULZ-ROSENGARTEN, R. VON HANXLEDEN, F. MALLET, R. DE SIMONE, J. DEANTONI. *Time in SCCharts*, in "Forum on specification & Design Languages", Munich, Germany, September 2018, pp. 5-16, Best Paper Award [*DOI :* 10.1109/FDL.2018.8524111], https://hal.inria.fr/hal-01898285

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.

Experimental version (2018) also includes algorithm for computing the symbolic semantics of Open Systems

- Participants: Antonio Cansado, Bartlomiej Szejna, Eric Madelaine, Ludovic Henrio, Marcela Rivera, Nassim Jibai, Oleksandra Kulankhina and Siqi Li
- 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

Framework for building social networks executable on web browsers, android and apple platforms
KEYWORDS: Framework - Peer-to-peer. - NoSQL - Mobile application - Social network - Publish-subscribe
SCIENTIFIC DESCRIPTION: [EN] myMed : an ad hoc framework to execute homogeneous social networks. The explosion of different Open Social Networks (OSN) "running" in the internet arena has changed the habits of mostly all of us. There are OSN almost for everything, from cooperative work, car pooling, healthcare, friendship, love, affairs, healthcare, information, gaming, etc. In almost all of the cases, there are no two OSN that are built by the same software producer, and - quite often - mostly of them work on a competitive basis, and - for many different reasons (business, privacy, politics, etc.) - they are not open source and they are hosted by their private servers. The interactions between those OSN is very little, since the Application Programming Interfaces (API) - if existing - are very weak and limited to access to the friends lists or to access OSN X with the password of OSN Y. Even worst, the possibility of programming or at least interconnecting common features (e.g. search on different data bases, link user names and passwords, chat, access to common data bases etc.) between different OSN advance slowly. The most used practice of interconnecting OSN is based on a form of "asymmetric viral communication" where one can relay (or post or publish) a duplicated copy or a pointer of a record published by user A in OSN X into another OSN Y, provided user A have also an account on OSN Y, and de facto forbidding in mostly cases the inverse operation, aka subscribing for a different user B on OSN Y to any publication of user A on OSN X even if user B does not have an account on OSN X. As such, the well know paradigm of Publish/Subscribe at the basis of many CSCW applications cannot be fully exploited inter social networks, leaving only their use intra social network, the latter use being of less impact in coordination and cooperation. Beside of the "business needs" of OSNs, the myMed (www.mymed.fr) meta-social network represents an important step toward the natural interconnections of social applications. This paper introduce the concept and the original structure of the myMed experimental system, as it was conceived as a common effort between five academic sites (Inria, Polytechnic of Turin, University of Turin, University of Piedmont Oriental, University of Nice Sophia Antipolis) and few local startups. myMed is an open source project, which facilitate and accelerate the development of ad hoc social applications (called in myMed jargon

"sociapps") running over an heterogeneous "Plateau" of platforms, such as PCs, Smartphones and Tablets running iOS and Android. myMed provides a rich framework for publishing, searching and subscribing to content: the engine is built on top of a distributed noSQL database. In its current version, it provides high scalability and fault tolerance. The myMed framework allows you to easily build social web applications: it features geolocalization, points of interest in charts, buddy lists, profile management, content/user reputation, built-in cooperation and coordination among different OSN running on it, proto OSN store, etc. A short description of the framework can be found in the appendix of this paper. In a nutshell the myMed framework is composed of: • a Software Development Kit (SDK) to develop fixed and mobile web sociapps, running on many Web browsers but also natively on Smartphones and Tablets equipped with Android or iOS. Sociapps, by their name, must have a strong social flavor (open social networks, closed social networks, enterprise social networks, micro/nano social networks and so on). Thanks to the rich, general-purpose, catalog of modules in the framework, every module can be freely used without interfering with other sociapps, in a true "Lego" fashion. The program is distributed under the Apache V2 free license. The TTM ("Time To Market") envisaged to develop a sociapp using the myMed SDK can be estimated from 1 to 3 months employing 1 or 2 senior programmers. • A "cloud" to execute the "sociapps" represented by a "backbone" of 50PCs, distributed through the "AlpMed" EuroRegion following some precise efficiency criteria (as example the presence of Internet running on optical fibers). Part of those PCs have a double function: o ensure the good behavior of all the running sociapps, and o offer services other than those offered by myMed, such as a web browser, an open-source Office suite, a private disk of little size, logically separated from the noSQL space that can be used by others users (we call this "an elastic usage of the myMed cloud PC"). Those PCs can be accessed via a private login and password generated by the framework on demand. The operating systems running on those PC (Ubuntu myMed Edition, UME) is also open source and it is based on a customized version of the Ubuntu operating system. To guarantee the quality of the execution of the sociapps, we require that all machines belonging to the backbone are constantly running (on state). • A little collection of "proof of concept" sociapps to validate, experiment, and testing the development kit and the execution cloud. These sociapps have been conceived with the precious help of the "Civil society" of the EuroRegion "AlpMed" (States, Regions, Prefectures, Associations, Chambers of Commerce, Municipalities, Universities, etc.) that have played a role of "maitre d'oeuvre" (or experts) in a given "applicative domain". The quasi totality of the sociapps are available on the myMed web platform at the address http://www.mymed.fr/?action=login but also on the most common mobile application stores, such as the Apple Store and the Google Play Store Markets. The myMed system is naturally divided into a backend and a frontend permitting a natural separation of concerns. The present and the future features of the myMed architecture together with the many open questions left open are: • myMed is distributed by construction and could be decentralized. The myMed backbone is 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. • myMed (should be) Easy to use. Start from the template, add or remove features, play with the design and the interface and you have an application ready to deploy on the myMed cloud and accessible to all myMed users via the proto store. • myMed is Extensible. myMed provides a modular architecture, since developers can easily install new modules and users can add or remove all the sociapps they like using the proto store. • myMed as a distributed Social Operating system? for many aspects myMed looks like an social operating system installed on a distributed and decentralize pull of PCs. Modifications of internal myMed modules would not affect the behavior of all sociapps using those modules. • myMed promotes collaboration and cooperation between OSN. The sharing of all social modules have the positive effect of greatly facilitate OSN interconnection • myMed users feature a two level profile. Having a myMed "basic profile" just give access to the store and to a "read only view" of each OSN X. For a full experience the user must fill the, so called, "extended profile" for OSN X which allows a full read/write access. • myMed and myMed sociapps can feature an ad hoc economical model? It is well know that economical models for OSN are quite often related to advertising, or buying intra OSN features. The myMed interconnectivity by construction, open a way to novel business models, like "the more you open the more you earn?" • myMed should feature an unique Human Computer Interface? a common template is provided to expert users that want to implement a proper OSN. Do they need to be compliant with some graphical chart? • myMed can run on different instances. Can different instances cooperate? As in higher-order languages, the same cooperation level featured between different OSNs running on one myMed instance can be applied on different myMed instances running on different hardware. This would be subject of a further evolution of inter-cooperation and connection of myMed instances and their OSN running inside it. A lot of care must be given in building coherent "meta basic profiles".

