

RESEARCH CENTRE

**Inria Center
at Université Côte d'Azur**

IN PARTNERSHIP WITH:
CNRS, Université Côte d'Azur

2022

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

The Inria logo is a stylized, cursive script in red, positioned in the bottom right corner of the page.

Contents

Project-Team KAIROS	1
1 Team members, visitors, external collaborators	2
2 Overall objectives	3
3 Research program	4
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	5
3.5 Code generation and optimization	5
3.6 Extensions for spatio-temporal modeling and mobile systems	6
3.7 IoT standardization	6
3.8 Formal semantics of process algebras underpinning synchronous languages	6
3.9 Logical Frameworks with union and intersection types	7
4 Application domains	7
4.1 Cyber-Physical and embedded system design	7
4.2 Smart Contracts for connected objects in the Internet of Things	7
4.3 Safe driving rules for automated driving	7
4.4 IoT Advanced Semantic Discovery	7
4.5 SARS-CoV2 virus detection using IoT entities	8
5 Social and environmental responsibility	8
5.1 Footprint of research activities	8
6 New software and platforms	8
6.1 New software	8
6.1.1 VerCors	8
6.1.2 TimeSquare	9
6.1.3 GEMOC Studio	9
6.1.4 BCool	10
6.1.5 JMaxGraph	10
6.1.6 Lopht	10
6.1.7 LoPhT-manycore	11
6.1.8 Bull ITP	12
6.1.9 CoSim20	13
6.1.10 Idawi	13
6.1.11 mlirlus	14
6.1.12 ACT	15
6.2 New platforms	15
6.2.1 Advanced Semantic Discovery	15
6.2.2 Asynchronous Contact Tracing	15
7 New results	15
7.1 Decision process for Logical Time	15
7.2 Formalizing and extending Smart Contracts languages with temporal and dynamic features	16
7.3 CCSL extension to stochastic Logical Time	16
7.4 Safety rules for autonomous driving	16
7.5 Take-over between automated and human automotive driving	17
7.6 On the generation of compiler friendly C code from CCSL specification	17
7.7 Continuation of the work on the program recognition with neural networks	17
7.8 Behavioral semantics in Model Driven Engineering	18
7.9 Efficient parallelism in shared memory	18

7.10	A language and compiler for embedded and Real-Time Machine Learning programming	18
7.11	Formal verification of Logical Execution Time applications	19
7.12	Interference analysis in Real-Time task mapping	20
7.13	Trustworthy Fleet deployment and management	20
7.14	Model-Based serverless platform for the Cloud-Edge-IoT continuum	21
7.15	Behavioral equivalence of Open Systems	21
7.16	Standardization and performance evaluation in ETSI oneM2M standard	21
7.17	Advanced Semantic Discovery for IoT objects and networks	22
7.18	Asynchronous Contact Tracing to catch Covid-19 using IoT objects and IoT networks	22
7.19	Federating heterogeneous Digital Contact Tracing platforms	23
7.20	Logical Time modeling and implementation of distributed systems	23
7.21	A middleware for the experimentation on IoT mobile networks	23
8	Bilateral contracts and grants with industry	24
9	Partnerships and cooperations	24
9.1	International initiatives	24
9.1.1	Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program	24
9.2	European initiatives	25
9.2.1	Horizon Europe	25
9.2.2	Other european programs/initiatives	26
9.3	National initiatives	26
9.4	Regional initiatives	27
10	Dissemination	27
10.1	Promoting scientific activities	27
10.1.1	Scientific events: organisation	27
10.1.2	Journal	28
10.1.3	Invited talks	28
10.1.4	Scientific expertise	28
10.1.5	Research administration	28
10.2	Teaching - Supervision - Juries	28
10.2.1	Teaching	28
10.2.2	Supervision	30
10.2.3	Juries	30
10.3	Popularization	30
10.3.1	Intervention	30
11	Scientific production	31
11.1	Major publications	31
11.2	Publications of the year	32
11.3	Cited publications	34

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 COTE D'AZUR, Professor, HDR]
- Nicolas Ferry [UNIV COTE D'AZUR, Associate Professor]
- Frédéric Mallet [UNIV COTE D'AZUR, Professor, HDR]
- Marie-Agnès Peraldi-Frati [UNIV COTE D'AZUR, Associate Professor]
- Sid Touati [UNIV COTE D'AZUR, Professor, HDR]

Post-Doctoral Fellow

- Ankica Barisic [UNIV COTE D'AZUR]

PhD Students

- Joelle Abou Faysal [GROUPE RENAULT, until Mar 2022]
- Baptiste Allorant [Inria, from Oct 2022]
- Joao Cambeiro [UNIV COTE D'AZUR]
- Frédéric Fort [IRT ST EXUPERY, from Sep 2022]
- Arseniy Gromovoy [UNIV COTE D'AZUR]
- Mansur Khazeev [UNIV COTE D'AZUR]
- Maksym Labzhaniia [UNIV COTE D'AZUR, from Sep 2022]
- Hugo Pompougnac [Inria]
- Fabien Siron [KRONO SAFE]
- Pavlo Tokariev [Inria]
- Enlin Zhu [CNRS]

Technical Staff

- Luc Hogie [CNRS]

Interns and Apprentices

- Alessio Di Dio [UNIV COTE D'AZUR and UNIV BOLOGNA, ITALY, from Mar 2022 until Aug 2022]
- Viacheslav Herasymov [UNIV COTE D'AZUR, from Mar 2022 until Jul 2022]
- Maksym Labzhaniia [UNIV COTE D'AZUR, from Mar 2022 until Aug 2022]

Administrative Assistant

- Patricia Riveill [Inria]

External Collaborator

- François Revest [UNIV COTE D'AZUR]

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, see Figure 1.

