

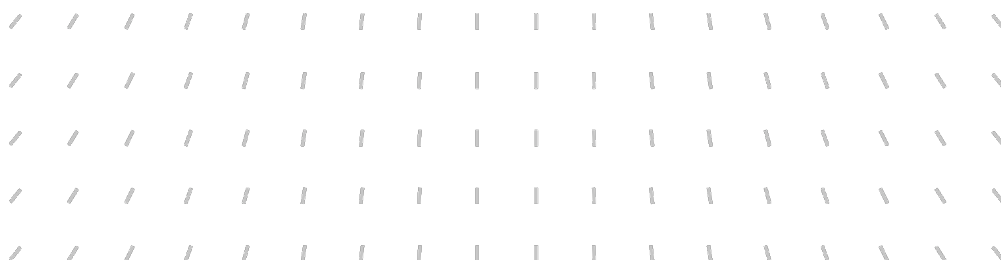
2025 Activity Report

RESEARCH CENTRE: Inria Paris Centre


Project-Team

KOPERNIC

Keeping worst case reasoning for different criticalities



Project-Team KOPERNIC

Creation of the Project-Team: 2021 October 01

Each year, Inria research teams publish an Activity Report presenting their work and results over the reporting period. These reports follow a common structure, with some optional sections depending on the specific team. They typically begin by outlining the overall objectives and research programme, including the main research themes, goals, and methodological approaches. They also describe the application domains targeted by the team, highlighting the scientific or societal contexts in which their work is situated. The reports then present the highlights of the year, covering major scientific achievements, software developments, or teaching contributions. When relevant, they include sections on software, platforms, and open data, detailing the tools developed and how they are shared. A substantial part is dedicated to new results, where scientific contributions are described in detail, often with subsections specifying participants and associated keywords. Finally, the Activity Report addresses funding, contracts, partnerships, and collaborations at various levels, from industrial agreements to international cooperations. It also covers dissemination and teaching activities, such as participation in scientific events, outreach, and supervision. The document concludes with a presentation of scientific production, including major publications and those produced during the year.

Keywords

Computer sciences and digital sciences

- A1.1.1. – Multicore, Manycore
- A1.1.2. – Hardware accelerators (GPGPU, FPGA, etc.)
- A1.5. – Complex systems
 - A1.5.1. – Systems of systems
 - A1.5.2. – Communicating systems
- A2.3. – Embedded and cyber-physical systems
 - A2.3.1. – Embedded systems
 - A2.3.2. – Cyber-physical systems
 - A2.3.3. – Real-time systems
- A4.5.1. – Static analysis
- A4.5.3. – Program proof

Other research topics and application domains

- B5.2. – Design and manufacturing
 - B5.2.1. – Road vehicles
 - B5.2.2. – Railway
 - B5.2.3. – Aviation
 - B5.2.4. – Aerospace
- B6.6. – Embedded systems

Contents

| | |
|--|-----------|
| Project-Team KOPERNIC | 1 |
| 1 Team members, visitors, external collaborators | 5 |
| 2 Overall objectives | 5 |
| 3 Research program | 8 |
| 3.1 Worst case execution time estimation of a program | 8 |
| 3.2 Building measurement-based benchmarks | 9 |
| 3.3 Scheduling of graph tasks on different resources within an energy budget | 9 |
| 4 Application domains | 10 |
| 4.1 Avionics | 10 |
| 4.2 Railway | 10 |
| 4.3 Autonomous cars | 10 |
| 4.4 Drones | 10 |
| 5 Social and environmental responsibility | 11 |
| 5.1 Impact of research results | 11 |
| 6 Highlights of the year | 11 |
| 6.1 Industrial transfer | 11 |
| 6.2 Keynotes | 11 |
| 7 Latest software developments, platforms, open data | 11 |
| 7.1 Latest software developments | 11 |
| 7.1.1 SynDEX | 11 |
| 7.1.2 EVT Kopernic | 12 |
| 7.2 Open data | 12 |
| 8 New results | 12 |
| 8.1 Worst case execution time estimation of a program | 13 |
| 8.2 Building measurement-based benchmarks: KDBench | 13 |
| 8.3 Scheduling of graph tasks on different resources within an energy budget | 13 |
| 9 Bilateral contracts and grants with industry | 14 |
| 9.1 CIFRE Grant funded by StatInf | 14 |
| 10 Partnerships and cooperations | 14 |
| 10.1 International initiatives | 14 |
| 10.1.1 Inria associate team not involved in an IIL or an international program | 14 |
| 10.2 International research visitors | 15 |
| 10.2.1 Visits of international scientists | 15 |
| 10.3 Public policy support | 15 |
| 11 Dissemination | 15 |
| 11.1 Animation of scientific events | 16 |
| 11.1.1 Journal | 16 |
| 11.1.2 Invited talks | 16 |
| 11.1.3 Leadership within the scientific community | 16 |
| 11.1.4 Scientific expertise | 16 |
| 11.1.5 Research administration | 16 |
| 11.2 Teaching - Supervision - Juries - Educational and pedagogical outreach | 16 |
| 11.2.1 Supervision | 17 |
| 11.2.2 Juries | 17 |

| | |
|---|-----------|
| 11.2.3 Educational and pedagogical outreach | 17 |
| 11.2.4 Participation in Live events | 17 |
| 12 Scientific production | 17 |
| 12.1 Major publications | 17 |
| 12.2 Publications of the year | 18 |
| 12.3 Cited publications | 18 |

1 Team members, visitors, external collaborators

Research Scientist

- Liliana Cucu [Team leader, INRIA, Senior Researcher, HDR]

PhD Students

- Hadjer Bendellaa [INRIA]
- Ismail Hawila [StatInf , CIFRE]
- Myriam Mabrouki [INRIA, from Nov 2025]

Technical Staff

- Masum Bin Alam [INRIA, Engineer, until Oct 2025]

