

RESEARCH CENTRE

Sophia Antipolis - Méditerranée

IN PARTNERSHIP WITH:

CNRS, Université Côte d'Azur

2021

ACTIVITY REPORT

Project-Team

KAIROS

## **Multiform Logical Time for Formal Cyber-Physical System Design**

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

### **DOMAIN**

**Algorithmics, Programming, Software  
and Architecture**

### **THEME**

**Embedded and Real-time Systems**

# Contents

<b>Project-Team KAIROS</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>2</b>
<b>2 Overall objectives</b>	<b>3</b>
<b>3 Research program</b>	<b>4</b>
3.1 Cyber-Physical co-modeling	4
3.2 Cyber-Physical co-simulation	5
3.3 Formal analysis and verification	5
3.4 Relation between Model and Code	6
3.5 Code generation and optimization	6
3.6 Extending logical frameworks with logical time	6
3.7 Adding Synchronous primitives and logical time to Object-oriented programming	7
3.8 Extensions for spatio-temporal modeling and mobile systems	7
3.9 Advanced Semantic Discovery protocols for IoT networks	7
3.10 Type Checking in Logical Frameworks with Union and Intersection Types (Delta-Framework)	7
<b>4 Application domains</b>	<b>7</b>
4.1 Cyber-Physical and Embedded System Design	7
4.2 Smart contracts for Connected Objects in the Internet Of Things	8
4.3 Safe Driving Rules for Automated Driving	8
4.4 IoT Advanced Network Resource Discovery	8
4.5 SARS-CoV2 virus detection using IoT entities	8
<b>5 Social and environmental responsibility</b>	<b>8</b>
5.1 Footprint of research activities	8
<b>6 New software and platforms</b>	<b>9</b>
6.1 New software	9
6.1.1 VerCors	9
6.1.2 TimeSquare	9
6.1.3 GEMOC Studio	10
6.1.4 BCool	11
6.1.5 JMaxGraph	11
6.1.6 Lopht	11
6.1.7 LoPhT-manycore	12
6.1.8 Bull ITP	13
6.1.9 CoSim20	13
6.1.10 Idawi	14
6.1.11 mlirlus	14
6.2 New platforms	15
6.2.1 Advanced Semantic Discovery	15
6.2.2 Asynchronous Contact Tracing	15
<b>7 New results</b>	<b>16</b>
7.1 Safety Rules for Autonomous Driving	16
7.2 Take-Over between Automated and Human Automotive Driving	16
7.3 Decision Process For Logical Time	16
7.4 Formalizing and extending Smart Contracts Languages with temporal and dynamic features	17
7.5 CCSL extension to Stochastic Logical Time	17
7.6 Semantic Resource Discovery in IoT	18
7.7 Federating Heterogeneous Digital Contact Tracing Platforms	18
7.8 Empirical study of Amdahl's law on multicore processors	18
7.9 Behavioral Equivalence of Open Systems	19

7.10 Co-Modeling for Better Co-Simulations . . . . .	19
7.11 Abstraction in Co-Modeling and Co-Simulation . . . . .	19
7.12 Engineering and Debugging of Concurrency . . . . .	20
7.13 Formal Operational Semantics of Existing Languages . . . . .	20
7.14 Efficient parallelism in shared memory . . . . .	21
7.15 A Language and Compiler for Embedded and Real-Time Machine Learning Programming . . . . .	21
7.16 Formal verification of logical execution time (LET) applications . . . . .	22
7.17 Trustworthy Fleet Deployment and Management . . . . .	23
7.18 Model-Based Serverless Platform for the Cloud-Edge-IoT Continuum . . . . .	23
7.19 On the generation of compiler friendly C code from CCSL specification . . . . .	24
7.20 On the program recognition with neural networks . . . . .	24
7.21 Advanced Semantic Discovery for IoT objects and Networks . . . . .	24
7.22 Asynchronous Contact Tracing to catch Covid-19 using IoT objects and IoT networks . . . . .	25
7.23 Interferences between parallel embedded applications . . . . .	25
<b>8 Bilateral contracts and grants with industry . . . . .</b>	<b>25</b>
8.1 Bilateral contracts with industry . . . . .	26
8.2 Bilateral grants with industry . . . . .	27
<b>9 Partnerships and cooperations . . . . .</b>	<b>27</b>
9.1 European initiatives . . . . .	27
9.1.1 Horizon Europe . . . . .	27
9.2 National initiatives . . . . .	27
<b>10 Dissemination . . . . .</b>	<b>28</b>
10.1 Promoting scientific activities . . . . .	28
10.1.1 Scientific events: organisation . . . . .	28
10.1.2 Scientific events: selection . . . . .	29
10.1.3 Journal . . . . .	29
10.1.4 Invited talks . . . . .	30
10.1.5 Scientific expertise . . . . .	30
10.1.6 Research administration . . . . .	30
10.2 Teaching - Supervision - Juries . . . . .	30
10.2.1 Teaching . . . . .	30
10.2.2 Supervision . . . . .	31
10.2.3 Juries . . . . .	32
10.3 Popularization . . . . .	32
10.3.1 Education . . . . .	32
10.3.2 Interventions . . . . .	32
<b>11 Scientific production . . . . .</b>	<b>32</b>
11.1 Major publications . . . . .	32
11.2 Publications of the year . . . . .	34
11.3 Other . . . . .	36
11.4 Cited publications . . . . .	37

## Project-Team KAIROS

*Creation of the Project-Team: 2019 July 01*

### Keywords

#### Computer sciences and digital sciences

- 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 (hardware, equipments...)
- B7.2. – Smart travel
- B8.1. – Smart building/home
- B8.2. – Connected city
- B9.5.1. – Computer science

## 1 Team members, 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, HDR]

### Faculty Members

- Julien Deantoni [Univ Côte d'Azur, Associate Professor, HDR]
- Nicolas Ferry [Univ Côte d'Azur, Associate Professor]
- Frederic Mallet [Univ Côte d'Azur, Professor, HDR]
- Marie-Agnès Peraldi Frati [Univ Côte d'Azur, Associate Professor]
- Sid Touati [Univ Côte d'Azur, Professor, HDR]

### Post-Doctoral Fellows

- Ankica Barisic [Univ Côte d'Azur]
- Jad Khatib [Inria, Until April 2021]

### PhD Students

- Joelle Abou Faysal [Renault, CIFRE]
- Joao Cambeiro [Univ Côte d'Azur]
- Mikhail Ilmenskii [IRT SystemX, From October 2021]
- Mansur Khazeev [Univ Côte d'Azur and Univ Innopolis - Russian Federation]
- Giovanni Liboni [Groupe SAFRAN, CIFRE, until Apr 2021]
- Hugo Pompougnac [Inria]
- Fabien Siron [Krono Safe, CIFRE]
- Enlin Zhu [CNRS]

### Technical Staff

- Luc Hogie [CNRS, Engineer]

## Interns and Apprentices

- Sebastien Aglae [Inria, from May 2021 until Jul 2021]
- Baptiste Allorant [École Normale Supérieure de Lyon, from Feb 2021 until Jun 2021]
- Corentin Fossati [Univ Côte d'Azur, from Mar 2021 until Aug 2021]
- Arseniy Gromovoy [Inria, from Mar 2021 until Aug 2021]
- Ryana Karaki [Inria, from May 2021 until Aug 2021]
- Abdul Qadir Khan [Inria, from May 2021 until Sep 2021]
- Oleksii Khramov [Inria, from Mar 2021 until Aug 2021]
- Ludovic Marti [Inria, from May 2021 until Aug 2021]
- Ines Najjar [CNRS, from Jun 2021 until Sep 2021]
- Behrad Shirmard [Inria and Univ Côte d'Azur and Politecnico di Torino, Italy, from May 2021 until Dec 2021]
- Pavlo Tokariev [Inria, from Mar 2021 until Aug 2021]

## Administrative Assistant

- Patricia Riveill [Inria]

## External Collaborator

- Francois Revest [Univ Côte d'Azur, from Sep 2021]

## 2 Overall objectives

The Kairos ambitions are 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

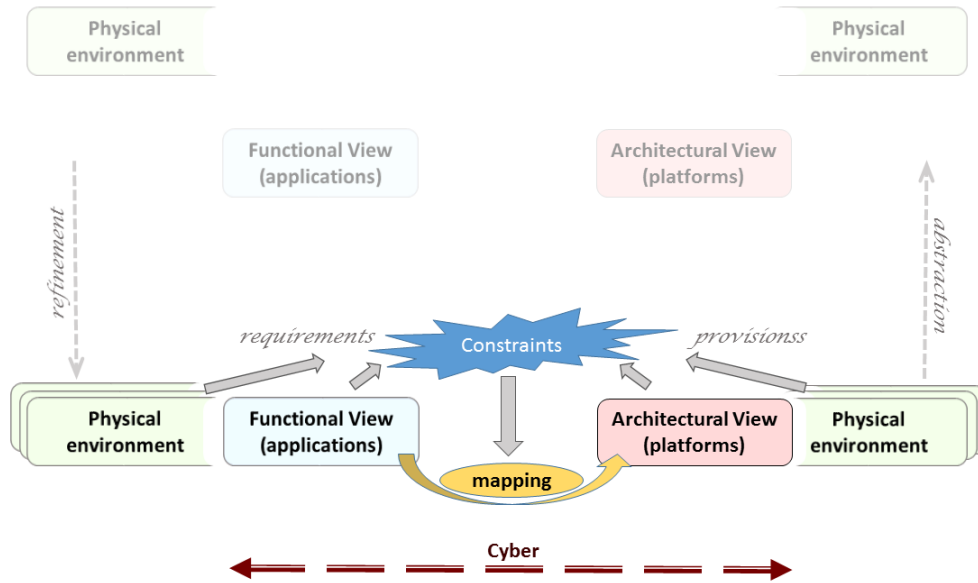


Figure 1: Cyber-Physical generic architectural features

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 [1, 12]. This language (named CCSL for Clock Constraint Specification Language), borrows from notions of Synchronous Reactive Languages [15], Real-Time Scheduling Theory, and Concurrent Models of Computations and Communication (MoCCs) in Concurrency Theory [13] 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 [6, 7, 9, 16] on the concern with resource discovery also in presence of mobility inherent to connected objects and Internet of Things [2, 10].

### 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 showed 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 [17]. 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 [5].



### 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 [3]. 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 of 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 level of specification for the compound design space -including functional application, execution platform, extra-functional requirements, implementation representation- 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-lambda-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 Adding Synchronous primitives and logical time to Object-oriented programming

We formalized in the past object-oriented programming features and safe static type systems featuring delegation-based or trait inheritance: well-typed program will never produce the `message-not-found` infamous run-time error. We will view logical time as a mean 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 kinds of 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 and Structured Operational Semantics à la Plotkin to capture valued-signals. Advances of Synchronous and Concurrent typed-calculi featuring those static time features could be applied for extending our object-calculus proposal [4].

### 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 CPSs (or mCPS) occur casually in real-life, 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 aim to provide fully distributed routing protocols to manage Semantic Resource Discovery in IoT.

### 3.9 Advanced Semantic Discovery protocols for IoT networks

Within the shield of the ETSI standardization Consortium, we study novel IoT Resource Discovery Protocols in the ETSI/oneM2M IoT standard. Our research leads to the production of few standards that will be included in the novel specification releases of the ETSI/oneM2M infrastructure, see [oneM2M release 4](#). We also developed a proof-of-concept platform, see Subsection 6.2.