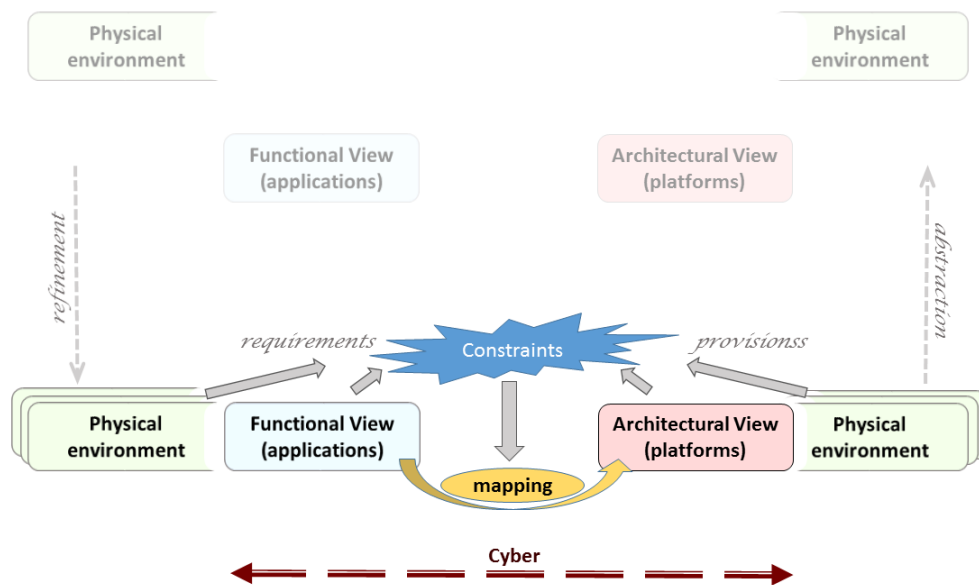


Figure 1: Cyber-Physical generic architectural features

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

Logical Multiform Time. It may be useful to provide an introduction and motivation for the notion of Logical Multiform Time (and Logical Clocks), as they play a central role in our approach to Design. We call Logical Clock any repetitive sequence of occurrences of an event (disregarding possible values carried by the event). It can be regularly linked to physical time (periodic), but not necessarily so: fancy

processors may change speeds, simulation engine change time-integration steps, or much more generally one may react with event-driven triggers of complex logical nature (do this after 3-times that unless this...). It is our belief that user specifications are generally expressed using such notions, with only partial timing correlations between distinct logical clocks, so that the process of realization (or “model-based compilation”) consists for part in establishing (by analysis or abstract simulation) the possible tighter relations between those clocks (unifying them from a partial order of local total orders to a global total order). We have defined in the past a small language of primitives expressing recognized constraints structuring the relations between distinct logical clocks [1, 13]. This language (named CCSL for Clock Constraint Specification Language), borrows from notions of Synchronous Reactive Languages [16], Real-Time Scheduling Theory, and Concurrent Models of Computations and Communication (MoCCs) in Concurrency Theory [14] 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, verification and relation from models to code, respectively.

3 Research program

3.1 Cyber-Physical co-modeling

In Real-Time embedded systems, timing criticality imposes to take time predictivity as much into account as functional determinism, and this from the very beginning design phases onward. In addition, 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, with multiple logical time sources and scales.

Figure 1 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 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 formalizing process algebras accounting logical time, on extending logical frameworks with logical time [6, 7], and on accounting for resource discovery also in presence of mobility inherent to connected objects and Internet of Things [2, 11].

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 [18]. 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, as introduced in [8]. 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 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 and to standardize it.

3.7 IoT standardization

Within the shield of the ETSI standardization consortium, we study protocols, models and performances of widely used CPS, mCPS and IoT frameworks and protocols: we focus on **oneM2M** which, in 2022, has been promoted by the Indian government as the *de facto standard* for all the smart cities applications and it is currently employed in France in (*e.g.* Bordeaux smart cities applications). We experiment, with suitable proof-of-concept, simulation software to evaluate performances and prototype applications, and, we employ the Team's theoretical and implementation skills to improve and make standards alive and useful, like *e.g.* in the Asynchronous Contact Tracing (ACT) ETSI protocol standard, intensively built upon the oneM2M¹ global standard partnership, see Subsection 6.2.

3.8 Formal semantics of process algebras underpinning synchronous languages

CPS and IoT and in general embedded systems are programmed using *ad hoc* synchronous languages: the latter need a special care in specifying their formal semantics and safety properties. We focus on syntax and static (types) and dynamic (rewriting) operational semantics applied to process calculi, inspired to new and old MoCC proposals in order to capture, in the same framework and with the same semantic tools (lambda-calculus, term-rewriting systems, type systems), both the *event-driven* and the *data-flow* programming styles.

¹oneM2M is a global partnership project founded in 2012 and constituted by 8 of the world's leading ICT standards development organizations, notably: ARIB (Japan), ATIS (United States), CCSA (China), ETSI (Europe), TTA (USA), TSDSI (India), TTC (Japan). The goal of the organization is to create a global technical standard for interoperability concerning the architecture, API specifications, security and enrolment solutions for Machine-to-Machine and IoT technologies based on requirements contributed by its members [Source Wikipedia].

3.9 Logical Frameworks with union and intersection types

Our work in this direction is embodied in Bull, a prototype interactive theorem prover based on a polymorphic λ -calculus defined in the Team, called Δ -Framework formally defined in [10], and described in the Software section. Bull [17] has a clear syntax, semantics, typing, subtyping, unification, refinement, and a simple REPL. It is commonly used as a tool to experiment, among others, our ideas and researches in applying type theories to synchronous process algebras.

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 (SC), as popularized in Blockchain frameworks, as a way to formally establish specification of behavioral system traces, applied to connected objects in an 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.

Thanks to the Team skills in higher-order and object-oriented programming, we also investigate ways to extend smart contract languages by the capability of modifying (or self-modifying) at run-time. Modifiable smart contracts are contracts that can be altered or updated after they have been deployed on the blockchain; they can also improve efficiency, as they allow for updates to be made to existing contracts rather than requiring the creation and deployment of entirely new contracts. They may also be more secure than non-modifiable contracts, as they allow for the correction of errors or vulnerabilities that may be discovered after deployment.

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 an 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 Semantic Discovery

Within the shield of ETSI (European Telecommunication Standard Institute), we participate to the ETSI IoT standardization (SmartM2M Technical Committee). Our skills on IoT Semantic Discovery Protocols [42]

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, we participate to the standardization of Digital Contact Tracing IoT Protocols (SmartM2M Technical Committee). We introduce 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) [12].

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

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, 2019-2022) also includes algorithms for computing the symbolic semantics of Open Systems, and testing bisimulation equivalences in a compositional way, using symbolic methods based on the Z3 SMT engine.

News of the Year: The 2022 release now includes algorithms for : (i) testing both strong and weak versions of our symbolic bisimulation equivalence, starting from a candidate relation between states provided by the user, (ii) 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, Bartłomiej Szejna, Eric Madelaine, Ludovic Henrio, Marcela Rivera, Nassim Jibai, Oleksandra Kulankhina, Siqi Li, Xudong Qin, Zechen Hou, Biyang Wang, Viacheslav Herasymov, Rabea Ameer-boulifa

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:

1) graphical and/or textual interactive specification of logical clocks and relative constraints between them, 2) definition and handling of user-defined clock constraint libraries, 3) automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction, 4) call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...). 5) 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. 6) 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):

1) 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. 2) 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. 3) 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. 4) BCOoL, a tool-supported meta-language dedicated to the specification of language coordination patterns to automatically coordinates the execution of, possibly heterogeneous, models. 5) Monilog, an extension for monitoring and logging executable domain-specific models 6) 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>

Publications: [hal-00850770](#), [hal-01355391](#), [hal-01609576](#), [hal-01651801](#), [hal-01152342](#), [hal-03374955](#), [hal-01614561](#), [hal-01616154](#)

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. Trollet 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, <https://www.absint.com/ait/index.htm>). 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 (<https://www.absint.com/ait/index.htm>) 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 Interactive 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. 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 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>

Publications: [hal-01701934](#), [hal-01963662](#), [hal-02573605](#)

Contact: Claude Stolze

Participants: Luigi Liquori, Claude Stolze

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 computing, Web Services, Parallel computing, Component models, Software Components, P2P, Dynamic components, Internet of things, Distributed Applications

Functional Description: Idawi is a middleware for the development and experimentation of distributed applications for mobile multi-hop networks, such as the IoT, the Edge, Mobile Ad hoc Networks, etc. Its development was initially motivated by our need to deploy scientific applications in clusters of computers, in order to run large experimentation campaigns of graph algorithms.