Interns and Apprentices

- Artiom Fliurta [INRIA, Intern, from Jun 2025 until Jun 2025]
- Kyrylo Hrynevych [INRIA, Intern, from Jun 2025 until Aug 2025]
- Mahé James [INRIA, Intern, from Jun 2025 until Jun 2025]
- Ilian Karaguilla [INRIA, Intern, from Jun 2025 until Jun 2025]
- Myriam Mabrouki [INRIA, Intern, from Apr 2025 until Oct 2025]
- Ahmed Sehili [INRIA, Intern, from May 2025 until Aug 2025]

Administrative Assistants

- Nelly Maloysel [INRIA]
- Abigail Palma [INRIA]

External Collaborators

- Slim Ben Amor [Statinf]
- Adriana Gogonel [Statinf]
- Kossivi Koumbenou [StatInf]
- Yves Sorel [INRIA]

2 Overall objectives

The Kopernic members are focusing their research on studying **time for embedded communicating systems**, also known as cyber-physical systems. More precisely, the team proposes a **system-oriented solution** to the problem of studying time properties of the cyber components of a CPS. The solution is expected to be obtained by composing probabilistic and non-probabilistic approaches for CPSs. Moreover, statistical approaches are expected to validate existing hypotheses or propose new ones for the models considered by probabilistic analyses [3, 4].

The term cyber-physical systems refers to a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities [16]. A defibrillator, a mobile phone, an autonomous car or an aircraft, they all are CPSs. Beside constraints like power consumption,

security, size and weight, CPSs may have cyber components required to fulfill their functions within a limited time interval (a.k.a. time dependability), often imposed by the environment, e.g., a physical process controlled by some cyber components. The appearance of communication channels between cyber-physical components, easing the CPS utilization within larger systems, forces cyber components with high criticality to interact with lower criticality cyber components. This interaction is completed by external events from the environment that has a time impact on the CPS. Moreover, some programs of the cyber components may be executed on time predictable processors and other programs on less time predictable processors.

Different research communities study separately the three design phases of these systems: the modeling, the design and the analysis of CPSs [28]. These phases are repeated iteratively until an appropriate solution is found. During the first phase, the behavior of a system is often described using model-based methods. Other methods exist, but model-driven approaches are widely used by both the research and the industry communities. A solution described by a model is proved (functionally) correct usually by a formal verification method used during the analysis phase (third phase described below).

During the second phase of the design, the physical components (e.g., sensors and actuators) and the cyber components (e.g., programs, messages and embedded processors) are chosen often among those available on the market. However, due to the ever increasing pressure of smartphone market, the microprocessor industry provides general purpose processors based on multicore and, in a near future, based on manycore processors. These processors have complex architectures that are not time predictable due to features like multiple levels of caches and pipelines, speculative branching, communicating through shared memory or/and through a network on chip, internet, etc. Due to the time unpredictability of some processors, nowadays the CPS industry is facing the great challenge of estimating worst case execution times (WCETs) of programs executed on these processors. Indeed, the current complexity of both processors and programs does not allow to propose reasonable worst case bounds. Then, the phase of design ends with the implementation of the cyber components on such processors, where the models are transformed in programs (or messages for the communication channels) manually or by code generation techniques [19].

During the third phase of analysis, the correctness of the cyber components is verified at program level where the functions of the cyber component are implemented. The execution times of programs are estimated either by static analysis, by measurements or by a combination of both approaches [37].

These WCETs are then used as inputs to scheduling problems [30], the highest level of formalization for verifying the time properties of a CPS. The programs are provided a start time within the schedule together with an assignment of resources (processor, memory, communication, etc.). Verifying that a schedule and an associated assignment are a solution for a scheduling problem is known as a schedulability analysis.

The current CPS design, exploiting formal description of the models and their transformation into physical and cyber parts of the CPS, ensures that the functional behavior of the CPS is correct. Unfortunately, there is no formal description guaranteeing today that the execution times of the generated programs is smaller than a given bound. Clearly all communities working on CPS design are aware that **computing takes time** [27], but there is **no CPS solution guaranteeing time predictability** of these systems as **the processors appear late within the design phase** (see Figure 1). Indeed, the choice of the processor is made at the end of the CPS design process, after writing or generating the programs.

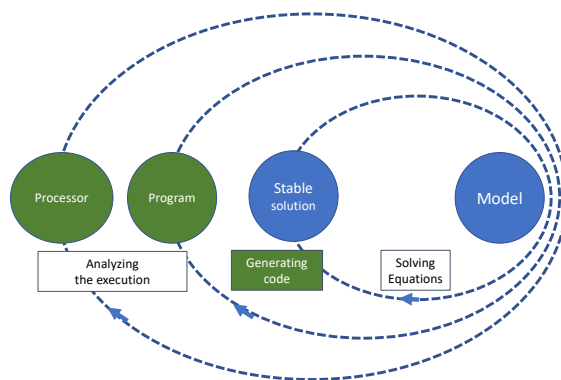


Figure 1: The CPSs design: from models towards analyzing the time properties of the cyber components

Since the processor appears late within the CPS design process, the CPS designer in charge of estimating the worst case execution time of a program or analyzing the schedulability of a set of programs inherits a difficult problem. The Kopernic main purpose is the proposition of compositional rules with respect to the time behaviour of a CPS, allowing to restrain the CPS design to analyzable instances of the WCET estimation problem and of the schedulability analysis problem.

With respect to the WCET estimation problem, we say that a rule \circ is compositional for any two sets of measured execution times C_1 and C_2 of a program A , and a WCET statistical estimator p , if we obtain a safe WCET estimation for A from $p(C_1 \circ C_2)$. For instance, C_1 may be the set of measured execution times of the program A while all processor features except the local cache L1 are deactivated, while C_2 is obtained, similarly, with a shared L2 cache activated. We consider that the variation of all input variables of the program A follows the same sequence of values, when measuring the execution time of the program p . With respect to the schedulability analysis problem, we are interested in analyzing graphs of communicating programs. A program A communicates with a program B if input variables of the program B are among output variables of the program A . A graph of communicating programs is a direct acyclic graph with programs as vertices. An edge from a program A to program B is defined if A communicates with B . The end to end response time of such graph is the longest path from any source vertex to any sink vertex of the graph, if there is, at least one path between these two vertices. A rule \odot is compositional for any set of measured response times \mathcal{R}_A of program A , any set of measured response times \mathcal{R}_B and a schedulability analysis S if we obtain a safe schedulability analysis from $S(\mathcal{R}_A \odot \mathcal{R}_B)$.