FUNCTIONAL DESCRIPTION: myMed is an experimental framework for implementing and deploying, on the top of a built-in cloud platform, many Open Social Networks (OSN) that could take advantage of sharing common software modules, hardware resources, making inter-communication and inter-interaction simpler and improving rapid development and deployement. myMed OSN are either accessible on web browsers and mobile platforms (android, ios). myMed is based on a peer-to-peer architecture and noSQL database technology. A number of experimental OSN are experimentally implemented and deployed to validate the framework : among them we mention myRiviera, myPaysduPaillon, mioConsolato, myBenevolat, myFondationSophiaAntipolis, myEurocin, myEurope, myAngel, ...

- Participants: Claudio Casetti, Luigi Liquori, 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

7. New Results

7.1. Schedulability of CCSL specifications via SMT

Participants: Frédéric Mallet, Robert de Simone.

The full expressive power of the CCSL language makes it very complex, if not impossible, to also find good, or even optimal, schedules as results of solving the CCSL constraints. Nevertheless, important subclasses can be devised, or efficient heuristics can be attempted. The study of CCSL scheduling decidability and efficient is a long-term source of theoretical developments in the team, here is a record of this year advances, split in two parts.

We have made progress on the inherent complexity of finding a schedule with a general CCSL specification. We have proved that the schedulability problem of CCSL is NP-hard. Then it makes sense to find whether there are still some practical ways to find solutions in specific cases. It turns out that in many cases, we can still find solutions in a reasonable duration. To do so, we have proposed [8] an encoding of CCSL specifications as an SMT (Satisfiability Modulo Theory) specification and we use Z3 and CVC4 as solvers for our experiments. Using a pure SAT solver is not possible for CCSL, as CCSL combines Boolean operations with arithmetics on unbounded integers. Using SMT allows to combine both. This encoding uses a sublogic called UFLIA that relies on quantified variables (boolean or integer), undefined functions on boolean and integers, and linear integer arithmetics. This logics is undecidable in the general case and the use of quantified variables makes it difficult to deal with, but we have found some interesting examples where we still get some results in a reasonable amount of time. We have also tried to identify subdomains where we get interesting results and we have focused on pure real-time schedulability problems. In that context, we showed that the schedulability problem for a set of real-time tasks reduces to the schedulability problem of CCSL specifications with a specific form (to be published).

The Clock Constraint Specification Language (CCSL) is a clock-based specification language for capturing causal and chronometric constraints between events in Real-Time Embedded Systems (RTESs). Due to the limitations of the existing verification approaches, CCSL lacks a full verification support for ’unsafe CCSL specifications’ and a unified proof framework. In this paper [18], we propose a novel verification approach based on theorem proving and SMT-checking. We firstly build a logic called CCSL Dynamic Logic (CDL), which extends the traditional dynamic logic with ’signals’ and ’clock relations’ as primitives, and with synchronous execution mechanism for modelling RTESs. Then we propose a sound and relatively complete proof system for CDL to provide the verification support. We show how CDL can be used to capture RTES and verify CCSL specifications by analyzing a simple case study.