### 3.10 Type Checking in Logical Frameworks with Union and Intersection Types (Delta-Framework)

Our work in this direction is embodied in the Bull theorem prover, a prototype theorem prover based on the Delta-Framework, described in the Software section.

## 4 Application domains

### 4.1 Cyber-Physical and Embedded System Design

System Engineering for CPS systems requires combinations of models, methods, and tools owing to multiple fields, software and system engineering methodologies as well as various digitalization of physical models (such as "Things", in Internet of Things (IoT)). Such methods and tools can be academic prototypes or industry-strength offers from tool vendors, and prominent companies are defining design flow usages around them. We have historical contacts with industrial and academic partners in the

domains of avionics and embedded electronics (Airbus, Thales, Safran). We also have new collaborations in the fields of satellites (Thales Alenia Space) and connected cars driving autonomously (Renault Software Factory). These provide us with current use cases and new issues in CPS co-modeling and co-design (Digital Twins) further described in New Results section. The purpose here is to insert our formal methods into existing design flows, to augment their analysis power where and when possible.

## 4.2 Smart contracts for 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 establish specification of behavioral system traces, applied to connected objects in a IoT environment. The new ANR project SIM is based on the definition of formal language to describe services for autonomous vehicles that would execute automatically based on the observation of what is happening on the vehicle or the driver. The key focus is on the design of a virtual passport for autonomous cars that would register the main events occurring on the car and would use them to operate automatic but trustworthy and reliable services.

In an independant effort, we investigate ways apply static type-checking for extensible smart-contracts in the domain of crypto-currencies and blockchains, while considering contracts that could be safely modified at run-time (*self-amended*) with a new behaviour.

## 4.3 Safe Driving Rules for Automated Driving

Self-driving cars will have to respect roughly the same safety-driving rules as currently observed by human drivers (and more). These rules may be expressed syntactically by temporal constraints (requirements and provisions) applied to the various meaningful events generated as vehicles interact with traffic signs, obstacles and other vehicles, distracted drivers and so on. We feel our formalisms based on Multiform Logical Time to be well suited to this aim, and follow this track in several collaborative projects with automotive industrial partners. This domain is a incentive to increase the expressiveness of our language and test the scalability of our analysis tools on real size data and scenari.

## 4.4 IoT Advanced Network Resource Discovery

Within the shield of ETSI (European Telecommunication Standard Institute), we participate to the ETSI IoT standardization (SmartM2M Technical Committee). Our skills on IoT Resource Discovery Protocols and on Type-based network protocols are of great help in providing realistic case studies for improving future releases of oneM2M IoT standard with powerful new IoT discovery features.

## 4.5 SARS-CoV2 virus detection using IoT entities

Within the shield of ETSI (European Telecommunication Standard Institute), we participate to the ETSI IoT standardization of Digital Contact Tracing IoT Protocols (SmartM2M Technical Committee). We introduces the method of *Asynchronous* Contact Tracing (ACT). ACT registers the presence of SARS-CoV-2 virus on IoT connected objects (waste water, or air conditioning filters, or dirty objects, or dirty cleaning tools, etc.) or connected locations (such as a shops, restaurants, corridors in a supermarket, sanitary facilities in a shopping mall, railway stations, airports terminals and gates, etc.) using Group Test (sometime called in the literature Pooling Test) [11].

# 5 Social and environmental responsibility

## 5.1 Footprint of research activities

Julien DeAntoni and Frédéric Mallet are members of the newly created working group at I3S that think about practical ways to measure and reduce the impact of our research activity on the environment. While digital technologies have potentially a big negative impact on the environment, they can also be part of the solution, sometimes.

## 6 New software and platforms

The team uses, evolves, and maintains the following softwares and platforms.

### 6.1 New software

#### 6.1.1 VerCors

**Name:** VERification of models for distributed communicating COmponents, 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 Contributions:** 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 (VerCors V4, 2019-2021) 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 2021 release now includes algorithms for : - testing both strong and weak versions of our symbolic bisimulation equivalence, starting from a candidate relation between states provided by the user, - on the fly generation of such symbolic bisimulations. As this process, due to the undecidability of symbolic equivalence, may not terminate, it comes in a "bounded" style, limiting the depth of the search.

**URL:** <https://team.inria.fr/scale/software/vercors/>

**Contact:** Eric Madelaine

**Participants:** Antonio Cansado, Bartlomiej Szejna, Eric Madelaine, Ludovic Henrio, Marcela Rivera, Nassim Jibai, Oleksandra Kulankhina, Siqi Li, Xudong Qin, Zechen Hou, Biyang Wang

**Partner:** East China Normal University Shanghai (ECNU)

#### 6.1.2 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.

**URL:** <http://timesquare.inria.fr>

**Contact:** Julien DeAntoni

**Participants:** Benoît Ferrero, Charles André, Frédéric Mallet, Julien DeAntoni, Nicolas Chleq

### 6.1.3 GEMOC Studio

**Name:** 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.
- Monilog, an extension for monitoring and logging executable domain-specific 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 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 designers to 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 provides various generic services such as graphical animation, debugging tools, trace and event managers, timeline.

**URL:** <http://gemoc.org/studio.html>

**Contact:** Benoît Combemale

**Participants:** Didier Vojtisek, Dorian Leroy, Erwan Bousse, Fabien Coulon, Julien DeAntoni

**Partners:** IRIT, ENSTA, I3S, OBEO, Thales TRT

#### 6.1.4 BCOol

**Name:** 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 coordinate the execution of, possibly heterogeneous, models.

**URL:** <http://www.gemoc.org>

**Contact:** Julien DeAntoni

**Participants:** Julien DeAntoni, Matias Vara Larsen, Benoît Combemale, Didier Vojtisek

#### 6.1.5 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.

**URL:** <http://www.i3s.unice.fr/~hogie/software/?name=jmaxgraph>

**Contact:** Luc Hogie

#### 6.1.6 Lopht

**Name:** 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 multi-processor 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.

**Contact:** Dumitru Potop Butucaru

**Participants:** Dumitru Potop Butucaru, Manel Djemal, Thomas Carle, Zhen Zhang

### 6.1.7 LoPhT-manycore

**Name:** 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-independent 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.

**Contact:** Dumitru Potop Butucaru

**Participants:** Dumitru Potop Butucaru, Keryan Didier

### 6.1.8 Bull ITP

**Name:** The Bull Proof Assistant

**Keywords:** Proof, Certification, Formalisation

**Scientific Description:** Bull is a prototype theorem prover based on the Delta-Framework, i.e. a fully-typed Logical Framework à la Edinburgh LF decorated with union and intersection types, as described in previous papers by the authors. Bull is composed by a syntax, semantics, typing, subtyping, unification, refinement, and REPL. Bull also implements a union and intersection subtyping algorithm. Bull has a command-line interface where the user can declare axioms, terms, and perform computations and some basic terminal-style features like error pretty-printing, subexpressions highlighting, and file loading. Moreover, Bull can typecheck a proof or normalize it. These lambda-terms can be incomplete, therefore the Bull's typechecking algorithm uses high-order unification to try to construct the missing subterms. Bull uses the syntax of Berardi's Pure Type Systems to improve the compactness and the modularity of its Kernel. Abstract and concrete syntax are mostly aligned and similar to the concrete syntax of the Coq theorem prover. Bull uses a higher-order unification algorithm for terms, while typechecking and partial type inference are done by a bidirectional refinement algorithm, similar to the one found in Matita and Beluga. The refinement can be split into two parts: the essence refinement relative to the computational part and the typing refinement relative to its logical content. Binders are implemented using commonly-used de Bruijn indices. Bull comes with a concrete language syntax that will allow user to write Delta-terms. Bull also features reduction rules and an evaluator performing an applicative order strategy. Bull also feature a refiner which does partial typechecking and type reconstruction. Bull distribution comes with classical examples of the intersection and union literature, such as the ones formalized by Pfenning with his Refinement Types in LF and by Pierce. Bull prototype experiment, in a proof theoretical setting, forms of polymorphism alternatives to Girard's parametric one.

**Functional Description:** Bull is a prototype theorem prover based on the Delta-Framework, i.e. a fully-typed Logical Framework à la Edinburgh LF decorated with union and intersection types. Bull also implements a subtyping algorithm. Bull has a command-line interface where the user can declare axioms, terms, and perform computations and some basic terminal-style features like error pretty-printing, subexpressions highlighting, and file loading. Moreover, it can typecheck a proof or normalize it. Further extensions will include adding a tactic language, code extraction, and induction.

**Release Contributions:** First stable version

**URL:** <https://github.com/cstolze/Bull/tree/master/bull>

**Contact:** Claude Stolze

**Participants:** Claude Stolze, Luigi Liquori

**Partners:** Inria, Université Paris-Diderot, Université d'Udine

### 6.1.9 CoSim20

**Name:** CoSim20: a Distributed Co-Simulation Environment

**Keywords:** Cosimulation, Development tool suite, Cyber-physical systems, Model-driven engineering



**Functional Description:** The Cosim20 IDE helps to build correct and efficient collaborative co-simulations. It is based on model coordination interfaces that provide the information required for coordination by abstracting the implementation details, ensuring intellectual property preservation. Based on such interfaces a system engineer can (graphically or textually) specify the intended communications between different simulation units. Then, the CoSim20 framework generates the artifacts allowing a distribution of the co-simulation where the simulation units communicate together at the required instants.

**URL:** <https://gitlab.inria.fr/glose/model-coordination-language>

**Publications:** [hal-03038527](#), [hal-02292048](#), [hal-01675396](#), [hal-03038547](#)

**Contact:** Julien DeAntoni

**Participants:** Giovanni Liboni, Julien DeAntoni

**Partner:** SAFRAN tech

#### 6.1.10 Idawi

**Keywords:** Java, Distributed, Distributed computing, Distributed Applications, Web Services, Parallel computing, Component models, Software Components, P2P, Dynamic components, Iot

**Functional Description:** Idawi is a middleware for the development and experimentation of distributed algorithms. It boasts a very general and flexible multi-hop component-oriented model that makes it applicable in many contexts such as parallel and distributed computing, cloud, Internet of Things (IOT), P2P networks. Idawi components can be deployed anywhere a SSH connection is possible. They exhibit services which communicate with each other via explicit messaging. Messages can be sent synchronously or asynchronously, and can be handled in either a procedural (with the optional use of futures) or reactive (event-driven) fashion. In the latter case, multi-threading is used to maximize both the speed and the number of components in the system. Idawi message transport is done via TCP, UDP, SSH or shared-memory.

Idawi is a synthesis of past developments of the COATI Research group in the field of graph algorithms for big graphs, and it is designed to be useful to the current and future Research project of COATI and KAIROS team-projects.

