IN PARTNERSHIP WITH:
Institut polytechnique de Grenoble

Activity Report 2017

Project-Team SPADES

Sound Programming of Adaptive Dependable Embedded Systems

IN COLLABORATION WITH: Laboratoire d'Informatique de Grenoble (LIG)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Embedded and Real-time Systems

Table of contents

Project-Team SPADES

Creation of the Team: 2013 January 01, updated into Project-Team: 2015 July 01

Keywords:

Computer Science and Digital Science:

A1.1.1. - Multicore, Manycore
A1.1.9. - Fault tolerant systems
A1.3. - Distributed Systems
A2.1.1. - Semantics of programming languages
A2.1.6. - Concurrent programming
A2.1.8. - Synchronous languages
A2.3. - Embedded and cyber-physical systems
A2.3.1. - Embedded systems
A2.3.2. - Cyber-physical systems
A2.3.3. - Real-time systems
A2.4.1. - Analysis
A2.4.3. - Proofs
A2.5.2. - Component-based Design
A7.3. - Computational models

Other Research Topics and Application Domains:

B5.2.1. - Road vehicles
B6.3.3. - Network Management
B6.4. - Internet of things
B6.6. - Embedded systems

1. Personnel

Research Scientists

Gregor Goessler [Team leader, Inria, Researcher, HDR]
Pascal Fradet [Inria, Researcher, HDR]
Alain Girault [Inria, Senior Researcher, HDR]
Sophie Quinton [Inria, Researcher]
Jean-Bernard Stefani [Inria, Researcher]

Faculty Member

Xavier Nicollin [Institut polytechnique de Grenoble, Associate Professor]

Post-Doctoral Fellow

Lijun Shan [Inria, until Apr 2017]

PhD Students

Xiaojie Guo [Univ Grenoble Alpes]
Maxime Lesourd [Univ Grenoble Alpes, from Sep 2017]
Stephan Plassart [Univ Grenoble Alpes]
Christophe Prévot [Thales Research and Technology]
Arash Shafiei [Inria then Orange Labs, from Jun 2017]
Martin Vassor [Inria, from Nov 2017]

Interns

Leila Jamshidian Sales [Inria, from Feb 2017 until Jul 2017]
Maxime Lesourd [Ecole Normale Supérieure Lyon, from Apr 2017 until Sep 2017]
Ebrahim Naeimimoshirian [Inria, from Feb 2017 until Aug 2017]
Louise Penz [Inria, from Jun 2017 until Jul 2017]
Baptiste Pollien [Inria, from Jun 2017 until Jul 2017]
Martin Vassor [Inria, from Feb 2017 until Aug 2017]

Administrative Assistant

Helen Pouchot-Rouge-Blanc [Inria]

Visiting Scientist

Roopak Sinha [Auckland University of Technology (AUT), from May 2017 until Jun 2017]

2. Overall Objectives

2.1. Overall Objectives

The SPADES project-team aims at contributing to meet the challenge of designing and programming dependable embedded systems in an increasingly distributed and dynamic context. Specifically, by exploiting formal methods and techniques, SPADES aims to answer three key questions:

1. How to program open networked embedded systems as dynamic adaptive modular structures?
2. How to program reactive systems with real-time and resource constraints on multicore architectures?
3. How to program reliable, fault-tolerant embedded systems with different levels of criticality?

These questions above are not new, but answering them in the context of modern embedded systems, which are increasingly distributed, open and dynamic in nature [27], makes them more pressing and more difficult to address: the targeted system properties – dynamic modularity, time-predictability, energy efficiency, and fault-tolerance – are largely antagonistic (*e.g.*, having a highly dynamic software structure is at variance with ensuring that resource and behavioral constraints are met). Tackling these questions together is crucial to address this antagonism, and constitutes a key point of the SPADES research program.

A few remarks are in order:

- We consider these questions to be central in the construction of future embedded systems, dealing as they are with, roughly, software architecture and the provision of real-time and fault-tolerance guarantees. Building a safety-critical embedded system cannot avoid dealing with these three concerns.
- The three questions above are highly connected. For instance, composability along time, resource consumption and reliability dimensions are key to the success of a component-based approach to embedded systems construction.
- For us, “Programming” means any constructive process to build a running system. It can encompass traditional programming as well as high-level design or “model-based engineering” activities, provided that the latter are supported by effective compiling tools to produce a running system.
- We aim to provide semantically sound programming tools for embedded systems. This translates into an emphasis on formal methods and tools for the development of provably dependable systems.

3. Research Program

3.1. Introduction

The SPADES research program is organized around three main themes, *Components and contracts*, *Real-time multicore programming*, and *Language-based fault tolerance*, that seek to answer the three key questions identified in Section 2.1. We plan to do so by developing and/or building on programming languages and techniques based on formal methods and formal semantics (hence the use of *"sound programming"* in the project-team title). In particular, we seek to support design where correctness is obtained by construction, relying on proven tools and verified constructs, with programming languages and programming abstractions designed with verification in mind.

3.2. Components and Contracts

Component-based construction has long been advocated as a key approach to the "correct-by-construction" design of complex embedded systems [56]. Witness component-based toolsets such as UC Berkeley's PTOLEMY [43], Verimag's BIP [31], or the modular architecture frameworks used, for instance, in the automotive industry (AUTOSAR) [23]. For building large, complex systems, a key feature of component-based construction is the ability to associate with components a set of *contracts*, which can be understood as rich behavioral types that can be composed and verified to guarantee a component assemblage will meet desired properties. The goal in this theme is to study the formal foundations of the component-based construction of embedded systems, to develop component and contract theories dealing with real-time, reliability and fault-tolerance aspects of components, and to develop proof-assistant-based tools for the computer-aided design and verification of component-based systems.

Formal models for component-based design are an active area of research (see *e.g.*, [24], [25]). However, we are still missing a comprehensive formal model and its associated behavioral theory able to deal *at the same time* with different forms of composition, dynamic component structures, and quantitative constraints (such as timing, fault-tolerance, or energy consumption). Notions of contracts and interface theories have been proposed to support modular and compositional design of correct-by-construction embedded systems (see *e.g.*, [34], [35] and the references therein), but having a comprehensive theory of contracts that deals with all the above aspects is still an open question [62]. In particular, it is not clear how to accomodate different forms of composition, reliability and fault-tolerance aspects, or to deal with evolving component structures in a theory of contracts.

Dealing in the same component theory with heterogeneous forms of composition, different quantitative aspects, and dynamic configurations, requires to consider together the three elements that comprise a component model: behavior, structure and types. *Behavior* refers to behavioral (interaction and execution) models that characterize the behavior of components and component assemblages (*e.g.*, transition systems and their multiple variants – timed, stochastic, etc.). *Structure* refers to the organization of component assemblages or configurations, and the composition operators they involve. *Types* refer to properties or contracts that can be attached to components and component interfaces to facilitate separate development and ensure the correctness of component configurations with respect to certain properties. Taking into account dynamicity requires to establish an explicit link between behavior and structure, as well as to consider higher-order systems, both of which have a direct impact on types.