Before enumerating our scientific objectives, we introduce the concept of variability factors. More precisely, the time properties of a cyber component are subject to variability factors. We understand by variability the distance between the smallest value and the largest value of a time property. With respect to the time properties of a CPS, the factors may be classified in three main classes:

- program structure: for instance, the execution time of a program that has two main branches is obtained, if appropriate composition principles apply, as the maximum between the largest execution time of each branch. In this case the branch is a variability factor on the execution time of the program;
- processor structure: for instance, the execution time of a program on a less predictable processor (e.g., one core, two levels of cache memory and one main memory) will have a larger variability than the execution time of the same program executed on a more predictable processor (e.g., one core, one main memory). In this case the cache memory is a variability factor on the execution time of the program;
- execution environment: for instance, the appearance of a pedestrian in front of a car triggers the execution of the program corresponding to the brakes in an autonomous car. In this case the pedestrian is a variability factor for the triggering of the program.

We identify three main scientific objectives to validate our research hypothesis. The three objectives are presented from program level, where we use statistical approaches, to the level of communicating programs, where we use probabilistic and non-probabilistic approaches.

The Kopernic scientific objectives are:

- **[O1] worst case execution time estimation of a program** - modern processors induce an increased variability of the execution time of programs, making difficult (or even impossible) a complete static analysis to estimate such worst case. Our objective is to propose a solution composing probabilistic and non-probabilistic approaches based both on static and on statistical analyses by answering the following **scientific challenges**:
 1. **a classification of the variability factors of execution times** of a program with respect to the processor features. The difficulty of this challenge is related to the definition of an element belonging to the set of variability factors and its mapping to the execution time of the program.
 2. **a compositional rule** of statistical models associated to each variability factor. The difficulty of this challenge comes from the fact that a global maximum of a multicore processor cannot be obtained by upper bounding the local maxima on each core.
- **[O2] deciding the schedulability of all programs running within the same cyber component, given an energy budget** - in this case the programs may have different time criticalities, but they share the

same processor, possibly multicore¹. Our objective is to propose a solution composing probabilistic and non-probabilistic approaches based on answers to the following **scientific challenges**:

1. **scheduling algorithms taking into account the interaction between different variability factors**. The existence of time parameters described by probability distributions imposes to answer to the challenge of revisiting scheduling algorithms that lose their optimality even in the case of an uncore processor [31]. Moreover, the multicore partitioning problem is recognized difficult for the non-probabilistic case [35];
 2. **schedulability analyses** based on the algorithms proposed previously. In the case of predictable processors, the schedulability analyses accounting for operating systems costs increase the time dependability of CPSs [33]. Moreover, in presence of variability factors, the composition property of non-probabilistic approaches is lost and new principles are required.
- **[O3] deciding the schedulability of all programs communicating through predictable and non-predictable networks, given an energy budget** - in this case the programs of the same cyber component execute on the same processor and they may communicate with the programs of other cyber components through networks that may be predictable (network on chip) or non-predictable (internet, telecommunications). Our objective is to propose a solution to this challenge by analysing schedulability of programs, for which existing (worst case) probabilistic solutions exist [32], communicating through networks, for which probabilistic worst-case solutions [20] and average solutions exist [29].

3 Research program

The research program for reaching these three objectives is organized according three main research axes

- Worst case execution time estimation of a program, detailed in Section 3.1;
- Building measurement-based benchmarks, detailed in Section 3.2;
- Scheduling of graph tasks on different resources within an energy budget in Section 3.3.

3.1 Worst case execution time estimation of a program

The temporal study of real-time systems is based on the estimation of the bounds for their temporal parameters and more precisely the WCET of a program executed on a given processor. The main analyses for estimating WCETs are static analyses [37], dynamic analyses [21], also called measurement-based analyses, and finally hybrid analyses that combine the two previous ones [37].

The Kopernic approach for solving the WCET estimation problem is based on (i) the identification of the impact of variability factors on the execution of a program on a processor and (ii) the proposition of compositional rules allowing to integrate the impact of each factor within a WCET estimation. Historically, the real-time community had illustrated the distribution of execution times for programs as heavy-tailed ones as intuitively the large values of execution times of programs are agreed to have a low probability of appearance. For instance Tia et al. are the first underlining this intuition within a paper introducing execution times described by probability distributions within a single core schedulability analysis [36]. Since [36], a low probability is associated to large values of execution times of a program executed on a single core processor. It is, finally, in 2000 that the group of Alan Burns, within the thesis of Stewart Edgar [22], formalizes this property as a conjecture indicating that a maximal bound on the execution times of a program may be estimated by the Extreme Value Theory [25]. No mathematical definition of what represents this bound for the execution time of a program has been proposed at that time. Two years later, a first attempt to define this bound has been done by Bernat et al. [18], but the proposed definition is extending the static WCET understanding as a combination of execution times of basic blocks of a program. Extremely pessimistic, the definition remains intuitive, without associating a mathematical description. After 2013, several publications from Liliana Cucu-Grosjean's group at Inria Nancy introduce a mathematical definition of a probabilistic worst-case execution time, respectively, probabilistic worst-case response time, as an

¹This case is referred as a mixed criticality approach.

appropriate formalization for a correct application of the Extreme Value Theory to the real-time problems [2, 1, 5].

We identify the following open research problems related to the first research axis:

1. the generalization of modes analysis to multi-dimensional, each dimension representing a program when several programs cooperate;
2. the proposition of a rules set for building programs that are time predictable for the internal architecture of a given single core and, then, of a multicore processor;
3. modeling the impact of processor features on the energy consumption to better consider both worst case execution time and schedulability analyses considered within the third research axis of this proposal.