**URL:** <https://github.com/lhogie/idawi>

**Contact:** Luc Hogie

#### 6.1.11 mlirlus

**Name:** Lustre-based reactive dialect for MLIR

**Keywords:** Machine learning, TensorFlow, MLIR, Reactive programming, Real time, Embedded systems, Compilers

**Scientific Description:** We are interested in the programming and compilation of reactive, real-time systems. More specifically, we would like to understand the fundamental principles common to general-purpose and synchronous languages—used to model reactive control systems—and from this to derive a compilation flow suitable for both high-performance and reactive aspects of a modern control application. To this end, we first identify the key operational mechanisms of synchronous languages that SSA does not cover: synchronization of computations with an external time base, cyclic I/O, and the semantic notion of absent value which allows the natural representation of variables whose initialization does not follow simple structural rules such as control flow dominance. Then, we show how the SSA form in its MLIR implementation can be seamlessly extended to cover these mechanisms, enabling the application of all SSA-based transformations and optimizations. We illustrate this on the representation and compilation of the Lustre dataflow synchronous language. Most notably, in the analysis and compilation of

Lustre embedded into MLIR, the initialization-related static analysis and code generation aspects can be fully separated from memory allocation and causality aspects, the latter being covered by the existing dominance-based algorithms of MLIR/SSA, resulting in a high degree of conceptual and code reuse. Our work allows the specification of both computational and control aspects of high-performance real-time applications. It paves the way for the definition of more efficient design and implementation flows where real-time resource allocation drives parallelization and optimization.

**Functional Description:** The Multi-Level Intermediate Representation (MLIR) is a new reusable and extensible compiler infrastructure distributed with LLVM. It stands at the core of the back-end of the TensorFlow Machine Learning framework. mlirlus extends MLIR with dialects allowing the representation of reactive control needed in embedded and real-time applications.

**Release Contributions:** First open-source and public version.

**News of the Year:** First open-source version of mlirlus. Integration with the iree back-end. Paper accepted to ACM TACO and HiPEAC 2022.

**URL:** <https://github.com/dpotop/mlir-lus-public>

**Publication:** [hal-03043623](#)

**Authors:** Hugo Pompougnac, Dumitru Potop Butucaru

**Contact:** Dumitru Potop Butucaru

**Partner:** Google

## 6.2 New platforms

### 6.2.1 Advanced Semantic Discovery

**Participants:** Luigi Liquori, Marie-Agnes Peraldi Frati.

The [ETSI lab repository ASD](#) contains the source code of the ETSI oneM2M Advanced Semantic Discovery protocol simulator developed in OMNeT++ with the goal of providing a proof of concept and a performance evaluation of the powerful resource discovery that will be included in the [release 4 of oneM2M standard](#).

This work has been conducted in the [STF 589](#) of SmartM2M/oneM2M: Task 3: oneM2M Discovery and Query solution(s) simulation and performance evaluation.

### 6.2.2 Asynchronous Contact Tracing

**Participants:** Luigi Liquori, Abdul Qadir-Khan.

The [gitHub ACT repository](#) contains the first reference implementation of the ETSI standard Asynchronous Contact Tracing [50] using the oneM2M reference implementation oneM2M/ICON, provided by Telecom Italia Network (TIM).

This work has been conducted in the ETSI TC SmartM2M and TS eHEALTH Technical Committee. In 2022 we plan to terminate a reference implementation of the ACT standard and promote it.

## 7 New results

### 7.1 Safety Rules for Autonomous Driving

**Participants:** Frédéric Mallet, Joëlle Abou Faysal, Robert de Simone, Ankica Barisic.

Within the context of the CIFRE PhD contract with Renault Software Factory we are providing a formal language to describe and verify safety rules for autonomous vehicles. While the concrete syntax is meant to be close from what Safety Engineers expect, the underlying verification tools are based on our modeling and verification environments around logical time. Some safety constraints are related to temporal aspects, others relate to spatial constraints, all are expressed by our formalism [29, 43]. This is inline with one of the objectives of the team to provide languages, methods and tools for expressing spatio-temporal constraints in Cyber-Physical systems.

These safety rules are to be used in the ADAVEC project (See 7.2) to give some of the conditions that may trigger a demand to enter the manual mode, and the conditions that must be met when the driver demands to go back to the automatic mode. The construction of the language and the related tools build on our expertise in the domain and on a large survey on multi-paradigm modeling languages for the design of Cyber-Physical Systems [18].

### 7.2 Take-Over between Automated and Human Automotive Driving

**Participants:** Robert de Simone, François Revest.

This work was conducted in the context of the PSPC project ADAVEC 9.2, which is coordinated for UCA and Inria by Amar Bouali, and with industrial partners the local companies EpicnPoc and Avisto, with back-up from Renault Research Factory in Sophia-Antipolis.

The main objective is to study potential take-over and disengagement between human and IA automated driving. To the end, we considered the formal model-based specification of all needed elements, as well as requirements for safe driving rules and modeling of both external traffic environment or driver awareness conditions. Then a framework allowing to link these central modeling components to more concrete, usually existing simulation frameworks (including expertise from the company partners). The assembled system has been demonstrated partly by EpicnPoc at the Consumer Electronic Show (CES) in Las Vegas, in January, and in an updated form to the funding institutions in the Fall. Articles are under submission. The project should also benefit for the starting postdoc period of Ankica Barisic.

### 7.3 Decision Process For Logical Time

**Participant:** Frédéric Mallet.

One of the goals of the team and of our associated-team Plot4IoT is to promote the use of logical time in various application domains. This requires to have efficient solvers for CCSL. We have made some progress to define dynamic logics that combines a reactive imperative style to describe programs and a declarative style inspired by CCSL to capture properties to be satisfied. We use SMT solvers to decide which rewriting rules, among those that define the semantics of the reactive program, should be fired to satisfy the properties expressed as constraints. The benefit from classical CCSL is that we only rely on a decidable theory in SMT while the full CCSL language requires logics that are not decidable.

This dynamic logics is used along two different lines:

- The first line is to consider general CCSL specifications without any assumptions on the application domains. We have shown examples where combining CCSL with a reactive program allows for

semi-automatic verifications. We have an interactive proof system, that helps prove that some reactive program satisfies a set of formulas at all time [22].

- Second, we restrict the application domain to schedulability analysis. The original part is that there is no assumption on the execution pattern of the tasks to be scheduled. Then the question we try to solve is whether or not solving the CCSL problem gives us a useful feedback on how to build a valid schedule [21].

We also get a new automatic process based on reinforcement learning to help designers build the right CCSL specification. To do so we ask them to give examples of valid specification (traces) and we complete a partial CCSL specification so that it remains valid for all the given traces [26].

## 7.4 Formalizing and extending Smart Contracts Languages with temporal and dynamic features

**Participants:** Frédéric Mallet, Enlin Zhu, Ankica Barisic, Marie-Agnès Peraldi Frati, Robert de Simone, Luigi Liquori, Mansur Khazeev, Luis-Josef Amas.

"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 UMRs LEAT and Gredeg, and industrial partners, such as Symag and Renault Software Labs. Local funding was obtained from UCA DS4H EUR, which allowed preparation of the ANR project SIM that was accepted in 2019 but that actually started only on January 2021. One goal is to get acceptance from the lawyers while still preserving strong semantics for verification or runtime verification.

One main result of this year is to show how central models can become in the design of smart contracts. They allow to have a front end for reasoning, conducting verification, before generating some code (now only Solidity). Models allow to investigate alternative solutions, for instance, in our case, for studying alternative recovery protocol with the internal state of a smart contract is corrupted [44].

The internship of Luis-Josef Amas explored the potentiality of the Solidity SCL to safely and dynamically change their behavior, and proposed improvements for enhanced flexibility, inspired from [4].

Mansur Khazeev was recruited as UCA PhD in September 2020, while funded by the University of Innopolis, Russia. His topic will focus on Modifying Smart Contract Languages (SCL), and more specifically on building extensions to the SCLs landscape allowing to safely and statically safe *self-extend* a behaviour of a Solidity SCL.

## 7.5 CCSL extension to Stochastic Logical Time

**Participants:** Robert de Simone, Arseniy Gromovoy, Frédéric Mallet.

As a follow-up of research from previous years, we considered in the internship of Ubinet student Arseniy Gromovoy the issue of model-checking a statistical version of the stochastic CCSL version. To put things bluntly (and certainly too simply), the idea is to first build as a fixpoint the space of all reachable nominal configurations, and then only to add those of uncertain behaviors with decreasing probability, until a statistical cut-off point. Proceeding in that way allows regular description of state spaces, and ease the complexity of the model-checking algorithms.

These preliminary ideas have been taken up for a PhD proposal, which was accepted for grant by the EDSTIC doctoral school, and so will will to the PhD of Arseniy Gromovoy to start January 2022.

## 7.6 Semantic Resource Discovery in IoT

**Participants:** Luigi Liquori, Marie-Agnès Peraldi Frati, Behrad Shirmard.

Within the standards for M2M and the Internet of Things, managed by ETSI, [oneM2M](#), and in the context of the ETSI 589 contract, 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 inter-networking 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 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” 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 etc. This proposal was taken up, and a consortium was formed that issued a standard ETSI proposal, see [\[49, 48, 47, 45\]](#).

## 7.7 Federating Heterogeneous Digital Contact Tracing Platforms

**Participant:** Luigi Liquori.

In this work, done in collaboration with the University of Novi-Sad and the Mathematical Institute of the Serbian Academy of Sciences and Arts, we present a comprehensive, yet simple, extension to the existing systems used for Digital Contacts Tracing in Covid-19 pandemic. The extension, called BAC19, will enable those systems, regardless of their underlying protocol, to enhance their sets of traced contacts and to improve global fight against pandemic during the phase of opening borders and enabling more traveling. BAC19 is a structured overlay network, or better, a Federation of mathematical Distributed Hash Tables. Its model is inspired by the Chord and Synapse structured overlay networks. The paper presents the architecture of the Overlay Network Federation and shows that the federation can be used as a formal model of Forward Contact Tracing [\[39\]](#).

## 7.8 Empirical study of Amdahl’s law on multicore processors

**Participants:** Carsten Bruns, Sid Touati.

Since many years, we have been observing 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. We analyzed empirically the effects multiprocessor resource sharing on programs performance, through reproducible experiments. We found that they can dominate scaling behavior, besides the usual effects linked to Amdahl’s law or synchronization/communication issues; we also considered temperature and power budget, modeled as if shared resources. Second, we proposed a formal modeling of the performances to allow deeper analysis for precise understanding of performance limiting factors, in views of crafting solutions to tune these parallel applications, see [\[23\]](#).

## 7.9 Behavioral Equivalence of Open Systems

**Participants:** Eric Madelaine, Biyang Wang.

We consider Open (concurrent) Systems where the environment is represented as a number of processes whose 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 kinds 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.

Weak bisimulation for open systems [37] is significantly more difficult to decide in practice, because of the presence of unbounded loops in the definition of weak (unobservable) transitions. We have defined two different algorithms about weak bisimulation for open systems:

- the first one, given two open systems and a candidate (weak) bisimulation relation between their states, searches for weak transitions (chains of strong symbolic transitions) establishing a correspondence between the 2 systems. This is a bounded search, to avoid infinite loops of unobservable moves. As a consequence, this algorithm is only semi-decidable.
- the second one rewrites (simplifies) the symbolic automaton modeling an open system behavior, using a pattern-based approach that preserves weak bisimulation by construction.

The corresponding development of algorithms and implementation on our VerCors platform, is in progress, in collaboration with ECNU Shanghai. A journal paper submitted in Jan. 2021 is under review. A conference paper describing the weak bisimulation algorithms was published in June 2021 [42].

## 7.10 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 are seen as a black box. We highlighted that it introduces performance and accuracy problems [57].

In order to improve the performance and correctness of co-simulations, we proposed a language to define 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 little information about the models to avoid intellectual property violations, but enough to allow an expert to make relevant choices concerning their coordination. This year, based on such description we proposed the *Cosim20* framework, which showed possible to automatically generate a distributed coordination algorithm, which is more accurate and efficient than the one from the state of the art. Additionally, as an enabler of the distributed coordination algorithm we defined a moldable and extensible API for the simulation units so that the semantic awareness is also true at runtime.