We plan to develop our component theory by progressing on two fronts: component calculi, and semantical framework. The work on typed component calculi aims to elicit process calculi that capture the main insights of component-based design and programming and that can serve as a bridge towards actual architecture description and programming language developments. The work on the semantical framework should, in the longer term, provide abstract mathematical models for the more operational and linguistic analysis afforded by component calculi. Our work on component theory will find its application in the development of a COQ-based toolchain for the certified design and construction of dependable embedded systems, which constitutes our third main objective for this axis.

3.3. Real-Time Multicore Programming

Programming real-time systems (*i.e.*, systems whose correct behavior depends on meeting timing constraints) requires appropriate languages (as exemplified by the family of synchronous languages [33]), but also the support of efficient scheduling policies, execution time and schedulability analyses to guarantee real-time constraints (*e.g.*, deadlines) while making the most effective use of available (processing, memory, or networking) resources. Schedulability analysis involves analyzing the worst-case behavior of real-time tasks under a given scheduling algorithm and is crucial to guarantee that time constraints are met in any possible execution of the system. Reactive programming and real-time scheduling and schedulability for multiprocessor systems are old subjects, but they are nowhere as mature as their uniprocessor counterparts, and still feature a number of open research questions [29], [41], in particular in relation with mixed criticality systems. The main goal in this theme is to address several of these open questions.

We intend to focus on two issues: multicriteria scheduling on multiprocessors, and schedulability analysis for real-time multiprocessor systems. Beyond real-time aspects, multiprocessor environments, and multicore ones in particular, are subject to several constraints *in conjunction*, typically involving real-time, reliability and energy-efficiency constraints, making the scheduling problem more complex for both the offline and the online cases. Schedulability analysis for multiprocessor systems, in particular for systems with mixed criticality tasks, is still very much an open research area.

Distributed reactive programming is rightly singled out as a major open issue in the recent, but heavily biased (it essentially ignores recent research in synchronous and dataflow programming), survey by Bainomugisha et al. [29]. For our part, we intend to focus on two questions: devising synchronous programming languages for distributed systems and precision-timed architectures, and devising dataflow languages for multiprocessors supporting dynamicity and parametricity while enjoying effective analyses for meeting real-time, resource and energy constraints in conjunction.

3.4. Language-Based Fault Tolerance

Tolerating faults is a clear and present necessity in networked embedded systems. At the hardware level, modern multicore architectures are manufactured using inherently unreliable technologies [37], [47]. The evolution of embedded systems towards increasingly distributed architectures highlighted in the introductory section means that dealing with partial failures, as in Web-based distributed systems, becomes an important issue. While fault-tolerance is an old and much researched topic, several important questions remain open: automation of fault-tolerance provision, composable abstractions for fault-tolerance, fault diagnosis, and fault isolation.

The first question is related to the old question of “system structure for fault-tolerance” as originally discussed by Randell for software fault tolerance [68], and concerns in part our ability to clearly separate fault-tolerance aspects from the design and programming of purely “functional” aspects of an application. The classical arguments in favor of a clear separation of fault-tolerance concerns from application code revolve around reduced code and maintenance complexity [42]. The second question concerns the definition of appropriate abstractions for the modular construction of fault-tolerant embedded systems. The current set of techniques available for building such systems spans a wide range, including exception handling facilities, transaction management schemes, rollback/recovery schemes, and replication protocols. Unfortunately, these different techniques do not necessarily compose well – for instance, combining exception handling and transactions is non trivial, witness the flurry of recent work on the topic, see *e.g.*, [55] and the references therein –, they have no common semantical basis, and they suffer from limited programming language support. The third question concerns the identification of causes for faulty behavior in component-based assemblages. It is directly related to the much researched area of fault diagnosis, fault detection and isolation [57].

We intend to address these questions by leveraging programming language techniques (programming constructs, formal semantics, static analyses, program transformations) with the goal to achieve provable fault-tolerance, *i.e.*, the construction of systems whose fault-tolerance can be formally ensured using verification tools and proof assistants. We aim in this axis to address some of the issues raised by the above open questions by using aspect-oriented programming techniques and program transformations to automate the inclusion of fault-tolerance in systems (software as well as hardware), by exploiting reversible programming models to investigate composable recovery abstractions, and by leveraging causality analyses to study fault-ascription in component-based systems. Compared to the huge literature on fault-tolerance in general, in particular in the systems area (see *e.g.*, [49] for an interesting but not so recent survey), we find by comparison much less work exploiting formal language techniques and tools to achieve or support fault-tolerance. The works reported in [36], [39], [40], [44], [58], [67], [73] provide a representative sample of recent such works.

A common theme in this axis is the use and exploitation of causality information. Causality, *i.e.*, the logical dependence of an effect on a cause, has long been studied in disciplines such as philosophy [63], natural sciences, law [64], and statistics [65], but it has only recently emerged as an important focus of research in computer science. The analysis of logical causality has applications in many areas of computer science. For instance, tracking and analyzing logical causality between events in the execution of a concurrent system is required to ensure reversibility [61], to allow the diagnosis of faults in a complex concurrent system [51], or to enforce accountability [60], that is, designing systems in such a way that it can be determined without ambiguity whether a required safety or security property has been violated, and why. More generally, the goal of fault-tolerance can be understood as being to prevent certain causal chains from occurring by designing systems such that each causal chain either has its premises outside of the fault model (*e.g.*, by introducing redundancy [49]), or is broken (*e.g.*, by limiting fault propagation [69]).

4. Application Domains

4.1. Industrial Applications

Our applications are in the embedded system area, typically: transportation, energy production, robotics, telecommunications, the Internet of things (IoT), systems on chip (SoC). In some areas, safety is critical, and motivates the investment in formal methods and techniques for design. But even in less critical contexts, like telecommunications and multimedia, these techniques can be beneficial in improving the efficiency and the quality of designs, as well as the cost of the programming and the validation processes.

Industrial acceptance of formal techniques, as well as their deployment, goes necessarily through their usability by specialists of the application domain, rather than of the formal techniques themselves. Hence, we are looking to propose domain-specific (but generic) realistic models, validated through experience (*e.g.*, control tasks systems), based on formal techniques with a high degree of automation (*e.g.*, synchronous models), and tailored for concrete functionalities (*e.g.*, code generation).

4.2. Industrial Design Tools

The commercially available design tools (such as UML with real-time extensions, MATLAB/ SIMULINK/ dSPACE [1]) and execution platforms (OS such as VXWORKS, QNX, real-time versions of LINUX ...) start now to provide, besides their core functionalities, design or verification methods. Some of them, founded on models of reactive systems, come close to tools with a formal basis, such as for example STATEMATE by iLOGIX.