3.2 Building measurement-based benchmarks

The real-time community is facing the lack of benchmarks adapted to measurement-based analyses. Existing benchmarks for the estimation of WCET [34, 26, 23] have been used to estimate WCETs mainly for static analyses. They contain very simple programs and are not accompanied by a measurement protocol. They do not take into account functional dependencies between programs, mainly due to shared global variables which, of course, influence their execution times. Furthermore, current benchmarks do not take into account interferences due to the competition for resources, e.g., the memory shared by the different cores in a multicore. On the other hand, measurement-based analyses require execution times measured while executing programs on embedded processors, similar to those used in the embedded systems industry. For example, the mobile phone industry uses multicore based on non predictable cores with complex internal architecture, such as those of the ARM Cortex-A family. In a near future, these multicore will be found in critical embedded systems found in application domains such as avionics, autonomous cars, railway, etc., in which the team is deeply involved. This increases dramatically the complexity of measurement-based analyses compared to analyses performed on general purpose personal computers as they are currently performed.

We understand by measurement-based benchmarks a 3-uple composed by a program, a processor and a measurement protocol. The associated measurement protocols should detail the variation of the input variables (associated to sensors) of these benchmarks and their impact on the output variables (associated to actuators), as well as the variation of the processor states.

Proposing reproducibility and representativity properties that measurement-based benchmarks should follow is the strength of this research axis [7]. We understand by the reproducibility, the property of a measurement protocol to provide the same ordered set of execution times for a fixed pair (program, processor). We understand by the representativity, the existence of a (sufficiently small) number of values for the input variables allowing a measurement protocol to provide an ordered set of execution times that ensure a convergence for the Extreme Value Index estimators.

Within this research axis we identify the following open problems:

1. proving reproducibility and representativity properties while extending current benchmarks from predictable uncore processors (e.g., ARM Cortex-M4) to non predictable ones (e.g., ARM Cortex-A53 or Cortex-A7);
2. proving reproducibility and representativity properties while extending uncore benchmarks to multicore processors. In this context, we face the supplementary difficulty of defining the principles that an operating system should satisfy in order to ensure a real-time behaviour.

3.3 Scheduling of graph tasks on different resources within an energy budget

Following the model-driven approach, the functional description of the cyber part of the CPS, is performed as a graph of dependent functions, e.g., a block diagram of functions in Simulink, the most widely used modeling/simulation tool in industry. Of course, a program is associated to every function. Since the graph of dependent programs becomes a set of dependent tasks when real-time constraints must be taken into account, we are facing the problem of verifying the schedulability of such dependent task sets when it is executed on a multicore processor.

Directed Acyclic Graphs (DAG) are widely used to model different types of dependent task sets. The typical model consists of a set of independent tasks where every task is described by a DAG of dependent sub-tasks with the same period inherited from the period of each task [17]. In such DAG, the sub-tasks are vertices and edges are dependencies between sub-tasks. This model is well suited to represent, for example, the engine controller of a car described with Simulink. The multicore schedulability analysis may be of two types, global or partitioned. To reduce interference and interactions between sub-tasks, we focus on partitioned scheduling where each sub-task is assigned to a given core [24, 6, 8].

In order to propose a general DAG task model, we identify the following open research problems:

1. solving the schedulability problem where probabilistic DAG tasks are executed on predictable and non predictable processors, and such that some tasks communicate through predictable networks, e.g., inside a multicore or a manycore processor, and non-predictable networks, e.g., between these processors through internet. Within this general schedulability problem; we consider five main classes of scheduling algorithms that we adapt to solve probabilistic DAG task scheduling problems. We compare the new algorithms with respect to their energy-consumption in order to propose new versions with a decreased energy consumption by integrating variation of frequencies for processor features like CPU or memory accesses.
2. the validation of the proposed framework on our multicore drone case study. To answer to the challenging objective of proposing time predictable platforms for drones, we currently migrate the PX4-RT programs on heterogeneous architectures. This includes an implementation of the scheduling algorithms detailed within this research axis within current operating system, NuttX².

4 Application domains

4.1 Avionics

Time critical solutions in this context are based on temporal and spatial isolation of the programs and the understanding of multicore interferences is crucial. Our contributions belong mainly to the solutions space for the objective [O1] identified previously.

4.2 Railway

Time critical solutions in this context concern both the proposition of an appropriate scheduler and associated schedulability analyses. Our contributions belong to the solutions space of problems dealt within objectives [O1] and [O2] identified previously.

4.3 Autonomous cars

Time critical solutions in this context concern the interaction between programs executed on multicore processors and messages transmitted through wireless communication channels. Our contributions belong to the solutions space of all three classes of problems dealt within all three Kopernic objectives identified previously.

4.4 Drones

As it is the case of autonomous cars, there is an interaction between programs and messages, suggesting that our contributions in this context belong to the solutions space of all three classes of problems dealt within the objectives identified previously.

²For more details, see the NuttX [webpage](#)

5 Social and environmental responsibility

5.1 Impact of research results

The Kopernic members provide theoretical means and adequate measures to quantify decreased processor utilization and energy consumption. Such gain is estimated by comparing associated worst case execution times, respectively, worst case energy consumption. While for execution times, values are compared at probabilities like 10^{-9} per hour of functioning, there is no similar common agreement within the community for the energy consumption. Our current work covers the definition of associated probabilities for the energy consumption allowing to the community to advance towards appropriate energy (worst case) measures.

6 Highlights of the year

6.1 Industrial transfer

Statinf, the Kopernic industrial spin-off has become part of Vector Informatik GmbH, while its product, RocqStat integrates the VectorCAST, a set of tools for automated testing activities across the software development lifecycle.

6.2 Keynotes

Liliana Cucu-Grosjean has gave a keynote entitled "Why can't large data replace the time understanding of embedded products?" at the 30th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2025).

7 Latest software developments, platforms, open data

Within this section, we present two software that illustrate the results of our research, SynDEx and EVT-Kopernic. They are not updated yearly but all long term objectives of our team are expected to be integrated within them. Moreover, we provide open source dynamic benchmarks under the name of KDBench in order to ensure research reproducibility of statistical and probabilistic methods.

7.1 Latest software developments

7.1.1 SynDEx

Keywords: Distributed, Optimization, Real time, Embedded systems, Scheduling analyses

Scientific Description: SynDEx is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. It is developed in OCaml.