Finally, it is worth noticing that Giovanni Liboni successfully defended his PhD thesis in Mars 2021 [36]. In its document Liboni presented a co-simulation that includes fault injection.

This work was realized in the context of the GLOSE project (see Section 8.1) in collaboration with Safran and other Inria project teams (namely HyCOMES and DiVerSE).

## 7.11 Abstraction in Co-Modeling and Co-Simulation

**Participants:** Julien Deantoni, Joao Cambeiro, Frédéric Mallet.

Nowadays, system engineering is an activity where different engineering domains are put together to define a solution to a specific complex problem. The particularity of system engineering lies in the need for several distinct stakeholders, where each of them is developing a part of the whole system with their own dedicated languages and tools. For each stakeholder, there are different tools and languages depending on the development cycle. At each stage of the development, they develop more or less abstract models, giving more or less fidelity according to the final product. Of course an increase in the fidelity of the model usually implies an increase in the simulation time of this model. In this context, Model Based System Engineering is using simulation to understand the emerging behavior of the whole system. For that it is required to coordinate/federate the artifacts produced by each stakeholder; and consequently to do a collaborative simulation of all of them.

Simulating the whole system may require a lot of time and computing power. This drastically reduces the possibility to explore new solutions by using design space exploration. This also drastically reduces the possibility for one stakeholder to understand the impact of its solution on the other stakeholders.

In the PhD thesis of Joao Cambeiro, started in October 2020, we are exploring different formalizations and frameworks to increase the agility and foster innovation in systems engineering. For that it should be possible to automatically use the appropriate level of fidelity for a specific model, according to the goal of the simulation; reducing then the simulation time of the whole system. Of course the selection of the fidelity level must be done cautiously since it must preserve the properties of interest. First results were obtained and a bird view of the framework is depicted in [24].

## 7.12 Engineering and Debugging of Concurrency

**Participants:** Julien Deantoni, Ludovic Marti, Ryana Karaki.

In the last year, in collaboration with Steffen Zschaler, King's college London, and the Diverse Inria project team, we experimented in the addition of different concurrent engines in the GEMOC studio. We figured out that it may be difficult for a user to debug concurrency since depending on the systems, some interleaving are not of interest and make difficult the exploration of significant ones.

From these experiments, we decided to built up upon the **Debugger Adapter Protocol** to allow concurrency debugging. It implies among other a place where *exploration strategies* can be defined by the user (the user being either the language designer or the system designer). The concurrency strategies hide the irrelevant interleaving to highlight the important ones. These strategies are defined independently of the execution engine and implemented into the GEMOC studio. A journal paper received a first minor review and is in the review pipe.

Also, in order to better understand the runtime state of a program under execution/debugging, we proposed a Domain Specific Language. When developping the behavioral semantics of such language, a language designer can add a description of what information should be represented during the debug and how. This was implemented in the GEMOC studio and the program animation was realized for two concrete syntax technologies, namely Xtext and KlighD. This work was presented in a paper and a poster available in [27].

## 7.13 Formal Operational Semantics of Existing Languages

**Participants:** Julien Deantoni, Joao Cambeiro.

Last year, in collaboration with other academic researchers, we developed the formal, concurrent and timed operational semantics of two existing languages: Lingua Franca and Platform Aware Model Sequence Diagrams.

**Lingua Franca** is a polyglot coordination language for concurrent and possibly time-sensitive applications ranging from low-level embedded code to distributed cloud and edge applications. It is developed by Edward Lee's group at University of Berkeley. The language is promising but no formal semantics has been defined, yet. In order to define the tool formal semantics of Lingua Franca, we defined, using our tool TimeSquare, a formal semantics proposal for Lingua Franca, see **MoCCML**. Currently, the working semantic proposal can be found [here](#).

Platform Aware Modal Sequence Diagram extends Modal Sequence Diagram, which allows to specify timed requirements, with a high level description of the hardware platform together with the deployment of Sequence Diagrams. This enable the early analysis of requirements validity. This has been extensively tested and the resulting artifacts are available [here](#), while a companion web page is accessible [here](#).

These exercises have different facets. 1) For Lingua Franca, the challenge is to specify the formal semantics of a language that requires time jumps (instead of time tick enumeration like usually done in CCSL); 2) for Platform aware MSD, it illustrates how the timing effects of an hardware platform can be studied early in the development process, in a formal way; 3) For both, it brings visibility to the work done in the team; 4) For both, it was the opportunity to develop the formal aspects of the GEMOC studio, and we now have generative techniques to automatically create an exhaustive exploration tool for any language developed using MoCCML in GEMOC studio. This exhaustive exploration encompasses the state of the concurrency model and the state of the runtime data, which can be abstracted by the user by overloading the *equals* operator. While still provided as an advanced prototype form, results of the exploration can be imported in the CADP tool for model checking, bringing model checking capabilities in any Domain-Specific Language written by following the GEMOC-MoCCML approach.

This has been published in [25] and [19] (to appear).

## 7.14 Efficient parallelism in shared memory

**Participant:** Dumitru Potop Butucaru.

This work took place in the framework of the PIA ES3CAP project.

The objective this year was to improve our coverage of the Kalray MPPA Coolidge many-core platform. To this end, we have:

- Improved our new memory access interference model. A research report has been published.
- Defined a method allowing our automatic parallelization method and tool (Lopht) to pilot code generation in the industrial SCADE MCG compiler, with the objective of generating code that runs under the Asterios RTOS.

## 7.15 A Language and Compiler for Embedded and Real-Time Machine Learning Programming

**Participants:** Hugo Pompougnac, Mikhail Ilmenskii, Dumitru Potop Butucaru.

We have continued and expanded this year this research axis aiming at connecting three research and engineering fields: Real-Time/Embedded programming, High-Performance Compilation, and Machine Learning. This work is financed by the Confiance.AI PIA project and by the ES3CAP project.

The Static Single Assignment (SSA) form has proven an extremely useful tool in the hands of compiler builders. First introduced as an intermediate representation (IR) meant to facilitate optimizations, it became a staple of optimizing compilers. More recently, its semantic properties<sup>1</sup> established it as a sound basis for High-Performance Computing (HPC) compilation frameworks such as **MLIR**, where

<sup>1</sup>Functional determinism while still allowing for limited concurrency.



different abstraction levels of the same application<sup>2</sup> share the structural and semantic principles of SSA, allowing them to co-exist while being subject to common analysis and optimization passes (in addition to specialized ones).

But while compilation frameworks such as MLIR concentrate the existing know-how in HPC compilation for virtually every execution platform, they lack a key ingredient needed in the high-performance embedded systems of the future—the ability to represent reactive control and real-time aspects of a system. They do not provide first-class representation and reasoning for systems with a cyclic execution model, synchronization with external time references (logical or physical), synchronization with other systems, tasks and I/O with multiple periods and execution modes.

And yet, while the standard SSA form does not cover these aspects, it shares strong structural and semantic ties with one of the main programming models for reactive real-time systems: dataflow synchrony, and its large and structured corpus of theory and practice of RTE systems design.

Relying on this syntactic and semantic proximity, we have extended the SSA-based MLIR framework to open it to synchronous reactive programming of real-time applications. We illustrated the expressiveness of our extension through the compilation of the pure dataflow core of the Lustre language. This allowed us to model and compile all data processing, computational and reactive control aspects of a signal processing application.<sup>3</sup> In the compilation of Lustre (as embedded in MLIR), following an initial normalization phase, all data type verification, buffer synthesis, and causality analysis can be handled using existing MLIR SSA algorithms. Only the initialization analysis specific to the synchronous model (a.k.a. clock calculus or analysis) requires specific handling during analysis and code generation phases, leading to significant code reuse.

The MLIR embedding of Lustre is non-trivial. As modularity based on function calls is no longer natural due to the cyclic execution model, we introduced a node instantiation mechanism. We also generalized the usage of the special undefined/absent value in SSA semantics and in low-level intermediate representations such as LLVM IR. We clarified its semantics and strongly linked it to the notion of absence and the related static analyses (clock calculi) of synchronous languages.

Our extension remains fully compatible with SSA analysis and code transformation algorithms. It allows giving semantics and an implementation to all correct SSA specifications. It also supports static analyses determining correctness from a synchronous semantics point of view.

First results, obtained in collaboration with Google, are presented in a paper accepted for publication in ACM TACO, and will be presented in the HiPEAC 2022 conference. A second paper is under writing. The mlirlus software has been registered as open-source.

A second PhD has been hired (Mikhail Ilmenskii) which will explore the specification of real-time properties and requirements for ML specifications, and the use of existing real-time mapping techniques to guide resource allocation in ML back-ends.

## 7.16 Formal verification of logical execution time (LET) applications

**Participants:** Robert de Simone, Dumitru Potop Butucaru, Fabien Siron.

We conducted deep investigations on how the PsyC language, developed by the Krono-Safe company, associates precisely features both from Synchronous Reactive languages (notions of Multiform Logical Time, logical and pseudophysical clocks, clock-based temporal programming primitives) and from Logical Execution Time languages (mandatory exact execution time intervals for non-elementary computation blocks). We derived a direct operational semantics as well as a translation semantics to Esterel or Lustre, which fully comply at the instants where external behaviors can be observed (input/output of temporal values). This led to a first WiP poster presentation at FDL 2021, and a more recent longer version accepted for publication at ERTS2022.

This work is conducted in the context of Fabien Siron CIFRE PhD thesis with the **Krono-Safe** SME, aiming at:

<sup>2</sup>Ranging from ML dataflow graphs and linear algebra specifications down to affine loop nests and optimized (tiled, vectorized,...) low-level code.

<sup>3</sup>A pitch tuning vocoder.

- Verifying the correctness of a static scheduling plan with respect to a logical execution time specification provided in the PsyC language designed and developed by Krono-Safe.
- Validating a logical time specification provided in PsyC with respect to logical time safety properties.

## 7.17 Trustworthy Fleet Deployment and Management

**Participant:** Nicolas Ferry.

This activity is starting as a result of Nicolas Ferry joining the team and as a continuation of his activities within the ENACT H2020 project. Continuous and automatic software deployment is still an open question for IoT systems, especially at the Edge and IoT ends. The state-of-the-art Infrastructure as Code (IaC) solutions are established on a clear specification about which part of the software goes to which types of resources. This is based on the assumption that, in the Cloud, one can always obtain the exact computing resources as required. However, this assumption is not valid on the Edge and IoT levels. In production, IoT systems typically contain hundreds or thousands of heterogeneous and distributed devices (also known as a *fleet of IoT/Edge devices*), each of which has a unique context, and whose connectivity and quality are not always guaranteed. In ENACT, we both investigated the challenge of automating the deployment of software on heterogeneous devices and of managing variants of the software which fit different types or contexts of Edge and IoT devices in the fleet.

Final results are presented in [30, 33, 34, 35, 20]. The natural next step is to investigate how to guarantee the trustworthiness of the deployment when (i) the quality of the devices is not guaranteed, (ii) the context of each device is continuously changing in an unanticipated manner, and (iii) software components are frequently evolving in the whole software stack of each device. In such context, ensuring the proper ordering and synchronization of the deployment actions is critical to improve the quality, trustworthiness and to minimize downtime. A proposal for an EU project (Horizon Europe call HORIZON-CL4-2021-DATA-01-05: Edge: Meta Operating Systems) has been submitted in October 2021 in which Kairos aims at leading research activities along this line.