Regarding the synchronous approach, commercial tools are available: SCADE [2] (based on LUSTRE), CONTROLBUILD and RT-BUILDER (based on SIGNAL) from GEENSYS [3] (part of DASSAULT SYSTEMES), specialized environments like CELLCONTROL for industrial automatism (by the INRIA spin-off ATHYS– now part of DASSAULT SYSTEMES). One can observe that behind the variety of actors, there is a real consistency of the synchronous technology, which makes sure that the results of our work related to the synchronous approach are not restricted to some language due to compatibility issues.

[1] http://www.dspaceinc.com
[2] http://www.esterel-technologies.com
[3] http://www.geensoft.com

4.3. Current Industrial Cooperations

Regarding applications and case studies with industrial end-users of our techniques, we cooperate with Thales on schedulability analysis for evolving or underspecified real-time embedded systems, with Orange Labs on software architecture for cloud services and with Daimler on reduction of nondeterminism and analysis of deadline miss models for the design of automotive systems.

5. New Software and Platforms

5.1. pyCPA_TCA

FUNCTIONAL DESCRIPTION: We are developing pyCPA_TCA , a pyCPA plugin for Typical Worst-Case Analysis as described in Section. pyCPA is an open-source Python implementation of Compositional Performance Analysis developed at TU Braunschweig, which allows in particular response-time analysis. pyCPA_TCA is an extension of this tool that is co-developed by Sophie Quinton and Zain Hammadeh at TU Braunschweig. It allows in particular the computation of weakly-hard guarantees for real-time tasks, i.e. number of deadline misses out of a sequence of executions. So far, pyCPA_TCA is restricted to uniprocessor systems of independent tasks, scheduled according to static priority scheduling.

- Contact: Sophie Quinton

5.2. LDDL

Coq proofs of circuit transformations for fault-tolerance
KEYWORDS: Fault-tolerance - Transformation - Coq - Semantics
FUNCTIONAL DESCRIPTION: We have been developing a Coq-based framework to formally verify the functional and fault-tolerance properties of circuit transformations. Circuits are described at the gate level using LDDL, a Low-level Dependent Description Language inspired from muFP. Our combinator language, equipped with dependent types, ensures that circuits are well-formed by construction (gates correctly plugged, no dangling wires, no combinational loops, . . .). Faults like Single-Event Upsets (SEUs) (i.e., bit-flips in flipflops) and SETs (i.e., glitches propagating in the combinational circuit) and fault-models like "at most 1 SEU or SET within n clock cycles" are described in the operational semantics of LDDL. Fault-tolerance techniques are described as transformations of LDDL circuits.

The framework has been used to prove the correctness of three fault-tolerance techniques: TMR, TTR and DTR. The size of specifications and proofs for the common part (LDDL syntax and semantics, libraries) is 5000 lines of Coq (excluding comments and blank lines), 700 for TMR, 3500 for TTR and 7000 for DTR.

- Authors: Pascal Fradet and Dmitry Burlyaev
- Contact: Pascal Fradet
- URL: https://team.inria.fr/spades/fthwproofs/

6. New Results

6.1. Components and contracts

Participants: Alain Girault, Christophe Prévot, Sophie Quinton, Jean-Bernard Stefani.

6.1.1. Contracts for the negotiation of embedded software updates

We address the issue of change during design and after deployment in safety-critical embedded system applications, in collaboration with Thales and also in the context of the CCC project (http://ccc-project.org/).

In collaboration with Thales, we mostly focus on timing aspects with the objective to anticipate, at design time, future software evolutions and identify potential schedulability bottlenecks. This year we have paved the way for an extension, to more complex systems, of the approach developed last year to quantify the flexibility of a system with respect to timing. Specifically, we have focused on systems with task chains, and have proposed new methods for computing upper and lower bounds on task chain latencies. This work will be submitted to a conference early 2018. Our methods are also being implemented in the Thales tool chain, in order to be used in industry.

6.1.2. Location graphs

The design of configurable systems can be streamlined and made more systematic by adopting a component-based structure, as demonstrated with the FRACTAL component model [38]. However, the formal foundations for configurable component-based systems, featuring higher-order capabilities where components can be dynamically instantiated and passivated, and non-hierarchical structures where components can be contained in different composites at the same time, are still an open topic. We have recently introduced the location graph model [70], where components are understood as graphs of locations hosting higher-order processes, and where component structures can be arbitrary graphs.

We have continued the development of location graphs, revisiting the underlying structural model (hypergraphs instead of graphs), and simplifying its operational semantics while preserving the model expressivity. Towards the development of a behavioral theory of location graphs, we have defined different notions of bisimilarity for location graphs and shown them to be congruences, although a fully fledged co-inductive characterization of contextual equivalence for location graphs is still in the works. This work has not yet been published.

6.2. Real-Time multicore programming

Participants: Pascal Fradet, Alain Girault, Gregor Goessler, Xavier Nicollin, Sophie Quinton.

6.2.1. Dynamicity in dataflow models

Recent dataflow programming environments support applications whose behavior is characterized by dynamic variations in resource requirements. The high expressive power of the underlying models (*e.g.*, Kahn Process Networks or the CAL actor language) makes it challenging to ensure predictable behavior. In particular, checking *liveness* (*i.e.*, no part of the system will deadlock) and *boundedness* (*i.e.*, the system can be executed in finite memory) is known to be hard or even undecidable for such models. This situation is troublesome for the design of high-quality embedded systems. In the past few years, we have proposed several parametric dataflow models of computation.

We have written a survey that provides a comprehensive description of the existing parametric dataflow MoCs (constructs, constraints, properties, static analyses) and compares them using a common example [10]. The main objectives are to help designers of streaming applications to choose the most suitable model for their needs and to pave the way for the design of new parametric MoCs.

We have studied *symbolic* analyses of dataflow graphs [11]. Symbolic analyses express the system performance as a function of parameters (*i.e.*, input and output rates, execution times). Such functions can be quickly evaluated for each different configuration or checked *w.r.t.* different quality-of-service requirements. These analyses are useful for parametric MoCs, partially specified graphs, and even for completely static SDF graphs. Our analyses compute the maximal throughput of acyclic synchronous dataflow graphs, the minimum required buffers for which as soon as possible (asap) scheduling achieves this throughput, and finally the corresponding input-output latency of the graph.

We have proposed an original method to deal with lossy communication channels in dataflow graphs. Lossy channels intrinsically violate the dataflow model of computation. Yet, many real-life applications encounter some form of lossy channels, for instance IoT applications. The challenge that is raised is how to manage the retransmissions in case of lost or corrupted tokens. The solution that we have proposed involves decomposing the execution of the dataflow graph into three phases: (i) an upstream phase where all the actors before the lossy channel are executed as usual; (ii) a lossy phase where only the two actors linked by the lossy channel are executed, as many times as required until all the tokens are correctly transmitted; and (iii) a downstream phase where all the actors after the lossy channel are executed as usual. When a graph includes several lossy channels, things become more complex. We rely on the Boolean parameters of BPDF [32] to encode enabling conditions on channels so that the execution follows this upstream-lossy-downstream semantics [12].