Architectures are represented as graphical block diagrams composed of programmable (processors) and non-programmable (ASIC, FPGA) computing components, interconnected by communication media (shared memories, links and busses for message passing). In order to deal with heterogeneous architectures it may feature several components of the same kind but with different characteristics.

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

Exploration of alternative allocations of the algorithm onto the architecture may be performed manually and/or automatically. The latter is achieved by performing real-time multiprocessor schedulability analyses and optimization heuristics based on the minimization of temporal or resource criteria. For

example while satisfying deadline and latency constraints they can minimize the total execution time (makespan) of the application onto the given architecture, as well as the amount of memory. The results of each exploration is visualized as timing diagrams simulating the distributed real-time implementation.

Finally, real-time distributed embedded code can be automatically generated for dedicated distributed real-time executives, possibly calling services of resident real-time operating systems such as Linux/RTAI or Osek for instance. These executives are deadlock-free, based on off-line scheduling policies. Dedicated executives induce minimal overhead, and are built from processor-dependent executive kernels. To this date, executive kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be achieved at reasonable cost following these examples as patterns.

Functional Description: Software for optimising the implementation of embedded distributed real-time applications and generating efficient and correct by construction code

URL: <http://www.syndex.org>

Contact: Yves Sorel

Participant: Yves Sorel

7.1.2 EVT Kopernic

Keywords: Embedded systems, Worst Case Execution Time, Real-time application, Statistics

Scientific Description: The EVT-Kopernic tool is an implementation of the Extreme Value Theory (EVT) for the problem of the statistical estimation of worst-case bounds for the execution time of a program on a processor. Our implementation uses the two versions of EVT - GEV and GPD - to propose two independent methods of estimation. Their results are compared and only results that are sufficiently close allow to validate an estimation. Our tool is proved predictable by its unique choice of block (GEV) and threshold (GPD) while proposing reproducible estimations.

Functional Description: EVT-Kopernic is tool proposing a statistical estimation for bounds on worst-case execution time of a program on a processor. The estimator takes into account dependences between execution times by learning from the history of execution, while dealing also with cases of small variability of the execution times.

URL: <http://www.statinf.fr>

Contact: Liliana Cucu

Participants: Adriana Gogonel, Liliana Cucu

7.2 Open data

Kopernic members contribute to the effort of reproducing research results and numerical evaluation by proposing dynamic benchmarks for statistical analysis of measured execution times, memory accesses and other traces obtained during the Hardware in the Loop execution of the open source programs PX4-RT. Measurements obtained from the execution of the PX4-RT programs on real boards are shared with the community under the name of KDBench, a.k.a, Kopernic Dynamic Benchmarks. The data related to KDBench are available and regularly updated at <https://team.inria.fr/kopernic/kdbench/>. The contact person is Liliana Cucu-Grosjean.

8 New results

During this year, the results of Kopernic members have covered all Kopernic research axes.

8.1 Worst case execution time estimation of a program

Participants: Liliana Cucu-Grosjean, Hadjer Bendellaa, Kyrylo Hrynevych.

We consider both WCET and WCEC (worst-case energy consumption) statistical estimators that are based on the utilization of the Extreme Value Theory [25]. Compared to existing methods [37], our results require the execution of the program under study on the targeted processor or at least a cycle-accurate simulator of this processor. We concentrate on the correct application of the Extreme Value Theory as it brings an important support to justifiable estimations.

This year we have concentrated our efforts on studying the impact of heterogeneous hardware GPU-CPU on the estimation of WCET of programs. Indeed, current real-time GPU scheduling methods depend on (non-existent) worst-case execution time estimates of programs due to the absence of a complete model for the GPU functioning. To tackle this absence, we conduct a sensitivity analysis on programs suited for execution on embedded GPU hardware. This analysis is a starting point towards the understanding of factors causing execution time variations on GPUs. We conduct a sensitivity analysis identifying primary sources influencing GPUs programs execution time among the following configuration parameters: active streaming multiprocessors, thread block size and program type. Our preliminary analysis is achieved through three main steps [9]:

- Experimental validation: reproduction of the RTGPU framework verifying measurement consistency and eliminating cloud-based deployment biases;
- Benchmark extension: transition to a realistic Kalman filter exposing larger timing variability than GPU traditional benchmarks;
- Parameter impact analysis: fine-grained, segment-based analysis to identify architectural and workload factors affecting execution time variations.

8.2 Building measurement-based benchmarks: KDBench

Participants: Masum Bin Alam, Liliana Cucu-Grosjean, Ismail Hawila, Kyrylo Hrynevych, Ahmed Sehili, Yves Sorel.

KDBench, our measurement-based benchmarks, are obtained by modifying open-source programs of the autopilot PX4 designed to control a wide range of air, land, sea and underwater drones. More precisely, the studied programs are executed on a standard Pixhawk drone board based on a predictable single core processor ARM Cortex-M4 and during several collaborative projects we have transformed this set of programs into a set of dependent tasks that satisfies real-time constraints, leading to a new version of PX4, called PX4-RT. As usual, the set of dependent real-time tasks is defined by a data dependency graph. An interested reader may refer to the web page of the Kopernic team at [the KDBench website](#).

During this year, we have finalized the migration of KDBench programs to a new board, NAVIO2 [12], while starting publicizing existing benchmarks obtained for the first board, PixHawk. The new PX4 and PX4-RT HitL simulations are executed on the RPI/Navio2 board including an ARM Cortex-A53 quadcore with three different operating systems: plain Linux with no patch (NPRT), Linux with the PREEMPT-RT (PRT) patch and Linux with the Xenomai-3 patch. While we collect measures, an user manual [12] is made available for the community to reproduce or make new measurements.

This year, we have kicked new studies integrating Network Calculus techniques to analyse the temporal series collected within the KDBench as one promising approach providing validation principles to statistical and probabilistic approaches.

8.3 Scheduling of graph tasks on different resources within an energy budget

Participants: Slim Ben Amor, Liliana Cucu-Grosjean, Ismail Hawila, Myriam Mabrouki.