Idawi is an innovative arrangement of many features found in existing tools into a fresh Open Source Java reference implementation.

Idawi defines applications elements as components organized into a multi-hop overlay network on top of TCP/UDP and SSH, to be able to communicate even in the presence of NATs and firewalls. In the usual use case, there will be only one component per device. But, in order to enable the simulation/emulation of large systems, components can deploy other components in their Java Virtual Machine (JVM) or in another JVM(s) in the same device.

Idawi proposes a structuring model of distributed applications, which then must conform to a specific Object-Oriented model in the style of SOA: it defines that components expose their functionality via services. Services hold data and implement functionality about the specific concern they are about. Functionality is then exposed via operations, which can be triggered remotely from anywhere in the component overlay.

The decentralized communication model of Idawi matches the very nature of mobile multi-hop networks. It defines that components communicate with each other via messages of bounded size.

Messaging can be both synchronous (imperative) and asynchronous (reactive/event-driven). It is powered by a default routing scheme and APIs that are tailored to collective communication, so as to offer native support of parallel processing.

Idawi comes with a set of built-in fully decentralized services for automatized quick deployment/bootstrapping of components through SSH, interoperability through a REST-based web interface, service provisioning and discovery, overlay management, and many other system-level functionality.

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

Publications: [hal-03863333](#), [hal-03886521](#), [hal-03562184](#)

Contact: Luc Hogue

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

6.1.12 ACT

Name: Asynchronous Contact Tracing

Keyword: Contact tracing

Scientific Description: Implementation of standard ETSI TS 103757

Functional Description: ACT consists in 3 modules: 1) an ETSI/oneM2M communication infrastructure, 2) a mobile application (android), and 3) a web application.

Release Contributions: First open-source and public version.

URL: https://github.com/AlessioDD98/Asynchronous_Contact_Tracing

Publications: [hal-03127890](#), [hal-03935906](#)

Contact: Luigi Liquori

Participants: Luigi Liquori, Alessio Di Dio

Partners: ETSI, Université Côte d'Azur (UCA)

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 [Advanced Semantic Discovery](#) (ASD) 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. Putting the discovery feature in the next oneM2M standard release is the main contribution of this year.

6.2.2 Asynchronous Contact Tracing

Participants: Luigi Liquori, Alessio Di Dio.

The main result of this year is the gitHub repository of [Asynchronous Contact Tracing](#) (ACT) that contains the first reference prototype implementation of the ETSI ACT standard [43] using the oneM2M standard platform provided by Telecom Italia Network (TIM). This work has been conducted in the ETSI TC SmartM2M and TS eHEALTH Technical Committee.

7 New results

7.1 Decision process for Logical Time

Participants: Frédéric Mallet, Pavlo Tokariev, Robert de Simone.

We are progressing the study of efficient methods for (syntactic) model-based expressiveness and (semantic) analysis/verification of multiform Logical Time (LT) formalisms, centered around our original CCSL constraint language (with possibly relevant extensions/restrictions) [20, 26]. This is a topic of collaboration with Chinese partners, mostly in the context of our associated-team Plot4IoT (while travel exchanges have been on long stand-by this year).

In the starting thesis of Pavlo Tokariev, we are considering how distinctive CCSL features may be handled best by proper combinations of Boolean model-checking and integer linear programming solvers.

Finally, the PhD thesis of Hui Zhao on modeling of cyber-physical systems could be defended, after delays due to distantiation [38].

7.2 Formalizing and extending Smart Contracts languages with temporal and dynamic features

Participants: Frédéric Mallet, Enlin Zhu, Luigi Liquori, Mansur Khazeev.

This year we have pursued the work on the ANR project SIM 9.3, which has started in January 2021. This project funds the PhD of Enlin Zhu. While we started focusing mainly on smart contracts and their acceptance by lawyers, we have now decided to focus on the distributed application as a whole since what can be done on the smart contract itself is not sufficient to have a global end-to-end control of the services. The main result is a state-based modeling of the recovery mechanism used in distributed ledger applications. This is a clear aspect where our formalisms are very pertinent to capture both state dependent properties and (logical) time requirement. This is also a place where the actual implementation and the temporality of the orchestration has a direct impact on the function to be achieved. It is indeed very good to show the benefit of a model-based approach to reason, conduct verification and then hope to generate correct-by-construction code. We mainly target the generation of the control part that puts into music the purely functional parts deployed either on smart contracts or on some of the actors of the distributed service (including oracles). While this result has been submitted twice, it was not yet accepted by a reviewing committee.

Another main result of this year is to understand how modifiable smart contracts, as introduced in an object-oriented setting [4], can increase smart contract flexibility, as modifiable contracts allow for changes to be made to the terms of the contract after it has been deployed. This can be particularly useful in situations where the contract needs to adapt to changing circumstances or evolving business needs. Modifiable smart contracts can also improve efficiency, as they allow for updates to be made to existing contracts rather than requiring the creation and deployment of entirely new contracts. This is the topic of the Ph.D. of Mansur Khazeev.

7.3 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 starting PhD thesis of Arseniy Gromovoy the "light-weight" stochastic extension of CCSL with statistical purposes. The idea is to admit that a clock can be faster than (or also included in) another one *safe for a few occasions*. Unlike fancy stochastic model-checkers, which aim at computer percentage likeness of property satisfaction, we attempt to reduce probability aspects as soon as possible to a known number of extra steps in reachable state space construction, to reduce the problem to more tractable, homogeneous data structures in verification so that to be able to reduce the problem.

7.4 Safety rules for autonomous driving

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

This work, conducted in the context of a CIFRE PhD contract with Renault Software Factory, led to the PhD defense of Joëlle Abou Faysal in June 2022 [37], and to her hiring as R&D engineer in this company. We provided a formal language to describe and verify safety rules for autonomous vehicles, with a concrete syntax meant to be close to Safety Engineers expectations, and a semantics allowing formal verification based on our modeling and verification environments around logical time [28, 34]. Note that from September 2022, we are revisiting this language in the newly started PhD thesis of Maksym Labzhaniia, where the notion of logical space, and its mapping to physical space is introduced.