We are now studying models allowing dynamic reconfigurations of the *topology* of the dataflow graphs. This would be of interest for C-RAN and 5G telecommunication applications. This is one of the research topic of Arash Shafiei's PhD in collaboration with Orange Labs.

6.2.2. Synthesis of switching controllers using approximately bisimilar multiscale abstractions

The use of discrete abstractions for continuous dynamics has become standard in hybrid systems design (see *e.g.*, [71] and the references therein). The main advantage of this approach is that it offers the possibility to leverage controller synthesis techniques developed in the areas of supervisory control of discrete-event systems [66]. The first attempts to compute discrete abstractions for hybrid systems were based on traditional systems behavioral relationships such as simulation or bisimulation, initially proposed for discrete systems most notably in the area of formal methods. These notions require inclusion or equivalence of observed behaviors which is often too restrictive when dealing with systems observed over metric spaces. For such systems, a more natural abstraction requirement is to ask for closeness of observed behaviors. This leads to the notions of approximate simulation and bisimulation introduced in [45].

These approaches are based on sampling of time and space where the sampling parameters must satisfy some relation in order to obtain abstractions of a prescribed precision. In particular, the smaller the time sampling parameter, the finer the lattice used for approximating the state-space; this may result in abstractions with a very large number of states when the sampling period is small. However, there are a number of applications where sampling has to be fast; though this is generally necessary only on a small part of the state-space. We have been exploring two approaches to overcome this state-space explosion [4].

We are currently investigating an approach using mode sequences of given length as symbolic states for our abstractions. By using mode sequences of variable length we are able to adapt the granularity of our abstraction to the dynamics of the system, so as to automatically trade off precision against controllability of the abstract states.

6.2.3. Schedulability of weakly-hard real-time systems

We focus on the problem of computing tight deadline miss models for real-time systems, which bound the number of potential deadline misses in a given sequence of activations of a task. In practical applications, such guarantees are often sufficient because many systems are in fact not hard real-time [3].

We have developed an extension of sensitivity analysis for budgeting in the design of weakly-hard real-time systems [18]. During design, it often happens that some parts of a task set are fully specified while other parameters, *e.g.*, regarding recovery or monitoring tasks, will be available only much later. In such cases, sensitivity analysis can help anticipate how these missing parameters can influence the behavior of the whole system so that a resource budget can be allocated to them. We have developed an extension of sensitivity analysis for deriving task budgets for systems with hard and weakly-hard requirements. This approach has been validated on synthetic test cases and a realistic case study given by our partner Thales.

A second contribution in this area is the application of our method for computing deadline miss models, called Typical Worst-Case Analysis (TWCA), to systems with finite queue capacity [9]. Finite ready queues, implemented by buffers, are a system reality in embedded real-time computing systems and networks. The dimensioning of queues is subject to constraints in industrial practice, and often the queue capacity is sufficient for typical system behavior, but is not sufficient in peak overload conditions. This may lead to overflow and consequently to the discarding of jobs. In this paper, we explore whether finite queue capacity can also be used as a mean of design in order to reduce workload peaks and thus shorten a transient overload phase. We have proposed an analysis method which is to the best of our knowledge the first one able to give (a) worst-case response times guarantees as well as (b) weakly-hard guarantees for tasks which are executed on a computing system with finite queues. Experimental results show that finite queue capacity may only have weak overload limiting effect. This unexpected outcome can be explained by the system behavior in the worst-case corner cases. The analysis shows nevertheless that a trade-off between weakly-hard guarantees and queue sizes is possible.

Finally, in collaboration with TU Braunschweig and Daimler we have worked on the application of the Logical Execution Time (LET) paradigm, according to which data are read and written at predefined time instants, to the automotive industry. Specifically, we have bridged the gap between LET, as it was originally proposed [59], and its current use in the automotive industry. One interesting outcome of this research is that it can nicely be combined with the use of TWCA. This work has not been published yet.

6.2.4. A Markov Decision Process approach for energy minimization policies

In the context of independent real-time sporadic jobs running on a single-core processor equipped with Dynamic Voltage and Frequency Scaling (DVFS), we have proposed a Markov Decision Process approach (MDP) to compute the scheduling policy that dynamically chooses the voltage and frequency level of the processor such that each job meets its deadline and the total energy consumption is minimized. We distinguish two cases: the finite case (there is a fixed time horizon) and the infinite case. In the finite case, several *offline* solutions exist, which all use the complete knowledge of all the jobs that will arrive within the time horizon [74], *i.e.*, their size and deadlines. But clearly this is unrealistic in the embedded context where the characteristics of the jobs are not known in advance. Then, an optimal offline policy called Optimal Available (OA) has been proposed in [30]. Our goal was to improve this result by taking into account the *statistical characteristics* of the upcoming jobs. When such information is available (for instance by profiling the jobs based on execution traces), we have proposed several speed policies that optimize the *expected* energy consumption. We have shown that this general constrained optimization problem can be modeled as an unconstrained MDP by choosing a proper state space that also encodes the constraints of the problem. In particular, this implies that the optimal speed at each time can be computed using a *dynamic programming* algorithm, and that the optimal speed at any time t will be a deterministic function of the current state at time t [21]. This is the topic of Stephan Plassart's PhD, funded by the CASERM Persyval project.

6.2.5. Formal proofs for schedulability analysis of real-time systems

We have started to lay the foundations for computer-assisted formal verification of schedulability analysis results. Specifically, we contribute to Prosa [26], a foundational Coq library of reusable concepts and proofs for real-time schedulability analysis. A key scientific challenge is to achieve a modular structure of proofs for response time analysis. We intend to use this library for:

1. a better understanding of the role played by some assumptions in existing proofs;
2. a formal comparison of different analysis techniques; and
3. the verification of proof certificates generated by instrumenting (existing and efficient) analysis tools.

Two schedulability analyses for uniprocessor systems have been formalized and mechanically verified in Coq for:

- sporadic task sets scheduled according to the Time Division Multiple Access (TDMA) policy.
- periodic task sets with offsets scheduled according to the Fixed Priority Preemptive (FPP) policy [15].

The analysis for TDMA has mainly served to familiarize ourselves with the Prosa library. Schedulability analysis in presence of offsets is a non-trivial problem with a high computational complexity. In contrast to the traditional (offset oblivious) analysis, many scenarios must be tested and compared to identify which one represents the worst-case scenario. We have formalized and proved in Coq the basic analysis presented by Tindell [72]. This has allowed us to: (1) underline implicit assumptions made in Tindell's informal analysis; (2) ease the generalization of the verified analysis; (3) generate a certifier and an analyzer. We are investigating these two tools in terms of computational complexity and implementation effort, in order to provide a good solution to guarantee schedulability of industrial systems.