7.2. Logical Time for the semantics of Reactive Languages

Participants: Frédéric Mallet, Robert de Simone.

This work was initiated during the sabbatical period of Reihard von Hanxleden, on leave from the University of Kiel (Germany), funded by the the UMR I3S laboratory.

The results won Best Paper Award at the Federated Design Languages (FDL) conference edition of 2018 [16]. The paper abstract follows:
Synchronous languages, such as the recently proposed SCCharts language, have been designed for the rigorous specification of real-time systems. Their sound semantics, which builds on an abstraction from physical execution time, make these languages appealing, in particular for safety-critical systems. However, they traditionally lack built-in support for physical time. This makes it rather cumbersome to express things like time-outs or periodic executions within the language. We here propose several mechanisms to reconcile the synchronous paradigm with physical time. Specifically, we propose extensions to the SCCharts language to express clocks and execution periods within the model. We draw on several sources, in particular timed automata, the Clock Constraint Specification Language, and the recently proposed concept of dynamic ticks. We illustrate how these extensions can be mapped to the SCChart language core, with minimal requirements on the run-time system, and we argue that the same concepts could be applied to other synchronous languages such as Esterel, Lustre or SCADE.

7.3. Dealing with uncertainty in logical time

Participants: Frédéric Mallet, Robert de Simone.

When uplifting the target of models to heterogeneous Cyber-Physical Systems, the relations from physical time (which governs Physical components) to logical time becomes an issue for proper abstraction in the design. Often, the other engineering discipline may know of "proto-logical" timing abstraction, but involving probabilistic/stochastic ingredients to link the declared logical clocks/events. As a results, several attempts have been made at extending the language to allow perceptive probabilistic structuring operators, that may link (unreachable) physical rhythms with their discretized, manageable counter-parts. Of course the feasibility of constraint solving remains the key issue for allowing extensions scarcely. Nevertheless, it should be noted that the focus on relevancy of relations between physical and logical times may in some case be an important concerns for non-IT scientists.

The reports on how early attempts can be found in [6], [10], [12]. The topic is far from closed, but as such these are valuable starts.

In the future, we plan to exploit these model extensions on practical application fields, including car trajectory computation with Renault Software Lab, security properties "with Time"in the ILP SPAI with other Inria teams, and micro-satellites in the ATIPPIC IRT Saint-Exupery project with Thales Alenia Space.

7.4. Behavioral semantics and equivalence notions for Open Systems

Participants: Eric Madelaine, Tengfei Li, Zechen Hou.

Model-Based Design 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. Following last year results we have published an experience paper [23] showing the applicability of this approach to show properties of a piece of the control software of a nano-satellite, specified using BIP architectures. Our work now turns on designing specific symbolic algorithms for model checking and equivalence checking (bisimulation) of such open systems, and also, as a specific application domain, to formalize the encoding of BIP architecture, extended with data constraints, into open pNets, aiming at a full approach for compositional verification of such systems. This work is done in collaboration with researchers from ENS Lyon and Inria Lille, and from ECNU Shanghai [23].

7.5. Logical Time for Safety Analysis and dependability

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

We have studied in the past the relevance of Logical Time for modeling of dynamic Non-Functional Properties (NFP) aspects of functional applications and/or execution platforms. In this setting, any recurring events may be seen as generating its own “rythm”, as a logical clock. The most obvious NFP aspects to consider were performance and power consumption, as important concerns of Real-Time Embedded systems. Recently we have turned towards fault tolerance and availability/dependability aspects. This was motivated by demands from industrial partners inside IRT Saint-Exupery, who tried to design in real terms the digital computing structure of micro-satellites using ordinary processor components from the Shelf (COTS), extremely sensible to solar radiations (creatings faults). We have put up a full model-based design of the proposed use case, which includes modeling of the fault-tolerant features, but also the independent modeling of waterfall propagation schemes from incidental faults to fully recognized dysfunctions, where the system is no longer operational. Current results are encouraging, as they build up natural specification styles using logical time on top of existing formalisms such as AltaRica, widely used in industry. Methodological advances are proposed to industrial partners in IRT Saint-Exupery, and primarily Thales Alenia Space. We plan to comfort our approach next year with dedicated tools for modeling and analysis, as well as translation towards existing formalisms such as AltaRica, seen as lower level in our context.

7.6. Co-Simulation of Cyber-Physical Systems

Participants: Julien Deantoni, Giovanni Liboni, Robert de Simone.

While we continued to study and envision the past, present and future of co-simulation in [11], we already obtained promising results. In [14], we highlighted the current problems of the FMI co-simulation standards and more generally of existing coordination between actors of the co-simulation. We also shown that providing appropriate mean to communicate with the actors according to their internal semantics allows for dedicated coordinator providing better results than existing ones (speed up can reach 25 with a perfect accuracy). As shown in [14], the functional correctness of co-simulation can be violated by a non appropriate coordination of co-simulation actors. To avoid such phenomenon, we explored in [17] the possibility to formally prove the correctness of a coordinator according to properties defined by the actors. This last work is greatly exploratory but Julien Deantoni did a Short Term Scientific Mission (in the context of the MPM4CPS cost action [1]) in the MSDL Lab in Antwerp to understand more deeply the problem and potential solutions. Preliminary interesting results have been obtained [2] and may be published in 2019.