7.5 Take-over between automated and human automotive driving

Participants: Robert de Simone, François Revest, Ankica Barisic, Julien Deantoni.

This work was conducted as part of PSPC project ADAVEC 9.4, coordinated for UCA and Inria by Amar Bouali, and with industrial partners the local companies EpicnPoc and Avisto, and a back-up from Renault Research Factory in Sophia-Antipolis.

The goal is to study modeling of disengagement principles, by which driving responsibility may be taken over in between either a human or an automatic IA driver. This requires the formalization of take-over rules, which strongly rely on observation of both the outside road environment, but also the cockpit and the human driver state. This results in a complex software system that can be connected either to simulators of various levels, or to actual sensors embedded in the "Bowl", a physical modular driving cockpit meant for experimentation and demonstrated by EpicnPoc at the Consumer Electronic Show (CES) in Las Vegas, in January 2022. A special focus was put this year on Software-Oriented Architecture (SOA) as a mean to connect software components efficiently. Work on CPS sustainability has also been shown in the project [39].

7.6 On the generation of compiler friendly C code from CCSL specification

Participant: Baptiste Allorant, Sid Touati, Frédéric Mallet.

This represents essentially the starting PhD work of Baptiste Allorant, newly hired in the team. The goal is to produce efficient sequential C codes from a CCSL description. A first version of the tool named ECCOGEN is completed, and under intensive testing. We are currently conducting extensive performance evaluation to test thousands of generated compiler friendly codes. The tool can run on four prevalent compilers: GNU gcc, INTEL icc, CompCert, and CLANG/LLVM. In addition to the small benchmarks of CCSL at hand, we produce synthetic ones with random generation techniques as well. A detailed research report is under way. The next step will be to extend ECOGEN in order to generate tuned *parallel* C code.

7.7 Continuation of the work on the program recognition with neural networks

Participant: Sid Touati.

This is joint work with Enrico Formenti, professor at UCA, including the co-supervision of master 2 UCA intern Alessandro Pepegna, also as a follow-up of work started last year with Corentin Fossati. Alessandro added two main contributions:

1. Implementing with LLVM a tool that computes the data dependency graph of a C code, and prints it in a graph file format. This graph is annotated with some compiler semantic attributes to grab the semantics of the C code.
2. Implementing with LLVM a tool that generates multiple data dependency versions from the same C code, by considering multiple combinations of compiler flags and phase orderings. All these versions of the same C code will be used in the machine learning phase (training the neural network).

7.8 Behavioral semantics in Model Driven Engineering

Participants: Julien Deantoni, Joao Cambeiro.

In the last years, in collaboration with other academic researchers, we developed the GEMOC studio, a language workbench to develop new modeling or programming languages and their interactions, based on a formal concurrency model.

Two new results emerged this year. First, we developed the formal, concurrent and timed operational semantics of platform aware Modal Sequence Diagrams [22] (MSD). The originality was to encode in the behavioral semantics the refinement resulting from the deployment of an application into a specific hardware architecture. More precisely, 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 enabled 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](#). Second, 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 [27]. These strategies are defined independently of the execution engine and implemented into the GEMOC studio. The companion webpage is accessible [here](#). Currently, we are investigating further the notion of refinement and its impact at the system engineering level through the notion of model fidelity. This is specifically addressed in the PhD thesis of Joao Cambeiro where we are characterizing the capability of different models to mimic a same reality.

7.9 Efficient parallelism in shared memory

Participant: Dumitru Potop Butucaru.

We progressed further the technology underlying the LoPhT tool 6.1.6. Recent results are published in [32], and we will continue these efforts as part of the recently accepted ANR *Caotic* project. Ongoing discussions are taking place with Airbus concerning the continued evaluation of LoPhT for industrial use.

7.10 A language and compiler for embedded and Real-Time Machine Learning programming

Participants: Hugo Pompougnac, Dumitru Potop Butucaru.

This research axis connects three research and engineering fields: Real-Time/Embedded programming, High-Performance Compilation, and Machine Learning. Work has been funded by the *Confiance . AI PIA* project and by the ES3CAP project. Hugo Pompougnac defended his PhD thesis on this topic in December 2022.

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 properties² established it as a sound basis for High-Performance Computing (HPC) compilation frameworks such as *MLIR*, where different abstraction levels of the same application³ 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 application⁴. 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, have been published in *ACM TACO* [23] and was presented in the *HiPEAC 2022* conference. A second paper is under writing. The *mlirlus* software has been published as open-source.

An ANR project proposal has been submitted, and our participation in the *Confiance . AI* national initiative (in project EC7 - embedded AI) continues.

7.11 Formal verification of Logical Execution Time 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 Logi-

²Functional determinism while still allowing for limited concurrency.

³Ranging from ML dataflow graphs and linear algebra specifications down to affine loop nests and optimized (tiled, vectorized,...) low-level code.

⁴A pitch tuning vocoder.

cal Time, logical and pseudophysical clocks, clock-based temporal programming primitives) and from Logical Execution Time (LET) 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). A technical report is under progress, which can be considered as a formal semantics description of the main language features. A consistent set of results has been presented at ERTS2022 conference [33].

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

- 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.

7.12 Interference analysis in Real-Time task mapping

Participants: Frédéric Fort, Robert de Simone.

Concurrent real-time tasks, from a single application or simply run on a common multicore processor, may interfere mutually because of shared resources. As a result, the global end-to-end worst-case latency is not always a simple combination of individual worst-case execution times taken in isolation. We considered the issue of modeling-for-analysis for such system-level features, with a concern on compatibility with existing design practices of industrial partners (Renault Software Factory and Thales Alenia Space in that case), in the context of IRT Saint-Exupery *Archeocs* collaborative project 8. This work led to the post-doc programme of Frédéric Fort, started in November 2022.

7.13 Trustworthy Fleet deployment and management

Participant: Nicolas Ferry, Marie-Agnès Peraldi Frati, Julien Deantoni.

This activity is a follow-up of Nicolas Ferry previous activities in ENACT H2020 project, now renewed since his arrival in Kairos in the TrustFleet PHC and DYNABIC HEU projects 9.2.2. 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.

Results are presented in [24, 25]. In 2022, GeneSIS (a part of the aforementioned results) was recognized by the EC innovation radar as highly innovant with high maturity level. 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 and trustworthiness and to minimize the downtime. Kairos led a proposal for an EU project (Horizon Europe call HORIZON-CL3-2022-CS-01-02: Trustworthy methodologies, tools and data security “by design” for dynamic testing of potentially vulnerable, insecure hardware and software components), which has been submitted in November 2022 and where we aim at leading research activities along the main activities of the project on Co-modeling.