In parallel, we have worked on a Coq formalization of Typical Worst Case Analysis (TWCA). We aim to provide certified generic results for weakly-hard real-time systems in the form of (m, k) guarantees (a task may miss at most m deadlines out of k consecutive activations). So far, we have adapted the initial TWCA for arbitrary schedulers. The proof relies on a practical definition of the concept of busy window which amounts to being able to perform a local response time analysis. We provide such an instantiation for Fixed Priority Preemptive (FPP) schedulers as in the original paper. Future work includes making the state of the art TWCA suitable for formal proofs, exploring more complex systems (*e.g.*, bounded buffers) and providing instantiations of our results for other scheduling policies.

6.3. Language Based Fault-Tolerance

Participants: Pascal Fradet, Alain Girault, Gregor Goessler, Jean-Bernard Stefani, Martin Vassor.

6.3.1. Fault Ascription in Concurrent Systems

The failure of one component may entail a cascade of failures in other components; several components may also fail independently. In such cases, elucidating the exact scenario that led to the failure is a complex and tedious task that requires significant expertise.

The notion of causality *(did an event e cause an event e'?)* has been studied in many disciplines, including philosophy, logic, statistics, and law. The definitions of causality studied in these disciplines usually amount to variants of the counterfactual test "e is a cause of e' if both e and e' have occurred, and in a world that is as close as possible to the actual world but where e does not occur, e' does not occur either". In computer science, almost all definitions of logical causality — including the landmark definition of [54] and its derivatives — rely on a causal model that. However, this model may not be known, for instance in presence of black-box components. For such systems, we have been developing a framework for blaming that helps us establish the causal relationship between component failures and system failures, given an observed system execution trace. The analysis is based on a formalization of counterfactual reasoning [6].

In [16] we have discussed several shortcomings of existing approaches to counterfactual causality from the computer science perspective, and sketched lines of work to try and overcome these issues. In particular, research on counterfactual causality analysis has been marked, since its early days, by a succession of definitions of causality that are informally (in)validated against human intuition on mostly simple examples, see *e.g.*, [54], [53]. We call this approach TEGAR, *textbook example guided analysis refinement*. As pointed out in [48], it suffers from its dependence on the tiny number and incompleteness of examples in the literature, and from the lack of stability of the intuitive judgments against which the definitions are validated. We have argued that we need a formalization of counterfactual causality based on *first principles*, in the sense that causality definitions should not be driven by individual examples but constructed from a set of precisely specified requirements. Example of such requirements are robustness of causation under equivalence of models, and well-defined behavior under abstraction and refinement. To the best of our knowledge, none of the existing causality analysis techniques provides sufficient guarantees in this regard.

We are currently working on a revised version of our general semantic framework for fault ascription in [50] that satisfies a set of formally stated requirements, and on its instantiation to acyclic models of computation, in order to compare our approach with the standard definition of *actual causality* proposed by Halpern and Pearl.

6.3.2. Tradeoff exploration between energy consumption and execution time

We have continued our work on multi-criteria scheduling, in two directions. First, in the context of dynamic applications that are launched and terminated on an embedded homogeneous multi-core chip, under execution time and energy consumption constraints, we have proposed a two layer adaptive scheduling method [14]. In the first layer, each application (represented as a DAG of tasks) is scheduled statically on subsets of cores: 2 cores, 3 cores, 4 cores, and so on. For each size of these sets (2, 3, 4, ...), there may be only one topology or several topologies. For instance, for 2 or 3 cores there is only one topology (a "line"), while for 4 cores there are three distinct topologies ("line", "square", and "T shape"). Moreover, for each topology, we generate statically several schedules, each one subject to a different total energy consumption constraint, and consequently with a different Worst-Case Reaction Time (WCRT). Coping with the energy consumption constraints is achieved thanks to Dynamic Frequency and Voltage Scaling (DVFS). In the second layer, we use these pre-generated static schedules to reconfigure dynamically the applications running on the multi-core each time a new application is launched or an existing one is stopped. The goal of the second layer is to perform a dynamic global optimization of the configuration, such that each running application meets a pre-defined quality-of-service constraint (translated into an upper bound on its WCRT) and such that the total energy consumption be minimized. For this, we *(i)* allocate a sufficient number of cores to each active application, *(ii)* allocate the unassigned cores to the applications yielding the largest gain in energy, and *(iii)* choose for each application the best topology for its subset of cores (*i.e.*, better than the by default “line” topology). This is a joint work with Ismail Assayad (U. Casablanca, Morocco) who visits the team regularly.

Second, we have proposed the first of its kind multi-criteria scheduling heuristics for a DAG of tasks onto an homogeneous multi-core chip, optimizing the execution time, the reliability, the power consumption, and the temperature. Specifically, we have worked on the static scheduling minimizing the execution time of the application under the multiple constraints that the reliability, the power consumption, and the temperature remain below some given thresholds. There are multiple difficulties: *(i)* the reliability is not an invariant measure w.r.t. time, which makes it impossible to use backtrack-free scheduling algorithms such as list scheduling [28]; to overcome this, we adopt instead the Global System Failure Rate (GSFR) as a measure of the system’s reliability, which is invariant with time [46]; *(ii)* keeping the power consumption under a given threshold requires to lower the voltage and frequency, but this has a negative impact both on the execution time and on the GSFR; keeping the GSFR below a given threshold requires to replicate the tasks on multiple cores, but this has a negative impact both on the execution time, on the power consumption, and on the temperature; *(iii)* keeping the temperature below a given threshold is even more difficult because the temperature continues to increase even after the activity stops, so each scheduling decision must be assessed not based on the current state of the chip (*i.e.*, the temperature of each core) but on the state of the chip at the end of the candidate task, and cooling slacks must be inserted. We have proposed a multi-criteria scheduling heuristics to address these challenges. It produces a static schedule of the given application graph and the given architecture description, such that the GSFR, power, and temperature thresholds are satisfied, and such that the execution time is minimized. We then combine our heuristic with a variant of the ε-constraint method [52] in order to produce, for a given application graph and a given architecture description, its entire Pareto front in the 4D space (exec. time, GSFR, power, temp.). This is a joint work with Athena Abdi and Hamid Zarandi from Amirkabir U., Iran, who have visited the team in 2016.

6.3.3. Concurrent flexible reversibility

Reversible concurrent models of computation provide natively what appears to be very fine-grained checkpoint and recovery capabilities. We have made this intuition clear by formally comparing a distributed algorithm for checkpointing and recovery based on causal information, and the distributed backtracking algorithm that lies at the heart of our reversible higher-order pi-calculus. We have shown that (a variant of) the reversible higher-order calculus with explicit rollback can faithfully encode a distributed causal checkpoint and recovery algorithm. The reverse is also true but under precise conditions, which restrict the ability to rollback a computation to an identified checkpoint. This work has currently not been published.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

- INRIA and Orange Labs have established in 2015 a joint virtual research laboratory, called I/O LAB. We have been heavily involved in the creation of the laboratory and are actively involved in its operation (Jean-Bernard Stefani is one of the two co-directors of the lab). I/O LAB focuses on the network virtualization and cloudification. As part of the work of I/O LAB, we have cooperated with Orange Lab, as part of a cooperative research contract funded by Orange, on defining architectural principles and frameworks for network cloud infrastructures encompassing control and management of computing, storage and network resources.