## 7.18 Model-Based Serverless Platform for the Cloud-Edge-IoT Continuum

**Participants:** Nicolas Ferry, Ines Najjar.

One of the most prominent implementation of the serverless programming model is Function-as-a-Service (FaaS). Using FaaS, application developers provide source code of serverless functions, typically describing only parts of a larger application, and define triggers for executing these functions on infrastructure components managed by the FaaS provider. The event-based nature of the FaaS model is a great fit for Internet of Things (IoT) event and data processing. However, there are still challenges that hinder the proper adoption of the FaaS model on the whole IoT-Edge-Cloud continuum. These include (i) vendor lock-in, (ii) the need to deploy and adapt serverless functions as well as their supporting services and software stack to the cyber-physical context in which they will execute, and (iii) the proper orchestration and scheduling of serverless functions deployed on the whole continuum. To address the aforementioned challenges, as a collaboration with the SIS research group in SINTEF (Oslo, Norway) and as part of Ines Najjar master internship, we initiated a first prototyping platform for the design, deployment, as well as maintenance of applications over the IoT-Edge-Cloud continuum. In particular, our platform enables the specification and deployment of serverless functions on Cloud and Edge resources as well as the deployment of their supporting services and software stack over the whole IoT-Edge-Cloud continuum. Next step will be to investigate solutions for the proper and timely orchestration and scheduling of these functions. A proposal for a PHC Aurora project has been submitted together with SINTEF, Oslo, to further investigate these aspects.

## 7.19 On the generation of compiler friendly C code from CCSL specification

**Participants:** Baptiste Allorant, Sid Touati, Frédéric Mallet.

This work constitutes the research master internship of Baptiste Allorant, student from École Normale Supérieure de Lyon. The aim of the internship was to create a C code generator from CCSL that would be able to compute efficiently a schedule from a CCSL specification. CCSL was indeed defined alongside the UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) so as to provide a way to give an explicit time interpretation for a given model. The language inherits from synchronous languages the notion of discrete clock as well as some of its constructions but it gets rid of the variable and values associated with the clock to focus on the computation of the clock scheduling. It then adds new asynchronous constructors that greatly improve the expressivity of the language. A prototype named ECCOGEN has been implemented and some experiments were conducted to assess its efficiency. It is a quite reliable C code generator from CCSL. We have also managed to heavily improve its performances using profiling tools and trying to make compiler-friendly code.

There are however many improvements that can still be achieved. Semantically these improvements could consist in modifying the solver so that it became provable right and not only reliable. This would also mean investigating the issue caused by dead-ends (or bad paths) in order to see under which condition it is possible to avoid them. In terms of performance, the next step of the optimization would probably be the multi-threading of the generated code. Then it would be interesting to see whether GPU or architecture-specific instructions could be used to improve the overall performance. All this are planned to be continued during a PhD project that is scheduled to start in 2022.

## 7.20 On the program recognition with neural networks

**Participants:** Corentin Fossati, Sid Touati.

This is joint work with Enrico Formenti, professor at UCA.

Automatic program recognition is a well known undecidable problem in computer science. In this project, we aim to tackle this interesting question with artificial intelligence, precisely with neural network methods. Detecting what a code is doing or computing has many applications in reverse engineering, security code analysis, optimising compilation, etc. We are willing to address this open problem with pragmatic tools. We started this project with a research master internship of Corentin Fossati, from University Côte d'Azur. We worked on two main issues: 1) what would be an appropriate program representation for capturing its precise semantic and 2) how can we use neural network methods to compare between program representations. For the first point, we start by using simple data dependence graphs to represent the semantic of a code. For the second point, we had an initial tentative with image representation that failed. Then we found recent methods based on graph comparisons with neural networks. Our empirical results on some benchmarks are very promising, pushing us to continue this collaboration in the upcoming future.

## 7.21 Advanced Semantic Discovery for IoT objects and Networks

**Participants:** Luigi Liquori, Marie-Agnès Peraldi Frati, Abdul Qadir Khan, Oleksii Khramov, Behrad Shirmard.

The activity proposed is the study and development of Semantic Discovery and Query capabilities for oneM2M and its contribution to the oneM2M standard. The goal is to enable an easy and efficient discovery of information and a proper interworking with external source/consumers of information (e.g.

a distributed data base in a smart city or in a firm), or to directly search information in the oneM2M system for big data purposes.

oneM2M has currently 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 well scoped and designed (e.g. the lights in a house). When oneM2M is used to discover wide sets of data or unknown sets of data, the functionality is typically integrated by ad hoc applications that are expanding the oneM2M functionality. This means that this core function may be implemented with different flavours and this is not optimal for interworking and interoperability. See [48, 49, 47, 45] and STF 589 and ETSI Labs ASD.

## 7.22 Asynchronous Contact Tracing to catch Covid-19 using IoT objects and IoT networks

**Participant:** Luigi Liquori.

Asynchronous Contact Tracing (ACT) traces the IoT connected object that may have been infected by the Covid-19 virus (or future pandemic viruses). This shifts the paradigm, from searching for a person in the process of infecting another to the tracing of both potential contamination and infections, and leveraging on the combination of the two information, [50].

The scope of this research vein is to standardize the full support of Asynchronous Contact Tracing (ACT) by means of 1) providing some examples of use and deployment of ACT by means of a few explanatory use cases; 2) specifying the ACT method and its interaction with deployed contact tracing applications for human and systems. This includes the interaction with the different technologies used by non ACT contact tracing solutions; 3) specifying the ACT system including application protocols and API providing a reference implementation using Telecom Italia Mobile oneM2M/ICON platform [46].

The new ACT method will require the use of existing ready-to-market IoT-based technology and well-established wireless network techniques, in particular the ones specified in the ETSI standards ecosystem. Moreover, it will preserve the user's privacy in accordance with GDPR and/or other regional requirements not requiring the transmission of any personal information by the user.

## 7.23 Interferences between parallel embedded applications

**Participants:** Robert de Simone, Julien Deantoni, Dumitru Potop Butucaru.

Model-Based embedded System Engineering can provide an abstract modeling level where to study the important issue of mapping various concurrent applications (often parts of the same general design) onto a selected platform architecture for execution, at a higher level than target code. As the time complexity of communications and data transfers now exceeds that of actual computations, the question of efficient sharing of interconnects and intermediate memory levels for these assemblies of applications becomes crucial, as interferences lead to performance bottlenecks. The focus of the IRT Saint-Exupery project Archeocs is specifically to consider the importance of these phenomena in real-life applications of industrial partners. We worked closely with the Renault Software Labs and Thales Alenia Space divisions in Sophia-Antipolis to exhibit relevant models and formalize their needs and capacities in this respect. This will result soon in a contractual agreement establishing our collaboration inside the Archeocs project.

## 8 Bilateral contracts and grants with industry

**Participants:** Robert de Simone, Frédéric Mallet, Julien Deantoni, Dumitru Potop Butucaru, Marie-Agnès Peraldi Frati.

## 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 7.10. A CIFRE PhD is funded by Renault Software Factory on related topics.

**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 7.14.

**IRT Saint-Exupéry.** 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.

WE are due to start a new collaborative project in similar context, named ARCHEOCS, on the development of formal development methods for optimized mapping (allocation + scheduling) in embedded system engineering flows.

**Renault Software Labs.** 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.

**ETSI STF.** The STF (Specialist Task Force) is funded by ETSI (2020-2021) and integrated within the Work Programme of the Technical Committee SmartM2M (TC SmartM2M).

We received support to conduct the specification and proof-of-concept of Advanced Semantic Resource Discovery standard during the course of its development and adoption. The activity proposed is the study and development of semantic Discovery and Query capabilities for oneM2M and its contribution to the oneM2M standard.

The goal is to enable an easy and efficient discovery of information and a proper interworking with external source/consumers of information (e.g. a distributed data base in a smart city or in a firm), or to directly search information in the oneM2M system for big data purposes.

**ETSI TTF.** The TTF (Testing Task Force) is funded by ETSI (2022-2024) and integrated within the Work Programme of the Technical Committee SmartM2M (TC SmartM2M).

We received support to conduct the Performance Evaluation and Analysis for oneM2M Planning and Deployment. oneM2M, the global standard initiative for M2M communications and the IoT, has published several releases. Each release has added new and advanced features for the oneM2M standard in several aspects: connectivity, interoperability, data management, security, complex architecture, etc.

The oneM2M standard is now mature and multiple deployments exist all over the world at both experimental and operational levels. The experimental deployments are conducted for multiple

reasons: to evaluate the capabilities of the standard in terms of expressiveness, usability on specific equipment, connection with specific existing systems or performance evaluation.

## 8.2 Bilateral grants with industry

**Krono-Safe CIFRE collaboration.** The PhD of Fabien Siron, reported in 7.16, came with a grants from this company. Joint efforts extend partly to the objectives of the Archeocs IRT project 7.23.

# 9 Partnerships and cooperations

**Participants:** Frédéric Mallet, Enlin Zhu, Robert de Simone, François Revest.

## 9.1 European initiatives

### 9.1.1 Horizon Europe

**H2020 project ENACT** The ENACT H2020 project was a research and innovation action project. The project was led by SINTEF with Nicolas Ferry as technical manager. The project was launched in January 2018, receiving around €5 million in EU funding, and concluded in April 2021. The ENACT consortium investigated the main challenges preventing the adoption of DevOps for trustworthy smart IoT systems and developed new techniques in order to address such challenges. The main result is the ENACT DevOps framework for such systems, comprising 10 loosely coupled open-source software tools, the ENACT enablers, facilitating IoT application providers in development, operation, and quality assurance of secure IoT systems. At the final review, the project was rated excellent and is listed as a **success story** by the EU.

## 9.2 National initiatives

**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).

**Competitvity 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 most prominent actions is to build, in partnership with Aix-Marseille University, 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).

**PIA ES3CAP.** ES3CAP (Embedded Smart Safe Secure Computing Autonomous Platform) is a PIA (Programme d'Investissements d'Avenir) project. Its budget is of 22.2MEuros (end: 02/2022). 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 7.14. This project partially finances Hugo Pompougnac's PhD and Jad Khatib's post-doc.

**Grand Défi - Confiace.AI.** We participate to the Confiace.AI program, in the project "Embedded AI", which currently finances the PhD of Mikhail Ilmenskii.

Confiace.AI is the technological pillar of the Grand Défi "Securing, certifying and enhancing the reliability of systems based on artificial intelligence" launched by the Innovation Council. It is the largest technological research programme in the AIforHumanity plan, which is designed to make France one of the leading countries in artificial intelligence (AI).

Our technical contributions to this project are described in 7.15.

**PSPC ADAVEC.** This cooperation, headed for academic partners UCA and Inria by Amar Bouali, and including in the partnership the local Sophia companies EpicnPoc and Avisto, with the indirect support of Renault Software Labs, funded the IE position of François Revest for two years, and soon the post-doc position of Ankica Baresic. Technical advances are reported in 7.2.

## 10 Dissemination

**Participants:** Kairos permanent members.

### 10.1 Promoting scientific activities

#### 10.1.1 Scientific events: organisation

**General chair, scientific chair, steering, working groups.**

- Julien Deantoni was general chair of the 24th IEEE Forum on specification and Design Languages (FDL 2021).
- Robert de Simone is member of the Steering Committee for EmSoft (International Conference on Embedded Software).
- Luigi Liquori is member of [IFIP Working Group 1.6 Rewriting](#).
- Luigi Liquori is member of the Steering Board of the leading conference [FSCD](#), International Conference on Formal Structures for Computation and Deduction.