Within this research axis and during this year, we provide solutions to two main problems:

- *Scheduling of control real-time tasks* is the main inspiration source for graph tasks within the real-time community and existing results for cyber-physical systems have been proposed to merge the requirements associated to the stability of the physical components and the schedulability of the cyber components. Nevertheless, none of the existing results has studied these requirements for multiple real-time cascade control tasks where their periods choice are dependent and affect stability. This year, we have proposed a methodology to evaluate the periods of the real-time cascade control tasks that ensures stability of the physical components, and presented a co-design problem for the period choice that guarantees good performance of the physical components and schedulability of the cyber components under fixed-priority scheduling. We then evaluate this methodology on a real use-case of a drone system. Our results show the importance of studying these requirements together as their relation has an impact on stable periods range. [11] [10].
- *Measurement-based energy models* do not exist to the best of our knowledge and this year we present first experimental results and measurement-based properties of programs executed on operating systems together with a real-time patch. Recognized as a source of increased average execution time and decreased worst-case execution time for programs, real-time patches do introduce new mechanisms in the operating system that may have an impact on the power dissipation of hardware components. Considering the cost of this dissipation within the design of embedded systems is an open problem due to the lack of evolution of power models for modern hardware architecture. For instance, a majority part of power models consider the memory either as a device or with a negligible impact on the power dissipation [15].

9 Bilateral contracts and grants with industry

9.1 CIFRE Grant funded by StatInf

Participants: Slim Ben Amor, Liliana Cucu-Grosjean, Ismail Hawila, Yves Sorel.

A CIFRE agreement between the Kopernic team and the start-up StatInf has started on October 1st, 2022. Its funding is related to study the relation between the control theory robustness and the schedulability problem using probabilistic descriptions according to Kopernic research objectives. The defense of the associated thesis has taken place on December 15th, 2025.

10 Partnerships and cooperations

10.1 International initiatives

10.1.1 Inria associate team not involved in an IIL or an international program

Giordano

Title: GUARANTEEING TIME AND DATA FOR EFFICIENT AUTONOMOUS CYBER-PHYSICAL SYSTEMS

Duration: 2025 -> 2027

Coordinator: George Lima (gmlima@ufba.br)

Partners:

- Universidade Federal da Bahia (Br sil)

Inria contact: Liliana Cucu

Summary: Today the term of cyber-physical systems (CPSs) refers to a new generation of systems integrating computational and physical capabilities to interact with humans. A defibrillator, a mobile phone, a car or an aircraft, they all are CPSs. Beside constraints like power consumption, security, size and weight, CPSs may have cyber components required to fulfill their functions within a limited time interval, property a.k.a safety, while hardware components moved to multicore processors, known for increasing the time variability of programs. Ensuring time predictability on multicore processors is our identified challenge. For instance, there is no program ensuring the safety critical behavior of an airplane running on multicore processors. Meanwhile, the industry has moved to digital twins for decreasing development costs and our purpose is to jointly study time constraints for original CPSs and their existing digital twins. Our hypothesis is that such validation should be data-driven. In this context, time guarantees are studied by using statistical and probabilistic modelisations. The Giordano project faces these challenges and aims at developing new mechanisms and techniques for supporting CPS applications on multicore processors, focusing on scheduling and timing analysis, for which probabilistic guarantees should be provided. The purpose of the Giordano project is to move the Kepler results from a single core to several cores using probabilistic approaches, while data-driven validations are studied for existing digital twins of these systems.

10.2 International research visitors

10.2.1 Visits of international scientists

Other international visits to the team

Arvind Easwaran

Status Associate Professor

Institution of origin: Nanyang Technological University

Country: Singapore

Dates: November 2025

Context of the visit: Consolidating of international relations with Singapore

Mobility program/type of mobility: Lecture

10.3 Public policy support

Participants: Liliana Cucu-Grosjean.

Kopernic members have strong relations with the embedded systems industry actors and associations. Within her role as academic ambassador, Liliana Cucu-Grosjean participates to working groups of the French Society Automotive Engineers [14] [13].

11 Dissemination

Participants: Liliana Cucu, Yves Sorel

11.1 Animation of scientific events

Member of the organizing committees Liliana Cucu-Grosjean has participated as scientific expert to the organization of the International Congress of the French Society of Automotive Engineers (SIA CESA 2025, Versailles)

Member of the conference program committees Kopernic members are regular PC members for relevant conferences like IEEE RTSS, IEEE RTAS, DATE, ETFA, RTNS and RTCSA.

11.1.1 Journal

Member of the editorial boards Liliana Cucu-Grosjean is associated editor of the Journal of System Architecture (JSA)

Reviewer - reviewing activities Kopernic members are regular reviewers for the main journals of our domain: Journal of Real-Time Systems, IEEE Transactions on Computer Science, Information Processing Letter, Journal of Heuristics, Journal of Systems Architecture, Journal of Signal Processing Systems, Leibniz Transactions on Embedded Systems, IEEE Transactions on Industrial Informatics, etc.

11.1.2 Invited talks

Liliana Cucu-Grosjean has gave a keynote entitled "Why can't large data replace the time understanding of embedded products?" at the 30th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2025).

11.1.3 Leadership within the scientific community

Liliana Cucu-Grosjean is the current Vice-Chair of the IEEE Technical Community on Real-Time Systems.

11.1.4 Scientific expertise

- Yves Sorel is a member of the Steering Committee of Advanced engineering and computing Hub of Systematic Paris-Region Cluster;
- Yves Sorel is a member of the Steering Committee Orientations and Programs of System X Institute for Technological Research (IRT);
- Liliana Cucu-Grosjean is a scientific expert within the French Society of Automotive Engineering
- Liliana Cucu-Grosjean is a member of the WK Embedded IA of Embedded France association.

11.1.5 Research administration

- Liliana Cucu-Grosjean is INRIA national referee on harassment at FS-CSA;
- Liliana Cucu-Grosjean is an elected member of INRIA Scientific Board.