- With Daimler (subcontracting via iUTBS): We have bridged the gap between LET as it was originally proposed [59] and its current use in the automotive industry.

7.2. Bilateral Grants with Industry

With Thales: Early Performance assessment for evolving and variable Cyber-Physical Systems. This CIFRE grant funds the PhD of Christophe Prévot.

With Orange: Programming IoT and sofware defined radio with dynamic dataflow models of computation. This CIFRE grant funds the PhD of Arash Shafiei.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. CASERM (PERSYVAL-Lab project)

Participants: Pascal Fradet, Alain Girault, Gregor Goessler, Xiaojie Guo, Xavier Nicollin, Stephan Plassart, Sophie Quinton, Jean-Bernard Stefani.

Despite recent advances, there exists currently no integrated formal methods and tools for the design and analysis of reconfigurable multi-view embedded systems. This is the goal of the CASERM project.

The CASERM project represents a significant effort towards a COQ-based design method for reconfigurable multi-view embedded systems, in order to formalize the structure and behavior of systems and to prove their main properties. The use of a proof assistant to support such a framework is motivated by the fact that the targeted systems are both extremely complex and critical. The challenges addressed are threefold:

1. to model software architectures for embedded systems taking into account their dynamicity and multiple constraints (functional as well as non functional);
2. to propose novel scheduling techniques for dynamically reconfiguring embedded systems; and
3. to advance the state of the art in automated proving for such systems.

The objectives of CASERM that address these challenges are organized in three tasks. They consist respectively in designing an architecture description framework based on a process calculus, in proposing online optimization methods for dynamic reconfiguration systems (this is the topic of Stephan Plassart's PhD), and in developing a formal framework for real-time analysis in the COQ proof assistant (this is the topic of Xiaojie Guo's and Maxime Lesourd's PhD). A fourth task focuses on common case studies for the evaluation of the obtained results.

The CASERM consortium gathers researchers from the G-SCOP, LIG and VERIMAG laboratories who are reknown specialists in these fields. The project started in November 2016 and will last three years.

8.2. National Initiatives

8.2.1. ANR

An ANR-PRCI project called RT-PROOFS will start in 2018, which involves the SPADES project-team, MPI-SWS, TU Braunschweig, and Onera.

8.3. European Initiatives

8.3.1. Collaborations with Major European Organizations

We have a strong collaboration with the Technische Universität Braunschweig in Germany. In particular, Sophie Quinton is involved in the CCC project (http://ccc-project.org/) to provide methods and mechanisms for the verification of software updates after deployment in safety-critical systems, and in the TypicalCPA project which aims at computing deadline miss models for distributed systems.

We also have a recent collaboration with the MPI-SWS in Kaiserslautern (Germany) on formal proofs for real-time systems. This collaboration will be concretized by an ANR-PRCI project called RT-PROOFS starting in 2018, which involves MPI-SWS, TU Braunschweig, INRIA, and Onera.

8.4. International Initiatives

8.4.1. Inria Associate Teams Not Involved in an Inria International Labs

8.4.1.1. Causalysis

Title: Causality Analysis for Safety-Critical Embedded Systems

International Partner (Institution - Laboratory - Researcher):

University of Pennsylvania (United States) - PRECISE center - Oleg Sokolsky

Start year: 2015

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

Today's embedded systems become more and more complex, while an increasing number of safety-critical functions rely on them. Determining the cause(s) of a system-level failure and elucidating the exact scenario that led to the failure is today a complex and tedious task that requires significant expertise. The CAUSALYSIS project will develop automated approaches to causality analysis on execution logs.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events Organisation

9.1.1.1. General Chair, Scientific Chair

- Alain Girault is member of the steering committee of the International Federated Conference on Distributed Computing Techniques (DISCOTEC) and of the ACM International Conference on Embedded Software (EMSOFT).
- Gregor Gössler is member of the steering committee of the International Workshop on Causal Reasoning for Embedded and Safety-critical Systems Technologies (CREST).

9.1.1.2. Member of the Organizing Committees

- Sophie Quinton was part of the organization committee of the 25th International Conference on Real-Time Networks and Systems (RTNS'17).
- Sophie Quinton was co-chair of the 2nd Tutorial on Tools for Real-Time Systems (TuToR'17), held as a satellite event of RTSS'17. http://tutor2017.inria.fr/
- Sophie Quinton was co-organizer of a tutorial entitled "Multicore Architectures in the Automotive Industry: Existing Solutions, Current Problems and Future Challenges" at ESWeek'17. http://2017.rtss.org/industrial-panel/
- Sophie Quinton was the organizer of a industry panel entitled "Beyond the Deadline: New Interfaces Between Control and Scheduling for the Design and Analysis of Critical Embedded Systems" at ESWeek'17. https://team.inria.fr/spades/beyond-the-deadline/

9.1.2. Scientific Events Selection

9.1.2.1. Chair of Conference Program Committees

- Alain Girault was co-chair of the track "Model-based Design and Verification for Embedded Systems" of the Design Automation and Test in Europe Conference (DATE'17, track E3).

- Sophie Quinton was co-chair of the 8th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS'17), held as a satellite event of ECRTS'17. http://waters2017.inria.fr

9.1.2.2. Member of the Conference Program Committees

- Alain Girault served in the program committees of the Symposium on Industrial Embedded Systems (SIES'17) and the Forum on specification and Design Languages (FDL'17).
- Gregor Gössler served in the program committees of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE'17) and the 2nd international Workshop on Causal Reasoning for Embedded and Safety-critical Systems Technologies (CREST'17).
- Sophie Quinton served in the program committees of the 29th Euromicro Conference on Real-Time Systems (ECRTS'17) and the ACM SIGBED International Conference on Embedded Software (EMSOFT'17).

9.1.3. Journal

9.1.3.1. Member of the Editorial Boards

- Alain Girault is a member of the editorial board of the Journal on Embedded Systems.

9.1.3.2. Reviewer – Reviewing Activities

- Alain Girault reviewed articles for IEEE Embedded Systems Letters (ESL), Microprocessors and Microsystems, IEEE Trans. on Industrial Informatics (TII), and ACM Trans. on Embedded Computing Systems (TECS).
- Gregor Gössler reviewed articles for Formal Methods in System Design (FMSD) and Engineering Applications of Artificial Intelligence (EAAI).