7.7. Early Interconnect Contention Analysis

Participants: Amin Oueslati, Julien Deantoni.

In the context of the Atippic project, industrial partners are using the Capella system engineering language (http://polarsys.org/capella) to migrate a satellite control software on a totally new architecture platform based on “COTS” dual core processors. In order to better deal with the potential contention on the interconnect between the different cores, it was required to help for contention analysis. In this context and based on one of our software (GEMOC Studio: http://eclipse.org/gemoc) we developed an executable extension to Capella, from which simulation of Capella model can be used to obtain bus latency and bandwidth.

We are currently extending this simulation approach to ease Design Space Exploration based on variation of some parameters (typically parameters of the tasks that create traffic like for instance, periods or consumed/produced data size). First results have already been demonstrated to the IRT Saint-Exupery and should be published early 2019.

7.8. Process network models with explicit data size handling

Participants: Amin Oueslati, Robert de Simone.

We concluded our activities in the definition of a process network, inspired from established formalisms such as Ptolemy’s SDF, StreaMIT, and Thales Array-OL task graph languages. Our next formalisms described accurately how regular data structures (2-dimensional arrays or matrices mostly) get assembled or deassembled in actual data-flow computations for streaming intensive data/signal processing. This allows to allocate these computations to similar dedicated architectures (GPUs, TPUs) while making all kinds of parallelism (data-, task-, streaming) explicit. The resulting forms of specification are intently very close to representations that may be expressed in OpenMP or MPI, and cover the important class of Deep Networks filter stream models, which have raised tremendous interest lately in Artificial Intelligence.

7.9. Union and Intersection constraints

Participants: Luigi Liquori, Claude Stolze.

In [21], we introduced an explicitly typed λ-calculus with strong pairs, projections and explicit type coercions. The calculus can be parameterized with different intersection type theories, producing a family of calculi with related intersection typed systems. We proved the main properties like Church-Rosser, unicity of type, subject reduction, strong normalization, decidability of type checking and type reconstruction. We stated the relationship between the intersection type assignment systems and the corresponding intersection typed systems by means of an essence function translating an explicitly typed Delta-term into a pure λ-term one. We finally translated a term with type coercions into an equivalent one without them; the translation is proved to be coherent because its essence is the identity. The resulting generic calculus can be parametrized to take into account other intersection type theories as the ones in the Barendregt *et al.* book.

[1] http://mpm4cps.eu/

[2] http://mpm4cps.eu/STSM/reports/material/STSM_DeantoniJulien_Report_527.pdf

7.10. Logical frameworks with Union and Intersection constraints and Oracles

Participants: Luigi Liquori, Claude Stolze.

In [13], we introduced the Δ-framework, DLF, a dependent type theory based on the Edinburgh Logical Framework LF, extended with the *strong proof-functional connectives*, i.e. strong intersection, minimal relevant implication and strong union. Strong proof-functional connectives take into account the shape of logical proofs, thus reflecting polymorphic features of proofs in formulæ. This is in contrast to classical or intuitionistic connectives where the meaning of a compound formula depends only on the truth value or the provability of its subformulæ. Our framework encompasses a wide range of type disciplines. Moreover, since relevant implication permits to express subtyping, DLF subsumes also Pfenning's refinement types. We discuss the design decisions which have led us to the formulation of DLF, study its metatheory, and provide various examples of applications. Our strong proof-functional type theory can be plugged in existing common interactive proof assistants.

Moreover, in [7], we introduced two further extensions of LF, featuring monadic *locks*. A lock is a monadic type construct that captures the effect of an *external call to an oracle*. The oracle can be invoked either to check that a constraint holds or to provide a suitable witness. Such calls are the basic tool for *plugging-in*, i.e. gluing together, different type theories and proof development environments.

7.11. Object reclassification

Participant: Luigi Liquori.

In [19], we investigated, in the context of *functional prototype-based languages*, a calculus of objects which might extend themselves upon receiving a message, a capability referred to by Cardelli as a *self-inflicted* operation. We introduced a sound type system for this calculus which guarantees that evaluating a well-typed expression will never yield a *message-not-found* run-time error. The resulting calculus is an attempt towards the definition of a language combining the safety advantage of static type checking with the flexibility normally found in dynamically typed languages.

7.12. Object discovery

Participant: Luigi Liquori.

In [20], we proposed a Content Name System (CNS) discovery service, extending the current TCP/IP hourglass Internet architecture, that provides a new network aware content discovery service. Contents are addressed using “hypernames", whose rich syntax allow to specify hosts, PKI, fingerprint and optional logical attributes (tags) attached to the content name, such as e.g. mutable vs. immutable contents, digital signatures, owner, availability, price, etc. The CNS behavior and architecture is, partly, inspired by the Domain Name Service (DNS), and whose discovery process logic uses the Border Gateway Protocol (BGP) information allowing Internet to route between different Autonomous Systems (AS). The service registers and discovers object names in each Autonomous System (AS), and the content discovery process is inspired to the so called “valley-free” property. In the routing among different ASes (i.e., the BGP protocol) this is a property that avoids unjustified AS transit costs.