11.2 Teaching - Supervision - Juries - Educational and pedagogical outreach

Liliana Cucu-Grosjean, Initiation to the research, MSc degree on Embedded Systems at University of Saclay

11.2.1 Supervision

- Myriam Mabrouki, Evolving energy models, Sorbonne University, started on November 2025, supervised by Liliana Cucu- Grosjean
- Hadjer Bendellaa, Dimensioning probabilistic embedded systems for efficient execution of artificial intelligence algorithms, Sorbonne University, started on November 2024, supervised by Liliana Cucu- Grosjean
- Ismail Hawila, Multicore scheduling of real-time control systems of probabilistic tasks with precedence constraints, Sorbonne university, started on October 2022 and defended on December 2025, supervised by Liliana Cucu- Grosjean and Slim Ben Amor (StatInf)

11.2.2 Juries

Liliana Cucu-Grosjean has been:

- Reviewer of the PhD thesis entitled *Mixed Criticality Mission Planning for Autonomous Robot Fleets* of Franco Petrone Cordeiro, Institut polytechnique de Paris
- Chair of the PhD defense jury of the thesis entitled *Dynamic approaches to shared memory cache management for a multi-critical system on a multi-core processor* of Al  xis G  n  res, University of Toulouse

11.2.3 Educational and pedagogical outreach

Kopernic members have hosted several interns for second degree pupils and longer stays have been organized for Mah   James, Ilian Karaguilla and Artiom Fliurta with the help of Hadjer Bendellaa, Ismail Hawila, Myriam Mabrouki and Ahmed Sehili. Liliana Cucu-Grosjean has been member of the NSI 2025 jury as well as participant to Chiche events at INRIA, Paris.

11.2.4 Participation in Live events

Liliana Cucu-Grosjean has participated to the SIA web event, SDV and open source hardware (see more details at SIA [website](#).)

12 Scientific production

12.1 Major publications

- [1] L. Cucu-Grosjean and A. Gogonel. ‘Simulation Device’. FR2016/050504 (France). Mar. 2016. URL: <https://hal.archives-ouvertes.fr/hal-01666599> (cit. on p. 9).
- [2] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Qui  nes and F. J. Cazorla. ‘Measurement-Based Probabilistic Timing Analysis for Multi-path Programs’. In: *the 24th Euromicro Conference on Real-Time Systems, ECRTS*. 2012, pp. 91–101 (cit. on p. 9).
- [3] R. Davis and L. Cucu-Grosjean. ‘A Survey of Probabilistic Schedulability Analysis Techniques for Real-Time Systems’. In: *Leibniz Transactions on Embedded Systems* 6.1 (2019), p. 53. DOI: [10.4230/LITES-v006-i001-a004](https://doi.org/10.4230/LITES-v006-i001-a004). URL: <https://inria.hal.science/hal-02158985> (cit. on p. 5).
- [4] R. Davis and L. Cucu-Grosjean. ‘A Survey of Probabilistic Timing Analysis Techniques for Real-Time Systems’. In: *Leibniz Transactions on Embedded Systems* 6.1 (2019), p. 60. DOI: [10.4230/LITES-v006-i001-a003](https://doi.org/10.4230/LITES-v006-i001-a003). URL: <https://inria.hal.science/hal-02158973> (cit. on p. 5).
- [5] A. Gogonel and L. Cucu-Grosjean. ‘Dispositif de caract  risation et/ou de mod  lisation de temps d’ex  cution pire-cas’. 1000408053 (France). June 2017. URL: <https://hal.archives-ouvertes.fr/hal-01666535> (cit. on p. 9).

- [6] T. Kloda, A. Bertout and Y. Sorel. ‘Latency analysis for data chains of real-time periodic tasks’. In: *the 23rd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA’18*. Sept. 2018 (cit. on p. 10).
- [7] C. Maxim, A. Gogonel, I. M. Asavae, M. Asavae and L. Cucu-Grosjean. ‘Reproducibility and representativity: mandatory properties for the compositionality of measurement-based WCET estimation approaches’. In: *SIGBED Review* 14.3 (2017), pp. 24–31 (cit. on p. 9).
- [8] S. E. Saidi, N. Pernet and Y. Sorel. ‘Scheduling Real-time HiL Co-simulation of Cyber-Physical Systems on Multi-core Architectures’. In: *the 24th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*. Aug. 2018 (cit. on p. 10).

12.2 Publications of the year

International peer-reviewed conferences

- [9] H. Bendellaa and L. Cucu-Grosjean. ‘Work in Progress: A WCET estimation model of programs on real-time GPUs’. In: *RTSS 2025 - 46th IEEE Real-time Systems Symposium*. Boston, United States, 2nd Dec. 2025. URL: <https://inria.hal.science/hal-05457002> (cit. on p. 13).
- [10] I. Hawila, L. Cucu-Grosjean and S. Ben Amor. ‘Period Assignment for Real-Time Cascade Control Tasks Under Stability and Schedulability Constraints’. In: *ECRTS 2025 - 37th Euromicro Conference on Real-Time Systems*. Bruxelles, Belgium: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 7th July 2025. DOI: [10.4230/LIPIcs.ECRTS.2025.7](https://doi.org/10.4230/LIPIcs.ECRTS.2025.7). URL: <https://inria.hal.science/hal-05456985> (cit. on p. 14).

Doctoral dissertations and habilitation theses

- [11] I. Hawila. ‘Scheduling of Probabilistic Real-Time Control Task Systems with Precedence Constraints’. Inria Paris, 15th Dec. 2025. URL: <https://hal.science/tel-05489302> (cit. on p. 14).

Reports & preprints

- [12] M. Bin Alam and Y. Sorel. *Drone Autopilot PX4 and PX4-RT HiL simulations User Manual*. INRIA, 2025, https://gitlab.inria.fr/kdbench/multicore/documents/\bibrangedash/blob/main/userManual/userManual.pdf?ref_type=heads. URL: <https://hal.science/hal-05489396> (cit. on p. 13).

Other scientific publications

- [13] L. Cucu-Grosjean. *Invitation à la Journée d’étude : Study Day Future Profing SDV, éditorial*. Dec. 2025. URL: <https://inria.hal.science/hal-05488984> (cit. on p. 15).
- [14] L. Cucu-Grosjean, P. Cuenot and E. Hamelin. *White Paper on Open Source Hardware*. 2025. URL: <https://inria.hal.science/hal-05488927> (cit. on p. 15).
- [15] M. Mabrouki, L. Cucu-Grosjean and S. Plassart. ‘Towards power dissipation estimation for real-time operating system patches’. In: *JRWRTC 2025 - the 18th Junior Researcher Workshop on Real-Time Computing*. Pisa, Italy, 5th Nov. 2025. URL: <https://inria.hal.science/hal-05457029> (cit. on p. 14).