9.1.4. Research Administration

- Pascal Fradet is head of the committee for doctoral studies ("Responsable du comité des études doctorales") of the INRIA Grenoble – Rhône-Alpes research center and local correspondent for the young researchers INRIA mission ("Mission jeunes chercheurs").
- Alain Girault is vice-chair of the Inria Evaluation Committee.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence : Pascal Fradet, Théorie des Langages 1 & 2, 36 HeqTD, niveau L3, Grenoble INP (Ensimag), France

Licence : Pascal Fradet, Modèles de Calcul : λ-calcul, 12 HeqTD, niveau L3, Univ. Grenoble Alpes, France

Master : Pascal Fradet, Langages et Traducteurs, 16 HeqTD, niveau M1, Polytech Grenoble, Univ. Grenoble Alpes, France

Licence : Gregor Gössler, Théorie des Langages 2, 36 HeqTD, niveau L3, Grenoble INP (Ensimag), France

Master : Xavier Nicollin, Sémantique et Analyse des Programmes, 45 HeqTD, niveau M1, Grenoble INP (Ensimag), France

Licence : Xavier Nicollin, Théorie des Langages 2, 36 HeqTD, niveau L3, Grenoble INP (Ensimag), France

Licence : Xavier Nicollin, Bases de la Programmation Impérative, 81 HeqTD (2016-2017), niveau L3, Grenoble INP (Ensimag), France

Licence : Sophie Quinton, Théorie des Langages 2, 18 HeqTD, niveau L3, Grenoble INP (Ensimag), France

Master : Sophie Quinton, Performance and Quantitative Properties, 6h, MOSIG, Univ. Grenoble Alpes, France

9.2.2. Supervision

- PhD in progress: Sihem Cherrared, “Fault Management in Multi-Tenant Programmable Networks”, Univ. Rennes 1, since October 2016, co-advised by Eric Fabre and Gregor Gössler.
- PhD in progress: Christophe Prévot, “Early Performance assessment for evolving and variable Cyber-Physical Systems”, Univ. Grenoble Alpes, since November 2015, co-advised by Alain Girault and Sophie Quinton.
- PhD in progress: Stephan Plassart, “On-line optimization in dynamic real-time systems”, Univ. Grenoble Alpes, since September 2016, co-advised by Bruno Gaujal and Alain Girault.
- PhD in progress: Xiaojie Guo, “Formal Proofs for the Analysis of Real-Time Systems in COQ”, Univ. Grenoble Alpes, since December 2016, co-advised by Pascal Fradet, Jean-François Monin, and Sophie Quinton.
- PhD in progress: Maxime Lesourd, “Generic Proofs for the Analysis of Real-Time Systems in COQ”, Univ. Grenoble Alpes, since September 2017, co-advised by Pascal Fradet, Jean-François Monin, and Sophie Quinton.
- PhD in progress: Arash Shafiei, “Programming IoT and sofware defined radio with dynamic dataflow models of computation”, Univ. Grenoble Alpes, since September 2017, co-advised by Pascal Fradet, Alain Girault, and Xavier Nicollin.
- PhD in progress: Martin Vassor, “Analysis and types for safe dynamic software reconfigurations”, Univ. Grenoble Alpes, since November 2017, co-advised by Pascal Fradet and Jean-Bernard Stefani.
- M2 MOSIG: Leila Jamshidian Sales, “Towards a Dataflow Model of Computation (MoC) for Internet of Things (IoT)”, Univ. Grenoble Alpes and Grenoble INP, September 2017, co-supervised by Pascal Fradet, Alain Girault, and Xavier Nicollin.
- M2 MOSIG: Ebrahim Naeimimoshirian, “Hierarchical actor model with encapsulation”, Univ. Grenoble Alpes and Grenoble INP, September 2017, co-supervised by Xavier Nicollin and Jean-Bernard Stefani.

9.2.3. Juries

- Alain Girault was examiner for the PhD jury of Maalej Maroua (ENS-Lyon) and president for the PhD jury of Aurélien Cavelan (ENS-Lyon).
- Sophie Quinton was member of the PhD jury of Antoine Blin (U. Pierre et Marie Curie à Paris).

10. Bibliography

Major publications by the team in recent years

[1] S. ANDALAM, P. ROOP, A. GIRAULT, C. TRAULSEN. *A Predictable Framework for Safety-Critical Embedded Systems*, in "IEEE Trans. on Computers", July 2014, vol. 63, n^{o} 7, pp. 1600–1612

[2] S. DJOKO DJOKO, R. DOUENCE, P. FRADET. *Aspects preserving properties*, in "Science of Computer Programming", 2012, vol. 77, n^{o} 3, pp. 393-422

[3] G. FREHSE, A. HAMANN, S. QUINTON, M. WÖHRLE. *Formal Analysis of Timing Effects on Closed-loop Properties of Control Software*, in "35th IEEE Real-Time Systems Symposium 2014 (RTSS)", Rome, Italy, December 2014, https://hal.inria.fr/hal-01097622

[4] A. GIRARD, G. GÖSSLER, S. MOUELHI. *Safety Controller Synthesis for Incrementally Stable Switched Systems Using Multiscale Symbolic Models*, in "IEEE Transactions on Automatic Control", 2016, vol. 61, n^{o} 6, pp. 1537-1549 [*DOI :* 10.1109/TAC.2015.2478131], https://hal.archives-ouvertes.fr/hal-01197426

[5] A. GIRAULT, H. KALLA. *A Novel Bicriteria Scheduling Heuristics Providing a Guaranteed Global System Failure Rate*, in "IEEE Trans. Dependable Secure Comput.", December 2009, vol. 6, n^{o} 4, pp. 241–254, Research report Inria 6319, http://hal.inria.fr/inria-00177117

[6] G. GÖSSLER, D. LE MÉTAYER. *A general framework for blaming in component-based systems*, in "Science of Computer Programming", 2015, vol. 113, Part 3 [*DOI :* 10.1016/J.SCICO.2015.06.010], https://hal.inria.fr/hal-01211484

[7] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Characterizing Contextual Equivalence in Calculi with Passivation*, in "Inf. Comput.", 2011, vol. 209, n^{o} 11, pp. 1390–1433

[8] S. QUINTON, M. HANKE, R. ERNST. *Formal analysis of sporadic overload in real-time systems*, in "2012 Design, Automation & Test in Europe Conference & Exhibition, DATE 2012, Dresden, Germany, March, 2012", 2012, pp. 515–520, http://dx.doi.org/10.1109/DATE.2012.6176523

Publications of the year

Articles in International Peer-Reviewed Journals

[9] L. AHRENDTS, S. QUINTON, R. ERNST. *Exploiting Execution Dynamics in Timing Analysis Using Job Sequences*, in "IEEE Design & Test of Computers", August 2017 [*DOI :* 10.1109/MDAT.2017.2746638], https://hal.inria.fr/hal-01674751