7.14 Model-Based serverless platform for the Cloud-Edge-IoT continuum

Participants: Nicolas Ferry, Barbara de Oliveira.

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. These observations are being confirmed by our ongoing systematic literature review initiated as part of the TrustFleet PHC aurora project in collaboration with SINTEF (Oslo, Norway), and performed as part of Barbara de Oliveira internship. In parallel, and to address the aforementioned challenges we initiated a first prototyping platform for the design, deployment, as well as maintenance of applications over the IoT-Edge-Cloud continuum [29]. 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.

7.15 Behavioral equivalence of Open Systems

Participant: Eric Madelaine.

This work is conducted in part joint with Rabea Ameer-Boulifa, associate professor at Telecom Paris. 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 kind of data parameters, including, but not limited to, time. The middle term goal is to build a formal framework, but also an effective tool set, for the compositional analysis of such programs. A journal article on these topics was accepted [19].

Weak bisimulation for open systems 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 (despite confinement issues).

7.16 Standardization and performance evaluation in ETSI oneM2M standard

Participants: Luigi Liquori, Marie-Agnès Peraldi Frati.

Within the standards for M2M and the Internet of Things, managed by ETSI, **oneM2M**, and in the context of the new started contract TTF 1019 (2022-2024), we are studying performance evaluation of some (but not all) open source initiatives of the oneM2M standard. 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 database 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).

7.17 Advanced Semantic Discovery for IoT objects and networks

Participants: Luigi Liquori, Marie-Agnès Peraldi Frati.

Within the shield of ETSI (European Telecommunication Standard Institute), we study and participate to the Advanced Semantic Discovery (ASD) and query capabilities for 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. This is a direct follow-up of our researches on looking for suitable mechanisms and protocols to perform an ASD as introduced in [9] and [42].

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 flavors, which is not optimal for interworking and interoperability. For further information, we refer to the ended contract **STF 589** and the software released in **ETSI Labs ASD**.

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

Participant: Luigi Liquori, Alessio Di Dio.

Within the shield of ETSI, we standardised [43] a novel contact tracing protocol, called Asynchronous Contact Tracing (ACT). ACT traces the IoT connected objects 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.

In 2022, with the Master student Alessio di Dio, we released the first implementation of ACT, consisting in 3 modules: (i) an ETSI/oneM2M communication infrastructure, (ii) a mobile application (android), and (iii) a web application [40], see Figure 2.

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 [36].



Figure 2: ACT Web and Mobile application

7.19 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, both in Serbia, we present a comprehensive, yet simple, extension to the existing systems used for Digital Contacts Tracing (DGT) in Covid-19 pandemic. The extension, called BAC19, will enable those systems, regardless of their underlying protocol, to enhance their sets of DGT 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 (DHT), as introduced in [11, 15]. Its model is inspired by the Chord and Synapse structured overlay networks. The published work presents the architecture of the overlay network federation and shows that the federation can be used as a formal model of forward contact tracing [21].

7.20 Logical Time modeling and implementation of distributed systems

Participant: Dumitru Potop Butucaru.

As part of a scientific collaboration with Google (US) we are exploring the modeling and implementation of distributed heterogenous systems using logical time formalisms. The collaboration should receive formal support in the beginning of 2023.

7.21 A middleware for the experimentation on IoT mobile networks

Participant: Luc Hogie.

In the context of studies on decentralized algorithms from mobile dynamic networks, we investigated the state of the art of the experimentation tools. We discovered that existing solutions, either coming from labs or companies, do not match the requirements of experimentation as it is usually done by Researchers. Indeed commercial products focus on reliability and interoperability at the expense of versatility, while lab tools most often serve as proof of concepts. The experimental study of algorithms requires the availability, in a single solution, of the following features: support for both synchronous and asynchronous communication, simulation/emulation of large systems, fast deployment, Web interoperability, and full decentralization, just to name a few. Idawi was designed and implemented to the very purpose of providing the Research community with a tool tailored to its needs. The resulting middleware has been

released on the MAVEN global distribution platform for Java software. MAVEN Central statistics indicate that Idawi binaries have been downloaded 350 times in the year 2022. In addition to this, the Idawi and its satellite tools are all Open Source. They are released under the Apache V2 license. For the sake of Open Science, their source code is fully available on GitHub, see [30, 31, 41].

8 Bilateral contracts and grants with industry

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

IRT Archeocs. This collaboration deals with identification of real-time interferences in the context of concurrent tasks mapped to a common multicore architecture. Industrial partners come from both the Toulouse and Sophia areas, and we have special links with local Thales_Alenia_Space and Renault Software Labs. Frédéric Fort was hired in October 2022 as postdoc to conduct work under our supervision. See 7.12 for technical content.

Krono-Safe CIFRE contract. The PhD of Fabien Siron, reported in 7.11, came with a grant from this company. Joint efforts extend partly to the objectives of the Archeocs IRT project 7.12.

Renault Software Labs CIFRE contract. 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 7.4. The CIFRE PhD of Joëlle Abou-Faysal was defended in March 2022.

ETSI TTF contract. The TTF (Testing Task Force) is funded by ETSI (2022-2024) and integrated within the Work Programme of the Technical Committee SmartM2M (TC SmartM2M). Last year we received support to extend the oneM2M standard with a powerful IoT semantic discovery ended STF contract. This year we received support to conduct the performance evaluation, analysis, planning and deployment for some (but not all) oneM2M open source initiatives. A systematic comparative study will be done to compare connectivity, interoperability, data management, security, and complex architecture issues.

9 Partnerships and cooperations

9.1 International initiatives

9.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program

Participants: Frédéric Mallet, Eric Madelaine, Robert de Simone.

PLoT4IoT

Title: Probabilistic Logical Time for IoT

Duration: extended since 2020 because of confinement stand-by.

Coordinator: Min Zhang (zhangmin@sei.ecnu.edu.cn)

Partner: East China Normal University Shanghai (Chine)

Inria contact: Frédéric Mallet

Summary: The growing importance of Connected Objects in the Internet of Things (IoT) poses new challenges concerning modeling and design of so-called Cyber-Physical Systems (CPS), where cyber/discrete controller programs interplay with physical (often continuous) environments. While there are generally well-established modeling practices in physical science domains (often including discretization), the need for equally formal modeling treatment of reactive control software itself becomes all the more important, since correctness of functional and non-functional properties relies on the whole range of models, including the software executable specification models. The scope of the PLoT4IoT proposal is entirely devoted to the definition and analysis of modeling paradigms relevant for the design of (mainly the cyber part of) CPS. In this context, we shall study extensions to Logical Time models for CPS, dedicated on one hand to uncertainty and variability, and on the other hand to spatio-temporal aspects and mobility. These models will be proposed for standardisation into the forthcoming version of the OMG UML MARTE profile. On a more abstract level, we shall study the semantics and the analysis methods for open (concurrent) systems featuring explicit handling of data, time, and locations, and propose new approaches to formal verification of safety properties of these systems. As a transversal mathematical tool for these works, we plan to develop new efficient strategies for "Satisfiability Modulo Theory" tools adapted to the three modeling theories above.