**Member of the organizing committees.**

- Julien Deantoni was co-chair of the MPM4CPS workshop 2021 (co-located with Models) organization committee.
- Nicolas Ferry was co-chair of the MESS workshop 2021 (co-located with STAF) organization committee.
- Luigi Liquori is in the organizing committee of [Computational Logic Autumn Summit 2022](#) (8 conferences and 4 schools).

**10.1.2 Scientific events: selection****Chair of conference program committees.**

- Frédéric Mallet was topic chair for DATE 2021, topic D1.

**Member of the conference program committees.**

- Julien Deantoni was TPC member of the following conferences and workshops: Modeling Language Engineering and Execution, Multi Paradigm Modeling for Cyber Physical Systems, Formal Co-Simulation of Cyber-Physical Systems, Specification and Design Language.
- Frédéric Mallet was TPC member of DATE, FDL, TASE, ModelsWard, FormReq.
- Nicolas Ferry was TPC member of SEAA 2021, 1stSwForumWs'21, DevOps@MODELS'21.
- Robert de Simone was TPC member for FDL, and Memocode.

**10.1.3 Journal****Member of the editorial boards**

- Julien Deantoni is editor of the ACM Transaction on Embedded Computing Systems special issue on Specification and Design Languages.
- Frédéric Mallet is member of the editorial board of Springer journal Software-Intensive Cyber-Physical Systems (SICS).
- Frédéric Mallet has edited a special issue for Elsevier Science of Computer Programming.
- Nicolas Ferry has edited a special issue on Model-Driven Engineering for IoT for the Springer journal on Software and System Modelling.

**Reviewer - reviewing activities**

- Julien Deantoni was reviewer for the followings journals: ACM Transaction on Embedded Computing Systems, Springer System and Software Modeling.
- Frédéric Mallet was reviewer for ACM Transaction on Software Engineering and Methodology (TOSEM), Elsevier Science of Computer Programming (SCP), Elsevier Journal of Systems Architecture (JSA).
- Nicolas Ferry was reviewer for the following journals: Springer journal on Software and System Modelling, Elsevier Journal of Software Systems.



#### 10.1.4 Invited talks

- Julien Deantoni has given two talks during a national event (namely GDR GPL): *CoSim20: un environnement pour la simulation collaborative et distribuée* and *Model Based co-simulation*.
- Luigi Liquori, during the *ETSI IoT Week 2021*, has given an invited talk and has participated to the panel discussion on the session *IoT in the Face of the Pandemic: Digitalization and Countermeasures*. He presented the standard *Asynchronous Contact Tracing (ACT)*. Video on [Vimeo](#).

#### 10.1.5 Scientific expertise

- Robert de Simone is member of the ANR selection jury for CE25 programme.
- Frédéric Mallet is a member of the National Committee for Universities since 2019. He has been reviewer for several projects (ANR, ANRT, FNRS, NSFC), and visiting committee member of two joint units for HCERES (LabSTICC and LIUPPA).
- Nicolas Ferry is member of the SODALITE H2020 project advisory board.
- Luigi Liquori is member of *CA COST Action 20111. European Research Network on Formal Proofs (EuroProofNet)*, WG 3 (program verification) and WG 6 (Type Theory).
- Luigi Liquori and Marie-Agnès Peraldi Frati are members of *ETSI Standardization Technical Committee ETSI/SmartM2M* - Smart Machine- to-Machine Communications et oneM2M Standards for M2M and the Internet of Things.
- Luigi Liquori is member of the *ETSI Standardization Technical Committee ETSI/eHEALTH* Technical standards-based market for health.

#### 10.1.6 Research administration

- Frédéric Mallet is deputy director of I3S Laboratory (Joint Unit between UCA and CNRS).
- Julien Deantoni is principal investigator of the bilateral GLOSE project in collaboration with Safran.
- Nicolas Ferry was the technical manager of the ENACT H2020 project (final review in May 2021).

## 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, M1, Université Côte d'Azur.
- International Master: Frédéric Mallet, Software Engineering, 32h, M1, Université Côte d'Azur.
- Master: Frédéric Mallet, Programmation Synchrones, 32h, M1, Université Côte d'Azur.
- Master: Robert de Simone, Formal Model-Based Design for Cyber-Physical Systems and IoT, 36 heures eq TD, M2 International Ubinet, Université Côte d'Azur.

- Licence: Marie-Agnès Peraldi Frati, Web security (20h eq TD), Security of connected objects (20h eq TD), IoT Infrastructure deployment (20 H) and Large scale platform for IoT (20h eq TD), in a licence cursus dedicated to Internet of Objects, Infrastructure and Applications, Université Côte d'Azur.
- Master: Marie-Agnès Peraldi Frati, Web Security and IoT Platform, 20h eq TD, Master 2 SICOM, Univ Avignon.
- Master: Luigi Liquori, Peer-to-peer systems, 32 eq TD, M2 CASPAR, Polytech Nice Sophia, Université Côte d'Azur.
- Master: Julien Deantoni, Finite State Machine, 54h eq TD, Polytech Nice Sophia, Université Côte d'Azur.
- Master: Julien Deantoni, Multi Paradigm Programming in C++, 54h eq TD, Polytech Nice Sophia, Université Côte d'Azur.
- Master: Julien Deantoni, Domain Specific Languages, 32h eq TD, Polytech Nice Sophia, Université Côte d'Azur.
- Licence: Julien Deantoni, Language and Compilation, 32h eq TD, L3IA.
- Master: Julien Deantoni, Micro-controller programming, 8h eq TD, Polytech Nice Sophia, Université Côte d'Azur.
- 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.
- Licence: Nicolas Ferry, Web programming for mobile devices (50h eq TD), Project Management and continuous delivery (25h eq TD), in a licence cursus dedicated to the development of mobile applications, Université Côte d'Azur.
- Licence: Nicolas Ferry, Software architecture (25h eq TD) and Programming Web Services (20h eq TD), in a licence cursus dedicated to Internet of Objects, Infrastructure and Applications, Université Côte d'Azur.
- Master: Nicolas Ferry, Web services for the Internet of Things (4h eq TD), M2 International Ubinet, Université Côte d'Azur.
- License: Luc Hogie, Distributed programming, 28h eq TD, DUT Informatique.

### 10.2.2 Supervision

- PhD defense: Giovanni Liboni, Co-simulation de systemes complexes avec le framework CoSim20, Université Côte d'Azur, CIFRE Safran, Apr. 21 2021, Dir: Frédéric Mallet, Adv: Julien DeAntoni.
- PhD in progress: Joao Cambeiro, Seamless Integration of Heterogeneous Multi Fidelity Models into a Systems Engineering Approach, Université Côte d'Azur, ends 2023, Adv: Julien DeAntoni.
- PhD in Progress: Hui Zhao, A Modeling LanguageBased MultiView Approach For Cyber Physical Systems, Université Côte d'Azur, ends 2022, Adv: Frédéric Mallet.
- PhD in Progress: Joelle Abou-Faysal, Safety Rules for Autonomous Vehicles, Université Cote d'Azur, ends 2022, CIFRE Renault, Adv: Frédéric Mallet.
- PhD in Progress: Enlin Zhu, Safe Smart Contracts with temporal constraints, Université Côte d'Azur, ends 2024, Adv: Frédéric Mallet.

- PhD in Progress: Pavlo Tokariev, Abstract Interpretation for Logical Time Constraints, Université Côte d'Azur, starts 2021, Adv: Frédéric Mallet.
- PhD in Progress: Arseniy Gromovoy, Efficient statistical model-checking of Multiform Logical time constraints, Université Côte d'Azur, starts 2022, Adv: Robert de Simone.
- PhD in Progress: Mansur Khazeev, Safe and Self-extending Smart Contracts Languages, Université Côte d'Azur, ends 2023, Adv: Luigi Liquori.

### 10.2.3 Juries

- Frédéric Mallet was PhD Jury President: Halchin Alexandra, ENSEEIHT, Toulouse-INP, Dec. 3rd 2021.
- Frédéric Mallet was PhD Jury President: Gharbi Amna, Institut Polytechnique de Paris, Nov. 10th 2021.
- Robert de Simone was PhD Reviewer: Breytner Joseph Fernandez Mesa, IMAG Grenoble, Oct. 2021.

## 10.3 Popularization

### 10.3.1 Education

- Frédéric Mallet is Head of the "Département Disciplinaire d'Informatique" since March 2020.

### 10.3.2 Interventions

- Julien Deantoni is part of the "cordées de la réussite" programme, where he does appearances in the Emile Roux school (le Cannet) to give 12 year-old kids courses on robotics.

## 11 Scientific production

### 11.1 Major publications

- [1] C. André, J. Deantoni, F. Mallet and 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*. Ed. by S. K. Shukla and J.-P. Talpin. Chapter 7. Springer Science+Business Media, LLC 2010, July 2010, p. 28. URL: <https://hal.inria.fr/inria-00495664>.
- [2] Y. Bao, M. Chen, Q. Zhu, T. Wei, F. Mallet and 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* 36.12 (Dec. 2017), pp. 1989–2002. DOI: [10.1109/TCAD.2017.2681076](https://doi.org/10.1109/TCAD.2017.2681076). URL: <https://hal.inria.fr/hal-01644285>.
- [3] T. Carle, D. Potop-Butucaru, Y. Sorel and D. Lesens. 'From Dataflow Specification to Multiprocessor Partitioned Time-triggered Real-time Implementation \*'. In: *Leibniz Transactions on Embedded Systems* (Nov. 2015). DOI: [10.4230/LITES-v002-i002-a001](https://doi.org/10.4230/LITES-v002-i002-a001). URL: <https://hal.inria.fr/hal-01263994>.
- [4] A. Ciaffaglione, P. Di Gianantonio, F. Honsell and L. Liquori. 'A prototype-based approach to object evolution'. In: *The Journal of Object Technology* 20.4 (1st Dec. 2020), pp. 1–24. URL: <https://hal.inria.fr/hal-01646168>.
- [5] L. Henrio, E. Madelaine and M. Zhang. 'A Theory for the Composition of Concurrent Processes'. In: *36th International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE)*. Ed. by E. Albert and I. Lanese. Vol. LNCS-9688. Formal Techniques for Distributed Objects, Components, and Systems. Heraklion, Greece, 2016, pp. 175–194. DOI: [10.1007/978-3-319-39570-8\\_12](https://doi.org/10.1007/978-3-319-39570-8_12). URL: <https://hal.inria.fr/hal-01432917>.