7.13. Code optimization for HPC and CPS programs

Participants: Sid Touati, Carsten Bruns, Robert de Simone.

Optimising HPC applications is a classical research area in computer science, complementary to intensive computation (which is an adjacent research community to HPC). Since decades, the most used languages are imperative ones (FORTRAN, C, etc). These languages are the closest to formal algorithms and low-level assembly codes. In intensive computing area, other kinds of languages and programming paradigms are used (interpreted languages for instance), but are far from HPC challenges, which tackle low level optimization (close to back-end compilation and processor micro-architectures).

We started a while ago to work on optimisation of HPC applications at C++ program level, where code and data are mixed in the same objects, allowing sophisticated programming methods that were not traditionally tackled in classical HPC programming (such as virtual classes, exception handling, etc). Currently, we are working on performance analysis and optimisation of linear algebra codes (BLAS) programmed with classes: this allows to extend BLAS computation to any kind of data (such as complex numbers) not only floating points. Our final aim is to apply and adjust this type of general C++ code optimization, to cover the spectrum of typical Kairos applications expressed from in C++ from high level formal specifications.

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

8.1.1. Safran: Desir/Glose

Participants: Julien Deantoni, Giovanni Liboni, Robert de Simone.

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 HyComes, and DiverSE Inria project teams. Technically, the goal of this project is to elaborate on the (under development) Safran's system engineering method to make it simulable at different steps of the development, possibly early in the design process and possibly mixing models at different maturity level. This project is strongly connected to results depicted in Section 7.6.

8.1.2. IRT Saint-Exupery ATIPPIC

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

In an attempt to build an extension of IRT Saint-Exupery from Occitanie to PACA region, the Thales Alenia Space company promoted the ATIPPIC project, to build the computing digital electronic structure of micro-satellites on ordinary, "COTS" processors. The project was accepted for 30 months, funds two temporary research engineers working under our own supervision, while exchanging extensively with the rest of the ATIPPIC project, which is actually hosted by Inria. The technical content of our contributions is described in Section 7.5 and 7.7.

8.1.3. Renault Software Lab

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

We have just 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 will start at the beginning of 2019.

8.1.4. Accenture Labs, Sophia

Participant: Luigi Liquori.

We started in 2018 a collaboration with Accenture Labs, Sophia on the following topics:

- Smart Contract languages for permissioned blockchains. We saw in the recent years the development of different platforms that focuses on the so-called private (or permissioned) blockchain(s) and digital ledgers. Almost the totality of private blockchain(s) 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. Both public and private blockchain often lack maturity and a formal semantic as they have been under pressure of the sudden and rapid explosion of blockchain popularity. A CIFRE PhD will start in 2019.
- Oracles in Smart Contract for IoT and and CPS. Oracles are third party services which are not part of the blockchain consensus mechanism. The main challenge with oracles is that people need to trust these sources of information. Whether a website or a sensor, the source of information needs to be trustworthy. The main challenges for oracles are dealing with small computation power, mobility, security and dealing with time. A CIFRE PhD is planned to start in 2019.

9. Partnerships and Cooperations

9.1. Regional Initiatives

9.1.1. Université Côte d'Azur Academy 1

In the context of the UCA Jedi IDEX, associated with the UCA ComUE, we have applied to a number of funding initiatives. The project Smart IoT for Mobility has been funded for three years by the Academy RISE. This project is lead by the LEAT and Kairos is building a formal language for the design of smart contracts in the context of a mobility project with Renault Software Labs. The smart contracts are persisted in a secured distributed ledgers (through blockchain technology). The SyMag company, a subsidiary of BNP Paribas, is providing the technology to access block chain with a ledger-agnostic API. A PhD (at LEAT) and a Post-doc (within Kairos) positions are funded by this project. A complementary funding has been asked to the ANR with the generic call 2019.

9.2. National Initiatives

9.2.1. Investissements d'Avenir: PIA Clarity

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

This project was funded by the LEOC Call (*Logiciel Embarqué et Objets Connectés*) of the national support programme *Investissements d'Avenir*. It ended in January 2018. Partners were: Thales (several divisions), Airbus, Areva, Altran, All4Tec, Artal, the Eclipse Fondation, Scilab Enterprises, CESAMES, U. Rennes, and Inria. The purpose of the project is to develop and promote an open-source version of the ARCADIA Melody system design environment from Thales, renamed CAPPELLA for that purpose. In this project we investigated extensions of Capella to enable simulation and analysis of mode automata in the context of model based system engineering.

9.2.2. 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,con Programming and Software Engineering (LaHMA group), LTP, Langages, Types et Preuves.

9.3. International Initiatives

9.3.1. Inria International Labs

The SACCADES LIAMA project came to a conclusion with the ending of the related Associated Team with ECNU Shanghai. We are actively working on a renewal of this colaboration, integrating the new generation of Professors there.