Standardization initiatives

Participants: Luigi Liquori, Marie-Agnès Peraldi Frati.

- Luigi Liquori and Peraldi Frati are members of ETSI, European Telecommunications Standards Institute, Technical Committee (TC) SmartM2M - Smart Machine-to-Machine Communications, and oneM2M - Standards for M2M and the Internet of Things.
- Luigi Liquori is member of ETSI TC eHEALTH - Technical standards-based market for health.
- Luigi Liquori is member of ITU's Telecommunication Standardization Sector (ITU-T).
- Luigi Liquori is member of ECMA, industry association dedicated to the standardization of information and communication systems, Technical Committee 39: Specifying JavaScript.
- Luigi Liquori is member of AFNOR, French Standardization Association, member body for France at the International Organization for Standardization (ISO), JTC1/SC22/WG14, Programming language C++.

9.2 European initiatives

9.2.1 Horizon Europe

Participant: Nicolas Ferry.

The DYNABIC EU project recently started in December 2022. DYNABIC stands for *Dynamic business continuity and response of critical systems against advanced cyber-physical threats*. The overall objective of DYNABIC is to increase the resilience and business continuity capabilities of European critical services in the face of advanced cyber-physical threats. The Kairos (Nicolas Ferry) and Sparks (Jean-Yves Tigli and Stéphane Lavitrotte) teams from I3S are involved in the project contributing to two research activities: (i) T4.3 - Cyber-physical system Monitoring and Business Continuity Situational Awareness (main contribution from Sparks as task leader) and (ii) T5.3 Multi-layer Security Orchestration and Automatic Response (contribution from Kairos). In addition, Nicolas Ferry is WP7 leader (Dissemination, Communication, and Exploitation). The project is a follow-up initiative of long-standing cooperations with SINTEF Norway 7.13.

9.2.2 Other european programs/initiatives

Participants: Nicolas Ferry, Julien Deantoni, Marie-Agnès Peraldi Frati.

The TrustFleet PHC Aurora project started in 2022. TrustFleet aimed at bootstrapping a collaboration between the Trustworthy Green IoT Software group in SINTEF (Oslo, Norway) and Kairos, see technical details in 7.13. The project already resulted in the submission of a EU proposal to the HORIZON-CL3-2022-CS-01-02 call. An application to extend the project for an extra year has been submitted in December 2022.

9.3 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 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 Contract 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).

CNRS GDRs We are registered members of three GDR funded by CNRS : **SoC²**, on topics of Hardware/Software codesign and Non-Functional Property modeling for co-simulation; **LTP**, on verification and language design for reactive CPS systems; **GPL**, on software engineering and Domain-Specific Languages.

PIA ES3CAP. ES3CAP (Embedded Smart Safe Secure Computing Autonomous Platform) was a PIA (Programme d'Investissements d'Avenir) project, ended in February 2022. The national coordinator was Kalray, and other partners included Safran, Renault, and MBDA. The objectives of the project were 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 7.9. This project partially finances Hugo Pompougnac's PhD and Jad Khatib's post-doc.

Grand Défi - Confiance.AI. We participate to the Confiance.AI programme, in the project "Embedded AI". It 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 7.10.

9.4 Regional initiatives

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, as well as the post-doc position of Ankica Baresic. Technical advances are reported in 7.5.

10 Dissemination

Participants: Robert de Simone, Julien Deantoni, Luigi Liquori, Frédéric Mallet, Nicolas Ferry.

10.1 Promoting scientific activities

10.1.1 Scientific events: organisation

- Julien Deantoni organized two working days on the notion of fidelity in System Engineering with 10 international specialists, invited specifically. The meeting was held in Lisbon and is affiliated to the PhD topic of Joao Cambeiro (see Section 7.8).
- Robert de Simone organized in November at the CNRS Center Villa Clythia in Frejus the yearly Synchron open seminar on Synchronous Reactive Languages, with about forty attending researchers (half-French, half international).
- Nicolas Ferry was Organization chair of the 2nd International Workshop on MDE for Smart IoT Systems (MESS), co-located with STAF'22.

Member of the organizing committees

- Nicolas Ferry was Organization chair of the 2nd International Workshop on MDE for Smart IoT Systems (MESS), co-located with STAF'22.
- Liquori and Peraldi Frati were Technical Program Committee of the ETSI IoT Week'22.
- Frederic Mallet was member of the program committee for *Journées Scientifiques Inria* organized in Rocquencourt.

Reviewer

- Nicolas Ferry was reviewer for IET Software, SOSYM, DevOps@Models'22.
- Luigi Liquori was reviewer for the ESOP 2023 European Conference and for the ACM Transactions on Programming Languages and Systems (TOPLAS) journal.

10.1.2 Journal

Member of the editorial boards

- Julien Deantoni is associate editor of the ACM TECS journal, special issue on Specification and Design Languages (CFP)
- Nicolas Ferry is associate editor of the Journal of Object Technology (JOT), special issue on MDE for Smart IoT Systems.

Editors

- Frédéric Mallet was editor of a special issue on selected papers from the the VIII International Scientific Conference IT&I-2021 : Information Technology and Implementation [35].

10.1.3 Invited talks

- Nicolas Ferry gave an invited talk entitled "ENACT: DevOps for trustworthy IoT systems" at Pole SCS day.
- Luigi Liquori gave an invited talk entitled " Intersection, Union, Dependent Types and SubType Systems" at Computational Logic Autumn Summit - CLAS 2022, Tbilisi, Georgia.

10.1.4 Scientific expertise

- Robert de Simone was member of the ANR project selection jury for the Call CES25 (*Réseaux de communication multi-usages, infrastructures de hautes performances, Sciences et technologies logicielles*).

10.1.5 Research administration

- Frédéric Mallet is the new Director of UMR I3S since January 2022.