- [6] F. Honsell, L. Liquori, P. Maksimovic and I. Scagnetto. ‘LLFP : A Logical Framework for modeling External Evidence, Side Conditions, and Proof Irrelevance using Monads’. In: *Logical Methods in Computer Science*. Special Issue in honor of Pierre Louis Curien (23rd Feb. 2017). URL: <https://hal.inria.fr/hal-01146059>.
- [7] F. Honsell, L. Liquori, C. Stolze and I. Scagnetto. ‘The Delta-framework’. In: 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, (FSTTCS) 2018. Vol. 122. 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS. Ahmedabad, India, 2018, 37:1–37:21. DOI: [10.4230/LIPIcs.FSTTCS.2018.37](https://doi.org/10.4230/LIPIcs.FSTTCS.2018.37). URL: <https://hal.archives-ouvertes.fr/hal-01701934>.
- [8] F. Jebali and D. P. 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. DOI: [10.1109/ACSD.2018.00019](https://doi.org/10.1109/ACSD.2018.00019). URL: <https://doi.ieeecomputersociety.org/10.1109/ACSD.2018.00019>.
- [9] L. Liquori and C. Stolze. ‘The Delta-calculus: syntax and types’. In: FSCD 2019 - 4th International Conference on Formal Structures for Computation and Deduction. Dortmund, Germany, 2019. DOI: [10.1007/978-3-030-44411-2\\_6](https://doi.org/10.1007/978-3-030-44411-2_6). URL: <https://hal.archives-ouvertes.fr/hal-01963662>.
- [10] L. Liquori, C. Tedeschi, L. Vanni, F. Bongiovanni, V. Ciancaglini and 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*. Ed. by M. Crovella, L. M. Feeney, D. Rubenstein and S. V. Raghavan. Vol. 6091. Lecture Notes in Computer Science. Chennai, India: Springer Verlag, May 2010, pp. 67–82. DOI: [10.1007/978-3-642-12963-6\\_6](https://doi.org/10.1007/978-3-642-12963-6_6). URL: <https://hal.inria.fr/hal-00909544>.
- [11] L. Liquori, S. Wood and E. Scarrone. *Asynchronous Contact Tracing*. 14th Dec. 2020. URL: <https://hal.inria.fr/hal-02989404>.
- [12] F. Mallet and R. De Simone. ‘Correctness issues on MARTE/CCSL constraints’. In: *Science of Computer Programming* 106 (Aug. 2015), pp. 78–92. DOI: [10.1016/j.scico.2015.03.001](https://doi.org/10.1016/j.scico.2015.03.001). URL: <https://hal.inria.fr/hal-01257978>.
- [13] J.-V. Millo and R. De Simone. ‘Periodic scheduling of marked graphs using balanced binary words’. In: *Theoretical Computer Science* 458.2 (Nov. 2012), pp. 113–130. DOI: [10.1016/j.tcs.2012.08.012](https://doi.org/10.1016/j.tcs.2012.08.012). URL: <https://hal.inria.fr/hal-00764076>.
- [14] G. Ngo Hoang, L. Liquori and 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*. Vol. 8314. Lecture Notes in Computer Science. Coimbatore, India: Springer Verlag, Jan. 2014, pp. 287–301. DOI: [10.1007/978-3-642-45249-9\\_19](https://doi.org/10.1007/978-3-642-45249-9_19). URL: <https://hal.inria.fr/hal-00906798>.
- [15] D. Potop-Butucaru, R. De Simone and J.-P. Talpin. ‘Synchronous hypothesis and polychronous languages’. In: *Embedded Systems Design and Verification*. Ed. by R. Zurawski. CRC Press, 2009, pp. 6-1-6–27. DOI: [10.1201/9781439807637.ch6](https://doi.org/10.1201/9781439807637.ch6). URL: <https://hal.inria.fr/hal-00788473>.
- [16] C. Stolze and L. Liquori. ‘A Type Checker for a Logical Framework with Union and Intersection Types’. In: FSCD 2020 - 5th International Conference on Formal Structures for Computation and Deduction. Paris, France, 2020. DOI: [10.4230/LIPIcs.FSCD.2020](https://doi.org/10.4230/LIPIcs.FSCD.2020). URL: <https://hal.archives-ouvertes.fr/hal-02573605>.
- [17] M. Zhang, F. Dai and F. Mallet. ‘Periodic scheduling for MARTE/CCSL: Theory and practice’. In: *Science of Computer Programming* 154 (Mar. 2018), pp. 42–60. DOI: [10.1016/j.scico.2017.08.015](https://doi.org/10.1016/j.scico.2017.08.015). URL: <https://hal.inria.fr/hal-01670450>.

## 11.2 Publications of the year

### International journals

- [18] A. Barisic, I. Ruchkin, D. Savić, M. Abshir Mohamed, R. Al-Ali, L. W. Li, H. Mkaouar, R. Eslampanah, M. Challenger, D. Blouin, O. Nikiforova and A. Cicchetti. ‘Multi-Paradigm Modeling for Cyber-Physical Systems: A Systematic Mapping Review’. In: *Journal of Systems and Software* (2021). DOI: [10.1016/j.jss.2021.111081](https://doi.org/10.1016/j.jss.2021.111081). URL: <https://hal.archives-ouvertes.fr/hal-03338594>.
- [19] J. Holtmann, J. Deantoni and M. Fockel. ‘Early timing analysis based on scenario requirements and platform models’. In: *Software and Systems Modeling* (2022). URL: <https://hal.inria.fr/hal-03375049>.
- [20] T. Rajmohan, P. Nguyen and N. Ferry. ‘A decade of research on patterns and architectures for IoT security’. In: *Journal of Cybersecurity* 5.1 (Jan. 2022). DOI: [10.1186/s42400-021-00104-7](https://doi.org/10.1186/s42400-021-00104-7). URL: <https://hal.archives-ouvertes.fr/hal-03511699>.
- [21] Y. Zhang, F. Mallet, H. Zhu, Y. Chen, B. Liu and Z. Liu. ‘A clock-based dynamic logic for schedulability analysis of CCSL specifications’. In: *Science of Computer Programming* 202 (Feb. 2021), p. 102546. DOI: [10.1016/j.scico.2020.102546](https://doi.org/10.1016/j.scico.2020.102546). URL: <https://hal.inria.fr/hal-03135429>.
- [22] Y. Zhang, H. Wu, Y. Chen and F. Mallet. ‘A clock-based dynamic logic for the verification of CCSL specifications in synchronous systems’. In: *Science of Computer Programming* 203 (Mar. 2021), p. 102591. DOI: [10.1016/j.scico.2020.102591](https://doi.org/10.1016/j.scico.2020.102591). URL: <https://hal.inria.fr/hal-03135428>.

### International peer-reviewed conferences

- [23] C. Bruns and S. Touati. ‘Understanding microarchitectural effects on the performance of parallel applications’. In: HPCS 2020 - 18th International Conference on High Performance Computing & Simulation. Virtual, Spain, 22nd Mar. 2021. URL: <https://hal.inria.fr/hal-03195799>.
- [24] J. Cambeiro, J. Deantoni and V. Amaral. ‘Supporting the Engineering of Multi-Fidelity Simulation Units With Simulation Goals’. In: International Workshop on Multi-Paradigm Modeling for Cyber-Physical Systems (MPM4CPS’21). Fukuoka, Japan, 10th Oct. 2021. URL: <https://hal.inria.fr/hal-03374990>.
- [25] J. Deantoni, J. Cambeiro, S. Bateni, S. Lin and M. Lohstroh. ‘Debugging and Verification Tools for LINGUA FRANCA in GEMOC Studio’. In: FDL 2021 - Forum on specification & Design Languages. Antibes, France, 8th Sept. 2021. DOI: [10.1109/FDL53530.2021.9568383](https://doi.org/10.1109/FDL53530.2021.9568383). URL: <https://hal.inria.fr/hal-03374955>.
- [26] M. Hu, J. Ding, M. Zhang, F. Mallet and M. Chen. ‘Enumeration and Deduction Driven Co-Synthesis of CCSL Specifications Using Reinforcement Learning’. In: 2021 IEEE Real-Time Systems Symposium (RTSS). Dortmund / Virtual, Germany: IEEE, 7th Dec. 2021, pp. 227–239. DOI: [10.1109/RTSS52674.2021.00030](https://doi.org/10.1109/RTSS52674.2021.00030). URL: <https://hal.inria.fr/hal-03525306>.
- [27] R. Karaki, L. Marti and J. Deantoni. ‘WIP: Domain Specific Debugging by using RUNSTAR’. In: FDL 2021 - Forum on specification & Design Languages. Antibes, France, 8th Sept. 2021. URL: <https://hal.inria.fr/hal-03375006>.

### National peer-reviewed Conferences

- [28] F. Siron, D. Potop-Butucaru, R. De Simone, D. Chabrol and A. Methni. ‘Vérification d’applications temps-réel basées sur le paradigme de Logical Execution Time (LET)’. In: École d’Été Temps Réel 2021. Vol. [https://etr2021.ensma.fr/files/actes\\_phdstudents\\_etr2021.pdf](https://etr2021.ensma.fr/files/actes_phdstudents_etr2021.pdf). ETR 2021 13. Poitiers, France, 20th Sept. 2021. URL: <https://hal.inria.fr/hal-03545758>.

### Conferences without proceedings

- [29] J. Abou Faysal, N. Zalmi, A. Barisic and F. Mallet. ‘EPSAAV: An Extensible Platform for Safety Analysis of Autonomous Vehicles’. In: MEDI 2021 - 10th International Conference on Model and Data Engineering. Tallinn, Estonia, 21st June 2021. DOI: [10.1007/978-3-030-87657-9\\_8](https://doi.org/10.1007/978-3-030-87657-9_8). URL: <https://hal.archives-ouvertes.fr/hal-03331190>.
- [30] R. Dautov, H. Song and N. Ferry. ‘Towards a Sustainable IoT with Last-Mile Software Deployment’. In: ISCC 2021 - IEEE Symposium on Computers and Communications. Athens, Greece: IEEE, 5th Sept. 2021, pp. 1–6. DOI: [10.1109/ISCC53001.2021.9631250](https://doi.org/10.1109/ISCC53001.2021.9631250). URL: <https://hal.archives-ouvertes.fr/hal-03502839>.
- [31] F. Siron, D. Potop-Butucaru, R. De Simone, D. Chabrol and A. Methni. ‘Programming and verifying real-time design using logical time’. In: FDL 2021 - Forum on specification & Design Languages. Antibes, France, 8th Sept. 2021. URL: <https://hal.inria.fr/hal-03537976>.

### Scientific books

- [32] N. Ferry, H. Song, A. Metzger and E. Rios. *DevOps for Trustworthy Smart IoT Systems*. Now Publishers, 2021. DOI: [10.1561/9781680838251](https://doi.org/10.1561/9781680838251). URL: <https://hal.inria.fr/hal-03281420>.

### Scientific book chapters

- [33] N. Ferry, H. Song, R. Dautov, P. Nguyen and F. Chauvel. ‘Model-based Continuous Deployment of SIS’. In: *DevOps for Trustworthy Smart IoT Systems*. Now Publishers, 2021. DOI: [10.1561/9781680838251.ch4](https://doi.org/10.1561/9781680838251.ch4). URL: <https://hal.inria.fr/hal-03281414>.
- [34] N. Ferry, H. Song, E. Rios and A. Metzger. ‘The ENACT approach’. In: *DevOps for Trustworthy Smart IoT Systems*. Now Publishers, 6th July 2021. DOI: [10.1561/9781680838251.ch2](https://doi.org/10.1561/9781680838251.ch2). URL: <https://hal.inria.fr/hal-03281377>.
- [35] E. Rios, N. Ferry, H. Song and A. Metzger. ‘Introduction’. In: *DevOps for Trustworthy Smart IoT Systems*. Now Publishers, 2021. DOI: [10.1561/9781680838251.ch1](https://doi.org/10.1561/9781680838251.ch1). URL: <https://hal.inria.fr/hal-03281403>.