9.3.2. Inria International Partners

9.3.2.1. Declared Inria International Partners

- Luigi Liquori has a steady collaboration with researchers from University of Udine, and Turin, Italy.
- We collaborate with the University of Verona on topics of CPS co-simulation. This partly funds a support engineer on their side.
- M.A Peraldi-Frati participates in an international cooperation between University Côte d'Azur, University of Danang (Vietnam) and AUF. This collaboration crystallized through the DNIIT excellence initiative between Univ of Danang and UCA. M.A Peraldi-Frati is involved in the SLEGO project (Specific domain Language for Experience Global Orchestration)[22].

9.3.2.2. TuMuLT

Title: Trustworthy Modeling using Logical Time

International Partner (Institution - Laboratory - Researcher):

ECNU (China) - Software Engineering Institute - Min Zhang

Duration: 2018 - 2022

Start year: 2018

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

We have four main research directions:

- Modeling the Uncertain Environments of Cyber-Physical Systems (CPS): Logical Time was one of the main scientific foundations of the AOSTE 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 CPS means widening those assumptions to consider the external environment as part of the design. The environment obeys the law of physics that usually depend on physical time consideration with models that are approximation of the reality and that necessarily introduce a wide uncertainty on the behavior. 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 [25].
- 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, Solving a system of logical-time constraints is NP-complete but we strive to find efficient algorithms to solve sub-classes of well defined systems. In that context, SMT is a promising solution to combine and solve systems that combine several theories. We had first results on this aspect [8] 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. In addition to the temporal factor omni-present in real-time and embedded systems, a physical location plays also a central role. Functions of the system (like a train) must be done both at the right time AND at the right location. This task focuses 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).
- Open pNets: 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 satisfiability checking engines (SAT and SMT). We are working on novel methods combining classical verification paradigms (state-space construction and minimization, model-checking) with SMT approaches to create symbolic and compositional verification methods and tool platforms. We have interesting preliminary results [26], and collaborate actively on both fundamental results and prototype development.

9.3.3. Participation in Other International Programs

- PHC Xu Quangqi funded by ANR for International collaborations with China in 2008.
  - PI: Frédéric Mallet (France) and Zhang Min (China)
  - Title: SMT FOR LOGICAL TIME
  - Description: The main goal of the project was to build an efficient encoding of logical time in SMT solvers. This goal has been achieved (see New Result in Section 7.1).

9.4. International Research Visitors

- Xue-Yang ZHU, assistant research professor at Institute of Software, Chinese Academy of Science, Beijing.
- Zhang Min, Assistant Professor, ECNU Shanghai, 2 weeks in August 2018,
- Changbo Wang, Professor, Dean of Computer Science Department, ECNU Shanghai, 2 weeks in August 2018.

9.4.1. Internships

Zechen HOU benefited from an Inria International Internship Grant.

9.4.2. Visits to International Teams

9.4.2.1. Explorer programme

Julien Deantoni has spent one week visiting the Modelling, Simulation and Design Lab (MSDL) in Antwerp, funded by the MPM4CPS European cost action.

9.4.2.2. Research Stays Abroad

Eric Madelaine has spent 1 month visiting the Software engineering and computer Science department at ECNU Shanghai (2 weeks in May, 2 week in October).

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events Organisation

10.1.1.1. General Chair, Scientific Chair

- Robert de Simone organized the Scientific Program for the yearly Synchron seminar, held in November in Saint-Raphaël. He is also Steering Committee member of IEEE/ACM EmSoft a conference part of Embedded System Week.
- Eric Madelaine is chair of the steering committee of the Int. Symposium on Formal Aspects of Component Software (FACS: http://sevlab.postech.ac.kr/facs18/committees/)
- Frédéric Mallet was track co-chair for DATE 2018.
- Julien Deantoni was track co-chair for IEEE-RIVF (http://rivf2019.udn.vn/).

10.1.1.2. Member of the Organizing Committees

M-A. Peraldi-Frati and R. de Simone organized the Open Workshop Synchron 2018 in Saint-Raphaël.

10.1.2. Scientific Events Selection

10.1.2.1. Member of the Conference Program Committees

- M.A Peraldi-Frati is member of the IEEE-RIVF 2019 Program Committee.
- E. Madelaine is member of the PC of FACS'2018, VECoS'2018.
- R. de Simone is PC member for the conference MeMoCode, FDL, and EmSoft.
- Frédéric Mallet. Member of program committee for DATE'18, Euromicro DSD'18, FTSCS'18, FDL'18, TASE'18, Modelsward'18.
- Julien Deantoni is PC member RIVF'19, EXE'18, GEMOC'18, MDebug'18, DSD'18, MoMo'18.

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

- Eric Madelaine is Guest Editor of the Science of Computer Programming special issue for selected papers of the FACS'2014 symposium.
- Frédéric Mallet. Managing Guest Editor for a special issue of Elsevier Science of Computer Programming (SCP).