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- Licence: Sid Touati, Fondement machine, 75h eq TD, L1 informatique, Université Côte d'Azur.
- Licence: Sid Touati, Architecture machine, 45h eq TD, L3 informatique, Université Côte d'Azur.
- Licence: Sid Touati, Compilation, 33h eq TD, L3 informatique, Université Côte d'Azur.
- Master: Sid Touati, Architectures et logiciels hautes performances, 81h eq TD, Master 1 informatique, Université Côte d'Azur.
- Master international: Sid Touati, Advanced operating systems, 30h eq TD, Master 1 informatique, Université Côte d'Azur.
- International Master: Frédéric Mallet, Safety-Critical Systems, 32h eq TD, M1, Université Côte d'Azur.
- International Master: Frédéric Mallet, Software Engineering, 32h eq TD, M1, Université Côte d'Azur.
- Master: Frédéric Mallet, Programmation Synchrones, 32h eq TD, M1, Université Côte d'Azur.
- Master: Robert de Simone, Formal Model-Based Design for Cyber-Physical Systems and IoT, 36h 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, 20h eq TD, 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, Université d'Avignon.
- Master: Luigi Liquori, Peer-to-peer systems, 32h eq TD, M2 CASPAR, Polytech Nice Sophia, Université Côte d'Azur.
- International Ph.D. School: Luigi Liquori, Summer School of the 19th International Colloquium on Theoretical Aspects of Computing, 9h eq TD, Tbilisi, Georgia.
- 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.
- Master: Julien Deantoni, Architecting IoT systems, Beyond Functional Correctness, 32h eq TD, Polytech Nice Sophia, Université Côte d'Azur.
- Licence: Julien Deantoni, Language and Compilation, 32h eq TD, L3IA, Université Côte d'Azur.
- Licence: Julien Deantoni, Introduction à l'informatique par le Web, 99h eq TD, DS4H portail science, Université Côte d'Azur.
- Licence: Julien Deantoni, Programmation et Conception Orientées Objets, 48h eq TD, DS4H, Université Côte d'Azur.
- 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 eq TD, EPITA Engineering School, Paris.
- Master: Dumitru Potop-Butucaru, Real-time embedded systems, 42h eq TD, EIDD, École d'Ingenieur Denis Diderot, Université Paris Cité.
- BUT: Nicolas Ferry, Software Quality, 18h eq TD, Software Quality 2, 20h eq TD, Software Architecture, 50h eq TD, IUT Nice Côte d'Azur, Université Côte d'Azur.
- 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, Architecting IoT systems, Beyond Functional Correctness, 8h eq TD, Polytech Nice Sophia, 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, Université Côte d'Azur.

10.2.2 Supervision

- Dumitru Potop-Butucaru supervised the PhD thesis of Hugo Pompougnac, defended in December 2022, Université Paris Sorbonne.
- Frédéric Mallet supervised the PhD thesis of Joëlle Abou-Faysal, defended in March 2022, Université Côte d'Azur.
- Julien Deantoni supervises of the PhD of Joao Cambeiro, funded on an EDSTIC scholarship, Université Côte d'Azur.
- Frédéric Mallet supervises the PhD thesis of Enlin Zhu, funded on the SIM ANR contract, Université Côte d'Azur.
- Frédéric Mallet supervises the PhD thesis of Pavlo Tokariev, funded on an Inria CORDI fellowship, Université Côte d'Azur.
- Robert de Simone supervises the PhD thesis of Arseniy Gromovoy, started in January 2022, on an EDSTIC scholarship, Université Côte d'Azur.
- Frédéric Mallet, Julien Deantoni, and Marie-Agnès Peraldi-Frati co-supervise the PhD thesis of Maksym Labzhaniia, started in September 2022, Université Côte d'Azur.
- Sid Touati and Frédéric Mallet co-supervise the PhD thesis of Baptiste Allorant, started in October 2022, Université Côte d'Azur.
- Robert de Simone and Dumitru Potop-Butucaru co-supervise the PhD thesis of Fabien Siron, funded on the CIFRE grant, Université Côte d'Azur.
- Luigi Liquori supervises the PhD thesis of Mansur Khazeev, Université Côte d'Azur.
- Nicolas Ferry supervised the Master 2 Ubinet internship of Barbara de Oliveira, Université Côte d'Azur. The topic of the internship was: *Function as a Service for the Internet of Things: a systematic literature review*.
- Luigi Liquori supervised the Master 2 Ubinet internship of Alessio di Dio, Université Côte d'Azur. The topic of the internship was: *Asynchronous Contact Tracing*.

10.2.3 Juries

- Luigi Liquori was jury member in the defense Ph.D. Thesis of Riccardo Treglia, defended at University of Turin, Italy.
- Frederic Mallet was referee and president of the HDR jury of Daniela Cancila, Researcher at CEA, and defended at Université Paris Saclay.
- Frederic Mallet was member of the PhD committee of Thibaud L'Yvonnet, defended at Université Côte d'Azur.
- Frederic Mallet was referee for the PhD defense of Meryem Afendi, defended at Université Paris-Est Créteil.
- Frederic Mallet was referee for the PhD defense of Adrien Jousse, defended at INSA Centre-Val de Loire.

10.3 Popularization

10.3.1 Intervention

- Marie-Agnès Peraldi Frati participated to "les cordées de la réussite" initiative by collaborating with the Émile Roux, Le Cannet school to introduce teenagers to 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 M. Sereno. *LogNet: Extending Internet with a Network Aware Discovery Service: [Extended abstract]*. 14th Nov. 2017. URL: <https://hal.inria.fr/hal-01323974>.
- [10] 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>.
- [11] 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>.
- [12] L. Liquori, S. Wood and E. Scarrone. *Asynchronous Contact Tracing*. 14th Dec. 2020. URL: <https://hal.inria.fr/hal-02989404>.

- [13] 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>.
- [14] 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>.
- [15] 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>.
- [16] 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>.
- [17] 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>.
- [18] 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

- [19] R. Ameur-Boulifa, L. Henrio and E. Madelaine. ‘Compositional Equivalences Based on Open pNets’. In: *Journal of Logical and Algebraic Methods in Programming* (2022). URL: <https://hal.science/hal-03894031>.
- [20] X. Chen, Q. Liu, F. Mallet, Q. Li, S. Cai and Z. Jin. ‘Formally verifying consistency of sequence diagrams for safety critical systems’. In: *Science of Computer Programming* 216 (Apr. 2022), p. 102777. DOI: [10.1016/j.scico.2022.102777](https://doi.org/10.1016/j.scico.2022.102777). URL: <https://hal.inria.fr/hal-03836230>.
- [21] S. Ghilezan, S. Kašterović, L. Liquori, B. Marinković, Z. Ognjanović and T. Stefanović. ‘Federating Digital Contact Tracing using Structured Overlay Networks’. In: *Computer Science and Information Systems*. Special Issue on Compilers, Languages, Related Technologies, and Applications 19.3 (1st Sept. 2022), pp. 1261–1282. DOI: [10.2298/CSIS210825029G](https://doi.org/10.2298/CSIS210825029G). URL: <https://hal.inria.fr/hal-03127890>.
- [22] J. Holtmann, J. Deantoni and M. Fockel. ‘Early timing analysis based on scenario requirements and platform models’. In: *Software and Systems Modeling* 21 (28th Apr. 2022). DOI: [10.1007/s10270-022-01002-3](https://doi.org/10.1007/s10270-022-01002-3). URL: <https://hal.inria.fr/hal-03375049>.
- [23] H. Pompougnac, U. Beaugnon, A. Cohen and D. P. Butucaru. ‘Weaving Synchronous Reactions into the Fabric of SSA-form Compilers’. In: *ACM Transactions on Architecture and Code Optimization* 19.2 (30th June 2022), pp. 1–25. DOI: [10.1145/3506706](https://doi.org/10.1145/3506706). URL: <https://hal.inria.fr/hal-03935275>.
- [24] T. Rajmohan, P. Nguyen and N. Ferry. ‘A decade of research on patterns and architectures for IoT security’. In: *Journal of Cybersecurity* 5.2 (5th 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>.
- [25] H. Song, R. Dautov, N. Ferry, A. Solberg and F. Fleurey. ‘Model-based fleet deployment in the IoT-edge-cloud continuum’. In: *Software and Systems Modeling* (3rd May 2022). DOI: [10.1007/s10270-022-01006-z](https://doi.org/10.1007/s10270-022-01006-z). URL: <https://hal.archives-ouvertes.fr/hal-03657771>.