### Doctoral dissertations and habilitation theses

- [36] G. Liboni. ‘Complex systems co-simulation with the CoSim20 framework : for efficient and accurate distributed co-simulations’. Université Côte d’Azur, 21st Apr. 2021. URL: <https://tel.archives-ouvertes.fr/tel-03382774>.

### Reports & preprints

- [37] R. Ameur-Boulifa, L. Henrio and E. Madelaine. *Compositional equivalences based on open pNets*. 8th Jan. 2021. URL: <https://hal.inria.fr/hal-03103607>.
- [38] A. Barisic, A. Cicchetti, I. Ruchkin and D. Blouin. *Literature Classification Data for a Systematic Mapping Study on Multi-Paradigm Modeling for Cyber-Physical Systems*. 2021. URL: <https://hal.archives-ouvertes.fr/hal-03338607>.
- [39] S. Ghilezan, S. Kašterović, L. Liquori, B. Marinković, Z. Ognjanović and T. Stefanović. *Federating Digital Contact Tracing using Structured Overlay Networks*. 13th Oct. 2021. URL: <https://hal.inria.fr/hal-03127890>.
- [40] L. Hogie. *Idawi: a middleware for distributed applications in the IOT, the fog and other multihop dynamic networks*. CNRS - Centre National de la Recherche Scientifique; Université Côte d’azur; Inria, 8th Feb. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03562184>.
- [41] D. Potop-Butucaru, J. Khatib and P. Baufreton. *Bounding memory access interferences on the Kalray MPPA3 compute cluster*. RR-9404. Inria, Apr. 2021, p. 24. URL: <https://hal.inria.fr/hal-03207510>.

- [42] B. Wang, E. Madelaine and M. Zhang. *Symbolic Weak Equivalences: Extension, Algorithms, and Minimization - Extended version*. RR-9389. Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France; East China Normal University (Shanghai), 30th Jan. 2021, p. 71. URL: <https://hal.inria.fr/hal-03126313>.

### Other scientific publications

- [43] J. Abou Faysal, N. Zalmi, A. Barisic and F. Mallet. 'An Approach to Ensure Safety of Autonomous Vehicles in Planned Trajectories'. In: FDL 2021 Forum on specification & Design Languages. ANTIBES, France, 8th Sept. 2021. URL: <https://hal.archives-ouvertes.fr/hal-03331199>.
- [44] A. Barisic, E. Zhu and F. Mallet. 'Model-driven approach for the design of multi-chain smart contracts'. In: 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS). Paris, France, 27th Sept. 2021. DOI: [10.1109/BRAINS52497.2021.9569809](https://doi.org/10.1109/BRAINS52497.2021.9569809). URL: <https://hal.archives-ouvertes.fr/hal-03338936>.
- [45] S. M. Jeong, S. Kumar, A. Cimmino, R. García Castro, L. Liquori, M.-A. Peraldi-Frati, E. Scarrone, J. Koss and F. Bob. *ETSI SmartM2M Technical Report 103717; Study for oneM2M; Discovery and Query specification development*. 1st June 2021. URL: <https://hal.inria.fr/hal-03261080>.
- [46] A. Q. Khan. 'Asynchronous Contact Tracing, Fighting Pandemics with Internet of Things'. INRIA Sophia Antipolis - I3S, 15th Sept. 2021. URL: <https://hal.inria.fr/hal-03410027>.
- [47] L. Liquori, M.-A. Peraldi-Frati, A. Cimmino, R. García Castro, A. Q. Khan, S. El Khatab, O. Khranov, E. Scarrone and J. Koss. *ETSI SmartM2M Technical Report 103716; oneM2M Discovery and Query solution(s) simulation and performance evaluation*. 1st June 2021. URL: <https://hal.inria.fr/hal-03261059>.
- [48] L. Liquori, E. Scarrone, M.-A. Peraldi-Frati, A. Cimmino, R. García Castro, J. Koss and S. M. Jeong. *ETSI SmartM2M Technical Report 103714: Study for oneM2M Discovery and Query use cases and requirements*. 19th Jan. 2021. URL: <https://hal.inria.fr/hal-03115482>.
- [49] L. Liquori, E. Scarrone, M.-A. Peraldi-Frati, S. M. Jeong, A. Cimmino, R. García Castro, J. Koss, A. Q. Khan, S. Kumar and S. El Khatab. *ETSI SmartM2M Technical Report 103715; Study for oneM2M; Discovery and Query solutions analysis & selection*. 19th Jan. 2021. URL: <https://hal.inria.fr/hal-03115497>.
- [50] L. Liquori, E. Scarrone, S. Wood, L. Cees, F. daSilva, M. Maass, F. Bob, T. Kessler, H. Taras and M. Vanetti. *ETSI Technical Specification TS 103757. SmartM2M; Asynchronous Contact Tracing System: Fighting pandemic disease with Internet of Things (IoT)*. 18th Dec. 2021. URL: <https://hal.inria.fr/hal-02989793>.

## 11.3 Other

### Softwares

- [51] [SW] M. Belaoucha, C. Alias, D. Barthou and S. Touati, *FADAlib: an open source C++ library for fuzzy array dataflow analysis*, 25th Nov. 2021. LIC: GNU Lesser General Public License v3.0 or later. HAL: [hal-03445991](https://hal.archives-ouvertes.fr/hal-03445991), URL: <https://hal.archives-ouvertes.fr/hal-03445991>, SWHID: [swh:1:dir:fc7481ee438316b9ce5b273ca894114bf658d3d9;origin=https://hal.archives-ouvertes.fr/hal-03445991;visit=swh:1:snp:518f2d28a2d2a1ad15ee2f630b40be3e24a0f8b1;anchor=swh:1:rel:488f5aa5aaa21fc92f24f0f7c9b571e56f1325ec;path=/](https://sw.hal.archives-ouvertes.fr/hal-03445991;visit=swh:1:snp:518f2d28a2d2a1ad15ee2f630b40be3e24a0f8b1;anchor=swh:1:rel:488f5aa5aaa21fc92f24f0f7c9b571e56f1325ec;path=/).
- [52] [SW] S. Briais and S. Touati, *A C-library for Register Pressure Reduction in Data Dependence Graphs*. 25th Nov. 2021. LIC: GNU General Public License v3.0 or later. HAL: [hal-03445804](https://hal.archives-ouvertes.fr/hal-03445804), URL: <https://hal.archives-ouvertes.fr/hal-03445804>, SWHID: [swh:1:dir:0b9ea5cf3f444bf0920d99d5df58af2b6b484133;origin=https://hal.archives-ouvertes.fr/hal-03445804;visit=swh:1:snp:dc58dc9c3866979847adb3ee5202b8e14ac7b9cd;anchor=swh:1:rel:b0827bb530d7e2258121543e63e06de4b2c640c0;path=/](https://sw.hal.archives-ouvertes.fr/hal-03445804;visit=swh:1:snp:dc58dc9c3866979847adb3ee5202b8e14ac7b9cd;anchor=swh:1:rel:b0827bb530d7e2258121543e63e06de4b2c640c0;path=/).

- [53] [SW] S. Briais and S. Touati, *A C-library for Register Saturation Computation*, 25th Nov. 2021. LIC: GNU General Public License v3.0 or later. HAL: [hal-03445821](https://hal.archives-ouvertes.fr/hal-03445821), URL: <https://hal.archives-ouvertes.fr/hal-03445821>, SWHID: [swh:1:dir:353f14fe73608aeb968e1b7d961293c4292bcc91;origin=https://hal.archives-ouvertes.fr/hal-03445821;visit=swh:1:snp:f19987035256ae2536dabfadf31f38afa689e5d7;anchor=swh:1:rel:20809d0c43faec53646297b5271a12e9beec8f24;path=/](https://sw.hal.science/swh:1:dir:353f14fe73608aeb968e1b7d961293c4292bcc91;origin=https://hal.archives-ouvertes.fr/hal-03445821;visit=swh:1:snp:f19987035256ae2536dabfadf31f38afa689e5d7;anchor=swh:1:rel:20809d0c43faec53646297b5271a12e9beec8f24;path=/).
- [54] [SW] S. Touati, *A C++ implementation of a  $O(n \log n)$  algorithm computing a maximal stable set of an interval graph*, 25th Nov. 2021. LIC: General Public License v3.0 or later. HAL: [hal-03445937](https://hal.archives-ouvertes.fr/hal-03445937), URL: <https://hal.archives-ouvertes.fr/hal-03445937>, SWHID: [swh:1:dir:95986a410683154bc0c9d55a2331839b74993c3d;origin=https://hal.archives-ouvertes.fr/hal-03445937;visit=swh:1:snp:63bdc74d56c678b7fa433ed87f0a4654050d02a3;anchor=swh:1:rel:9906213d31fdd2ce4d04d281a9b47747038836f1;path=/](https://sw.hal.science/swh:1:dir:95986a410683154bc0c9d55a2331839b74993c3d;origin=https://hal.archives-ouvertes.fr/hal-03445937;visit=swh:1:snp:63bdc74d56c678b7fa433ed87f0a4654050d02a3;anchor=swh:1:rel:9906213d31fdd2ce4d04d281a9b47747038836f1;path=/).
- [55] [SW] S. Touati, *DDG: A C++ High Level Data Dependence Graph Library*, 25th Nov. 2021. Université de Versailles Saint Quentin en Yvelines (UVSQ), France. LIC: GNU Lesser General Public License v3 or later. HAL: [hal-03445736](https://hal.archives-ouvertes.fr/hal-03445736), URL: <https://hal.archives-ouvertes.fr/hal-03445736>, SWHID: [swh:1:dir:4f3c8565cd0458811546f15c8cddb45d7b7d5e99;origin=https://hal.archives-ouvertes.fr/hal-03445736;visit=swh:1:snp:02f0329065aa0998f15261721418c6af35ffbf8e;anchor=swh:1:rel:44f40b0bdf1f3004777f705b5beaf0d5b1f8ac43;path=/](https://sw.hal.science/swh:1:dir:4f3c8565cd0458811546f15c8cddb45d7b7d5e99;origin=https://hal.archives-ouvertes.fr/hal-03445736;visit=swh:1:snp:02f0329065aa0998f15261721418c6af35ffbf8e;anchor=swh:1:rel:44f40b0bdf1f3004777f705b5beaf0d5b1f8ac43;path=/).
- [56] [SW] S. Touati, *Dilworth Decomposition in C++ using the LEDA Graph library*, 25th Nov. 2021. LIC: GNU General Public License v3.0 or later. HAL: [hal-03445891](https://hal.archives-ouvertes.fr/hal-03445891), URL: <https://hal.archives-ouvertes.fr/hal-03445891>, SWHID: [swh:1:dir:96814a40df85e15f23dec922f30ed0ce5c3f6336;origin=https://hal.archives-ouvertes.fr/hal-03445891;visit=swh:1:snp:a39dbdfa45e4a236e06b22efb7f9495b7b621416;anchor=swh:1:rel:945df584d6c66ea6973544f084124038c4d0a51f;path=/](https://sw.hal.science/swh:1:dir:96814a40df85e15f23dec922f30ed0ce5c3f6336;origin=https://hal.archives-ouvertes.fr/hal-03445891;visit=swh:1:snp:a39dbdfa45e4a236e06b22efb7f9495b7b621416;anchor=swh:1:rel:945df584d6c66ea6973544f084124038c4d0a51f;path=/).

## 11.4 Cited publications

- [57] G. Liboni, J. Deantoni, A. Portaluri, D. Quaglia and 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, Jan. 2018. URL: <https://hal.inria.fr/hal-01675396>.