10.1.3.2. Reviewer - Reviewing Activities

- Eric Madelaine is reviewer for the journals: Science of Computer Programming (SCP), and Journal of Logical and Algebraic Methods in Programming (JLAMP).
- Marie-Agnès Peraldi-Frati : ACM Transactions on CPS, Forte2018.
- Luigi Liquori. Journal reviewer : Fundamenta Informaticae. Conference TPC: NICS'18, ICCE'18
- Frédéric Mallet. Journal reviewer for IEEE Transactions on Computer Aided Design of Integrated Circuits (TCAD), ACM Transactions on Embedded Computing Systems (TECS), ACM Transactions on Design Automation of Electronic Systems (TODAES), Elsevier Computers In Industry.
- Julien Deantoni. Journal reviewer for Software and Systems Modeling (http://www.i3s.unice.fr/~deantoni/SoSyM-review-certificate-Julien-DeAntoni.pdf) and for Computer Languages, Systems & Structures.

10.1.4. Research Administration

- F. Mallet is Deputy Director of UMR I3S Laboratory and as such, member of the 'comité de direction', 'conseil de laboratoire', 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

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.

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

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

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

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, 24h eq TD, Polytech'Nice.

Master: Julien Deantoni, Language Interpreter, 24h eq TD, Polytech'Nice.

10.2.2. Teaching Administration

- Sid Touati was the responsible of first year of computer science licentiate since 2011 till 2018.
- Sid Touati is a vice-director of the computer science department since 2017, in charge of the graduate students (licence).
- Frédéric Mallet is the coordinator of the International track of the Master of Computer Science since 2015.
- Frédéric Mallet is a member of the steering committee of the Graduate School DS4H (EUR DS4H).
- Master: Julien Deantoni, computer science internship management.

10.2.3. Supervision

- PhD in progress : Claude Stolze, A proof-functional type theory for intersection and union types, Université Côte d'Azur, end 2019, Luigi Liquori.
- PhD in progress : Carsten BRUNS, Performance analysis and optimisation of C++ applications, Université Côte d'Azur, 2021, Sid TOUATI.
- PhD in progress : Hui Zhao, Multiview System Integration for Cyber Physical Systems, Université Cote d'Azur, end 2019, Frédéric Mallet
- PhD in progress : Giovanni Liboni, Coordination of discrete (Cyber) Models, Université Cote d'Azur, end 2021, Frédéric Mallet, Julien DeAntoni

10.2.4. Juries

Robert de Simone was reviewer of the PhD thesis of Amaury Greillat (VERIMAG, Grenoble), and of the Habilitation thesis of Katell Morin-Allaury (TIMA, Grenoble).

M.A Peraldi-Frati : Examinator Thesis jury Slim Medimegh - CentraleSupélec University- Dec 2018.

Frédéric Mallet : Reviewer for the PhD thesis of NGuyen Van Hai - Université Paris Saclay - Central/Supélec, 27/09/2018

Frédéric Mallet : Reviewer for the PhD thesis of Martial Chabot - Université Grenoble Alpes - TIMA, 30/10/2018

11. Bibliography

Major publications by the team in recent years

[1] E. BOUSSE, T. DEGUEULE, D. VOJTISEK, T. MAYERHOFER, J. DEANTONI, B. COMBEMALE. *Execution Framework of the GEMOC Studio (Tool Demo)*, in "Proceedings of the 2016 ACM SIGPLAN International Conference on Software Language Engineering", Amsterdam, Netherlands, SLE 2016, October 2016, 8 p. , https://hal.inria.fr/hal-01355391

[2] B. COMBEMALE, J. DEANTONI, B. BAUDRY, R. B. FRANCE, J.-M. JÉZÉQUEL, J. GRAY. *Globalizing Modeling Languages*, in "Computer", June 2014, pp. 10-13, https://hal.inria.fr/hal-00994551

[3] 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

[4] M. E. VARA LARSEN, J. DEANTONI, B. COMBEMALE, F. MALLET. *A Behavioral Coordination Operator Language (BCOoL)*, in "International Conference on Model Driven Engineering Languages and Systems (MODELS)", Ottawa, Canada, T. LETHBRIDGE, J. CABOT, A. EGYED (editors), ACM, September 2015, n^{o} 18, 462 p. , to be published in the proceedings of the Models 2015 conference, https://hal.inria.fr/hal-01182773

[5] 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

Articles in International Peer-Reviewed Journals

[6] D. DU, P. HUANG, K. JIANG, F. MALLET. *pCSSL: A stochastic extension to MARTE/CCSL for modeling uncertainty in Cyber Physical Systems*, in "Science of Computer Programming", November 2018, vol. 166, pp. 71 - 88 [*DOI :* 10.1016/J.SCICO.2018.05.005], https://hal.inria.fr/hal-01898202

[7] F. HONSELL, L. LIQUORI, P. MAKSIMOVIC, I. SCAGNETTO. *Plugging-in Proof Development Environments using Locks in LF*, in "Mathematical Structures in Computer Science", 2018, vol. 28, n^{o} 9, pp. 1578–1605, https://hal.inria.fr/hal-01272647

[8] 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

[9] 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", November 2018 [*DOI :* 10.1007/S11704-018-7054-8], https://hal.inria.fr/hal-01924463

Invited Conferences

[10] F. MALLET. *Model-Based Systems Engineering for Cyber-Physical Systems: a (possible) roadmap for MARTE*, in "3rd International workshop on TIming Performance engineering for Safety critical systems CONFESTA/TIPS", Beijing, China, September 2018, https://hal.inria.fr/hal-01898291