- [26] Y. Zhang, F. Mallet and Z. Liu. ‘A dynamic logic for verification of synchronous models based on theorem proving’. In: *Frontiers of Computer Science* 16.4 (Aug. 2022), p. 164407. DOI: [10.1007/s11704-022-1374-4](https://doi.org/10.1007/s11704-022-1374-4). URL: <https://hal.inria.fr/hal-03836232>.
- [27] S. Zschaler, E. Bousse, J. Deantoni and B. Combemale. ‘A Generic Framework for Representing and Analysing Model Concurrency’. In: *Software and Systems Modeling* (2022). URL: <https://hal.inria.fr/hal-03921704>.

International peer-reviewed conferences

- [28] J. Abou Faysal, N. Zalmi, A. Barisic and F. Mallet. ‘Safety analysis of inconsistencies using a formal verification tool for DSML’. In: DSC 2022 Europe VR - 21th Driving Simulation & Virtual reality Conference Europe. Strasbourg, France, 14th Sept. 2022. URL: <https://hal.inria.fr/hal-03846499>.
- [29] N. Ferry, R. Dautov and H. Song. ‘Towards a Model-Based Serverless Platform for the Cloud-Edge-IoT Continuum’. In: CCGrid 2022 - 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing2nd : Workshop on Cloud-to-Things continuum: towards the convergence of IoT, Edge and Cloud Computing. Taormina, Italy: IEEE, 16th May 2022, pp. 851–858. DOI: [10.1109/CCGrid54584.2022.00101](https://doi.org/10.1109/CCGrid54584.2022.00101). URL: <https://hal.inria.fr/hal-03729308>.
- [30] L. Hogie. ‘A Service-Oriented Middleware Enabling Decentralised Deployment in Mobile Multihop Networks’. In: 3rd International Workshop on Architectures for Future Mobile Computing and Internet of Things (FMCIoT 2022), collocated with the 20th International Conference on Service-Oriented Computing (ICSOC 2022). Sevilla, Spain, 29th Oct. 2022. URL: <https://hal.inria.fr/hal-03886521>.
- [31] L. Hogie. ‘Idawi: a decentralized middleware for achieving the full potential of the IoT, the fog, and other difficult computing environments’. In: Proceedings of the 1st Workshop on Middleware for the Edge. Quebec City, Canada: ACM; ACM, 7th Nov. 2022, pp. 1–5. DOI: [10.1145/3565385.3565876](https://doi.org/10.1145/3565385.3565876). URL: <https://hal.inria.fr/hal-03863333>.
- [32] G. Iooss, A. Cohen, D. Potop-Butucaru, M. Pouzet, V. Bregeon, J. Souyris and P. Baufreton. ‘Polyhedral Scheduling and Relaxation of Synchronous Reactive Systems’. In: IMPACT 2022 - 12th International Workshop on Polyhedral Compilation Techniques. Budapest, Hungary, 20th June 2022, pp. 1–12. URL: <https://hal.inria.fr/hal-03901645>.
- [33] F. Siron, D. Potop-Butucaru, R. de Simone, D. Chabrol and A. Methni. ‘The synchronous Logical Execution Time paradigm’. In: ERTS 2022 - Embedded real time systems. Toulouse, France, 1st June 2022. URL: <https://hal.inria.fr/hal-03694950>.

Conferences without proceedings

- [34] J. Abou Faysal, N. Zalmi, A. Barisic and F. Mallet. ‘Adaptation of an auto-generated code using a model-based approach to verify functional safety in real scenarios’. In: ERTS 2022 - Embedded Real Time Systems. Toulouse, France, 1st June 2022. URL: <https://hal.archives-ouvertes.fr/hal-03611183>.

Edition (books, proceedings, special issue of a journal)

- [35] A. Anisimov, V. Snytyuk, A. Chris, A. Pester, F. Mallet, H. Tanaka and I. Krak, eds. *IT&I-2021 : Information Technology and Implementation 2021: Selected Papers of the VIII International Scientific Conference*. Vol. CEUR-3132. CEUR-WS.org, 2nd May 2022. URL: <https://hal.inria.fr/hal-03836228>.
- [36] *How Research meets Standardization: the Asynchronous Contact Tracing ETSI Standard and the PANDESYS Research Activity*. From Research to Standards (2023). URL: <https://hal.inria.fr/hal-03940889>.

Doctoral dissertations and habilitation theses

- [37] J. Abou Faysal. 'Formal rule-based scenarios for the design of safe autonomous vehicles'. Université Côte d'Azur, 9th June 2022. URL: <https://theses.hal.science/tel-03814686>.
- [38] H. Zhao. 'Multi-view design for cyber-physical systems'. Université Côte d'Azur, 22nd Mar. 2022. URL: <https://theses.hal.science/tel-03775554>.

Reports & preprints

- [39] A. Barisic, J. Cunha, I. Ruchkin, A. Moreira, J. Araújo, M. Challenger, D. Savić and V. Amaral. *Modelling Sustainability in Cyber-Physical Systems: A Systematic Mapping Study*. 23rd Mar. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03616678>.
- [40] A. Di Dio and L. Liquori. *Asynchronous Contact Tracing, Fighting Pandemics with Internet of Things. Set up of the of oneM2M infrastructure, mobile and web applications*. Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France, 10th Oct. 2022. URL: <https://hal.inria.fr/hal-03935906>.
- [41] 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>.

11.3 Cited publications

- [42] L. Liquori, R. Gaeta and M. Sereno. 'A Network Aware Resource Discovery Service'. In: *EPEW 2019 - 16th European Performance Engineering Workshop*. Milano, Italy, Nov. 2019. DOI: [10.1007/978-3-030-44411-2_6](https://doi.org/10.1007/978-3-030-44411-2_6). URL: <https://hal.inria.fr/hal-01895452>.
- [43] 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*. Ed. by ETSI. Dec. 2021. URL: <https://hal.inria.fr/hal-02989793>.