12.3 Cited publications

- [16] R. Baheti and H. Gill. *Cyber-physical systems*. IEEE, 2011 (cit. on p. 5).
- [17] S. Baruah, V. Bonifaci, A. Marchetti-Spaccamela, L. Stougie and A. Wiese. ‘A Generalized Parallel Task Model for Recurrent Real-time Processes’. In: *2012 IEEE 33rd Real-Time Systems Symposium (RTSS)*. 2012, pp. 63–72 (cit. on p. 10).

- [18] G. Bernat, A. Colin and S. M. Petters. ‘WCET Analysis of Probabilistic Hard Real-Time System’. In: *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS’02)*. IEEE Computer Society, 2002, pp. 279–288 (cit. on p. 8).
- [19] T. Bourke, J. Colaço, B. Pagano, C. Pasteur and M. Pouzet. ‘A Synchronous-Based Code Generator for Explicit Hybrid Systems Languages’. In: *Compiler Construction - 24th International Conference, CC, Joint with ETAPS*. 2015, pp. 69–88 (cit. on p. 6).
- [20] L. Cucu. ‘Preliminary results for introducing dependent random variables in stochastic feasibility analysis on CAN’. In: *the WIP session of the 7th IEEE International Workshop on Factory Communication Systems (WFCS)*. 2008 (cit. on p. 8).
- [21] R. I. Davis and L. Cucu-Grosjean. ‘A Survey of Probabilistic Timing Analysis Techniques for Real-Time Systems’. In: *LITES 6.1* (2019), 03:1–03:60 (cit. on p. 8).
- [22] S. Edgar and A. Burns. ‘Statistical Analysis of WCET for Scheduling’. In: *the 22nd IEEE Real-Time Systems Symposium (RTSS)*. 2001, pp. 215–225 (cit. on p. 8).
- [23] H. Falk, S. Altmeyer, P. Hellinckx, B. Lisper, W. Puffitsch, C. Rochange, M. Schoeberl, R. B. Sorensen, P. Wägemann and S. Wegener. ‘TACLeBench: A Benchmark Collection to Support Worst-Case Execution Time Research’. In: *16th International Workshop on Worst-Case Execution Time Analysis (WCET)*. Vol. 55. OASICS. 2016, 2:1–2:10 (cit. on p. 9).
- [24] J. Fonseca, G. Nelissen, V. Nelis and L. Pinho. ‘Response time analysis of sporadic DAG tasks under partitioned scheduling’. In: *11th IEEE Symposium on Industrial Embedded Systems (SIES)*. May 2016, pp. 1–10 (cit. on p. 10).
- [25] S. J. Gil, I. Bate, G. Lima, L. Santinelli, A. Gogonel and L. Cucu-Grosjean. ‘Open Challenges for Probabilistic Measurement-Based Worst-Case Execution Time’. In: *Embedded Systems Letters* 9.3 (2017), pp. 69–72 (cit. on pp. 8, 13).
- [26] J. Gustafsson, A. Betts, A. Ermedahl and B. Lisper. ‘The Mälardalen WCET Benchmarks: Past, Present And Future’. In: *10th International Workshop on Worst-Case Execution Time Analysis (WCET)*. Vol. 15. OASICS. 2010, pp. 136–146 (cit. on p. 9).
- [27] E. Lee. ‘Computing Needs Time’. In: *Communications of ACM* 52.5 (2009) (cit. on p. 6).
- [28] E. Lee and S. Seshia. *Introduction to embedded systems - a cyber-physical systems approach*. MIT Press, 2017 (cit. on p. 6).
- [29] J. Lehoczky. ‘Real-Time Queueing Theory’. In: *the 10th IEEE Real-Time Systems Symposium (RTSS)*. 1996 (cit. on p. 8).
- [30] S. B. M. Bertogna and G. Buttazzo. *Multiprocessor Scheduling for Real-Time Systems*. Springer, 2015 (cit. on p. 6).
- [31] D. Maxim, O. Buffet, L. Santinelli, L. Cucu-Grosjean and R. I. Davis. ‘Optimal Priority Assignment Algorithms for Probabilistic Real-Time Systems’. In: *the 19th International Conference on Real-Time and Network Systems (RTNS)*. 2011 (cit. on p. 8).
- [32] D. Maxim and L. Cucu-Grosjean. ‘Response Time Analysis for Fixed-Priority Tasks with Multiple Probabilistic Parameters’. In: *the IEEE Real-Time Systems Symposium (RTSS)*. 2013 (cit. on p. 8).
- [33] F. Ndoye and Y. Sorel. ‘Monoprocessor Real-Time Scheduling of Data Dependent Tasks with Exact Preemption Cost for Embedded Systems’. In: *the 16th IEEE International Conference on Computational Science and Engineering (CSE)*. 2013 (cit. on p. 8).
- [34] F. Nemer, H. Cassé, P. Sainrat, J. P. Bahsoun and M. D. Michiel. ‘PapaBench: a Free Real-Time Benchmark’. In: *6th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis*. Vol. 4. OASICS. 2006 (cit. on p. 9).
- [35] S. E. Saidi, N. Pernet and Y. Sorel. ‘Automatic Parallelization of Multi-Rate FMI-based Co-Simulation On Multi-core’. In: *the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*. 2017 (cit. on p. 8).
- [36] T. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L. Wu and J. Liu. ‘Probabilistic Performance Guarantee for Real-Time Tasks with Varying Computation Times’. In: *IEEE Real-Time and Embedded Technology and Applications Symposium*. 1995 (cit. on p. 8).

- [37] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, G. Staschulat and P. Stenström. ‘The worst-case execution time problem: overview of methods and survey of tools’. In: *Trans. on Embedded Computing Systems* 7.3 (2008), pp. 1–53 (cit. on pp. 6, 8, 13).