International Conferences with Proceedings

[11] C. GOMES, C. THULE, J. DEANTONI, P. GORM LARSEN, H. VANGHELUWE. *Co-simulation: The Past, Future, and Open Challenges*, in "Leveraging Applications of Formal Methods, Verification and Validation. Distributed Systems", Limassol, Cyprus, B. MARGARIA (editor), Springer International Publishing, 2018, pp. 504–520, https://hal.archives-ouvertes.fr/hal-01913822

[12] C. Guan, Y. Ao, D. Du, F. Mallet. *xSHS: An Executable Domain-Specific Modeling Language for Modeling Stochastic and Hybrid Behaviors of Cyber-Physical Systems*, in "25th Asia-Pacific Software Engineering Conference", Nara, Japan, December 2018, https://hal.inria.fr/hal-01898219

[13] F. Honsell, L. Liquori, I. Scagnetto, C. Stolze. *The* $\Delta-$*framework*, in "38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018", Ahmedabad, India, 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS, December 2018, vol. 122, pp. 37:1–37:21 [*DOI :* 10.4230/LIPIcs.FSTTCS.2018.37], https://hal.archives-ouvertes.fr/hal-01701934

[14] 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

[15] F. Mallet, M. Zhang. *From Logical Time Scheduling to Real-Time Scheduling*, in "39th IEEE Real-Time Systems Symposium", Nashville, United States, December 2018, https://hal.inria.fr/hal-01971976

[16] *Best Paper*
A. Schulz-Rosengarten, R. von Hanxleden, F. Mallet, R. De Simone, J. Deantoni. *Time in SCCharts*, in "Forum on specification & Design Languages", Munich, Germany, September 2018, pp. 5-16, Best Paper Award [*DOI :* 10.1109/FDL.2018.8524111], https://hal.inria.fr/hal-01898285.

[17] C. Thule, C. Gomes, J. Deantoni, P. G. Larsen, J. Brauer, H. Vangheluwe. *Towards the Verification of Hybrid Co-simulation Algorithms*, in "Workshop on Formal Co-Simulation of Cyber-Physical Systems (SEFM satellite)", Toulouse, France, June 2018, https://hal.inria.fr/hal-01871531

[18] Y. Zhang, H. Wu, Y. Chen, F. Mallet. *Embedding CCSL into Dynamic Logic: A Logical Approach for the Verification of CCSL Specifications*, in "ICFEM / FTSCS 2018", Gold Coast, Australia, November 2018, https://hal.inria.fr/hal-01929184

Research Reports

[19] 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, 2018, https://hal.inria.fr/hal-01646168

[20] L. Liquori, R. Gaeta, M. Sereno. *A BGP-aware discovery service*, Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France, 2018, https://hal.inria.fr/hal-01895452

[21] L. Liquori, C. Stolze. *The* $\Delta - calculus : syntaxandtypes$, Inria & Université Nice Sophia Antipolis, CNRS, I3S, Sophia Antipolis, France, 2018, https://arxiv.org/abs/1803.09660 , https://hal.archives-ouvertes.fr/hal-01963662

[22] M.-A. Peraldi-Frati, J.-L. Salvat, N. Le Thanh, T.-H. Hoang, T.-H.-H. Nguyen. *Infrastructure & Design of Embedded Connected-Object Services: Application to Activity Daily Live monitoring*, Laboratoire I3S / UNS ; Institut DNIIT, May 2018, https://hal.inria.fr/hal-01878140

[23] X. Qin, S. Bliudze, E. Madelaine, M. Zhang. *Using SMT engine to generate Symbolic Automata -Extended version*, Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France ; inria, June 2018, n^{O} RR-9177, https://hal.inria.fr/hal-01823507

Other Publications

[24] F. Verdier, P. De Filippi, F. Mallet, P. Collet, L. Arena, A. Attour, M. Ballator, M. Chessa, A. Festré, P. Guitton-Ouhamou, R. Bernhard, B. Miramond. *Smart IoT for Mobility: Automating of Mobility Value Chain through the Adoption of Smart Contracts within IoT Platforms*, September 2018, 17th Driving Simulation & Virtual Reality Conference (DSC 2018), Poster, https://hal.archives-ouvertes.fr/hal-01903049

References in notes

[25] D. Du, P. Huang, F. Mallet, M. Yang, K. Jiang. *MARTE/pCCSL: Modeling and Refining Stochastic Behaviors of CPSs with Probabilistic Logical Clocks*, in "FACS'16", Springer, October 2016, pp. 111–133, https://doi.org/10.1007/978-3-319-57666-4_8

[26] L. Henrio, E. Madelaine, M. Zhang. *A Theory for the Composition of Concurrent Processes*, in "36th Int. Conf. on Formal Techniques for Distributed Objects, Components, and Systems (FORTE)", Heraklion, Greece, E. Albert, I. Lanese (editors), LNCS, June 2016, vol. 9688, pp. 175–194 [*DOI :* 10.1007/978-3-319-39570-8_12], https://hal.inria.fr/hal-01432917