[10] A. BOUAKAZ, P. FRADET, A. GIRAULT. *A Survey of Parametric Dataflow Models of Computation*, in "ACM Transactions on Design Automation of Electronic Systems (TODAES)", January 2017, https://hal.inria.fr/hal-01417126

[11] A. BOUAKAZ, P. FRADET, A. GIRAULT. *Symbolic Analyses of Dataflow Graphs*, in "ACM Transactions on Design Automation of Electronic Systems (TODAES)", January 2017, https://hal.inria.fr/hal-01417146

Invited Conferences

[12] P. FRADET, A. GIRAULT, L. JAMSHIDIAN, X. NICOLLIN, A. SHAFIEI. *Lossy channels in a dataflow model of computation*, in "Principles of Modeling, Festschrift in Honor of Edward A. Lee", Berkeley, United States, Lecture Notes in Computer Science, Springer, October 2017, https://hal.inria.fr/hal-01666568

[13] L. SHAN, S. GRAF, S. QUINTON, L. FEJOZ. *A Framework for Evaluating Schedulability Analysis Tools*, in "Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday", Aalborg, Denmark, August 2017, https://hal.inria.fr/hal-01674731

International Conferences with Proceedings

[14] I. ASSAYAD, A. GIRAULT. *Adaptive Mapping for Multiple Applications on Parallel Architectures*, in "Third International Symposium on Ubiquitous Networking, UNET'17", Casablanca, Morocco, May 2017, https://hal.inria.fr/hal-01672463

[15] X. Guo, S. Quinton, P. Fradet, J.-F. Monin. *Work In Progress: Toward a Coq-certified Tool for the Schedulability Analysis of Tasks with Offsets*, in "RTSS 2017 - IEEE Real-Time Systems Symposium", Paris, France, IEEE, December 2017, pp. 1-3, https://hal.inria.fr/hal-01629288

[16] G. Gössler, O. Sokolsky, J.-B. Stefani. *Counterfactual Causality from First Principles?*, in "2nd International Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies (CREST 2017)", Uppsala, Sweden, 2017, vol. 259, pp. 47 - 53, https://arxiv.org/abs/1710.03393 [*DOI :* 10.4204/EPTCS.259.5], https://hal.inria.fr/hal-01631415

[17] Z. A. H. Hammadeh, R. Ernst, S. Quinton, R. Henia, L. Rioux. *Bounding Deadline Misses in Weakly-Hard Real-Time Systems with Task Dependencies*, in "Design, Automation & Test in Europe Conference & Exhibition (DATE 2017)", Lausanne, Switzerland, March 2017, https://hal.inria.fr/hal-01426632

[18] Z. A. H. Hammadeh, S. Quinton, M. Panunzio, R. Henia, L. Rioux, R. Ernst. *Budgeting Under-Specified Tasks for Weakly-Hard Real-Time Systems*, in "29th Euromicro Conference on Real-Time Systems (ECRTS) 2017", Dubrovnik, Croatia, June 2017 [*DOI :* 10.4230/LIPIcs.ECRTS.2017.17], https://hal.inria.fr/hal-01674742

[19] S. Quinton, L. Ahrendts, R. Ernst. *Finite Ready Queues As a Mean for Overload Reduction in Weakly-Hard Real-Time Systems*, in "Proceedings of the 25th International Conference on Real-Time Networks and Systems (RTNS) 2017", Grenoble, France, October 2017 [*DOI :* 10.1145/3139258.3139259], https://hal.inria.fr/hal-01674737

[20] R. von Hanxleden, T. Bourke, A. Girault. *Real-Time Ticks for Synchronous Programming*, in "FDL 2017 - 12th Forum on Specification and Design Languages", Vérone, Italy, Electronic Chips & System Design Initiative (ECSI), September 2017, https://hal.inria.fr/hal-01575629

Research Reports

[21] B. Gaujal, A. Girault, S. Plassart. *Dynamic Speed Scaling Minimizing Expected Energy Consumption for Real-Time Tasks*, UGA - Université Grenoble Alpes ; Inria Grenoble Rhône-Alpes ; Université de Grenoble, October 2017, n^o RR-9101, pp. 1-35, https://hal.inria.fr/hal-01615835

Other Publications

[22] R. Henia, L. Rioux, N. Sordon, Z. A. H. Hammadeh, R. Ernst, S. Quinton. *Demo Abstract: Bounding Deadline Misses for Weakly-Hard Real-Time Systems Designed in CAPELLA*, April 2017, 2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) 2017, Poster, https://hal.inria.fr/hal-01674754

References in notes

[23] *Automotive Open System Architecture*, 2003, http://www.autosar.org

[24] G. Leavens, M. Sitaraman (editors). *Foundations of Component-Based Systems*, Cambridge University Press, 2000

[25] Z. Liu, H. Jifeng (editors). *Mathematical Frameworks for Component Software - Models for Analysis and Synthesis*, World Scientific, 2006

[26] *A Library for formally proven schedulability analysis*, http://prosa.mpi-sws.org/

[27] ARTEMIS JOINT UNDERTAKING. *ARTEMIS Strategic Research Agenda*, 2011

[28] I. ASSAYAD, A. GIRAULT, H. KALLA. *Tradeoff Exploration between Reliability, Power Consumption, and Execution Time for Embedded Systems*, in "Int. J. Software Tools for Technology Transfer", June 2013, vol. 15, n^{o} 3, pp. 229–245

[29] E. BAINOMUGISHA, A. CARRETON, T. VAN CUTSEM, S. MOSTINCKX, W. DE MEUTER. *A Survey on Reactive Programming*, in "ACM Computing Surveys", 2013, vol. 45, n^{o} 4

[30] N. BANSAL, T. KIMBREL, K. PRUHS. *Speed Scaling to Manage Energy and Temperature*, in "Journal of the ACM", 2007, vol. 54, n^{o} 1

[31] A. BASU, S. BENSALEM, M. BOZGA, J. COMBAZ, M. JABER, T.-H. NGUYEN, J. SIFAKIS. *Rigorous Component-Based System Design Using the BIP Framework*, in "IEEE Software", 2011, vol. 28, n^{o} 3

[32] V. BEBELIS, P. FRADET, A. GIRAULT, B. LAVIGUEUR. *BPDF: A Statically Analyzable Dataflow Model with Integer and Boolean Parameters*, in "International Conference on Embedded Software, EMSOFT'13", Montreal, Canada, ACM, September 2013

[33] A. BENVENISTE, P. CASPI, S. A. EDWARDS, N. HALBWACHS, P. LE GUERNIC, R. DE SIMONE. *The synchronous languages 12 years later*, in "Proceedings of the IEEE", 2003, vol. 91, n^{o} 1

[34] A. BENVENISTE, J. RACLET, B. CAILLAUD, D. NICKOVIC, R. PASSERONE, A. SANGIOVANNI-VICENTELLI, T. HENZINGER, K. LARSEN. *Contracts for the Design of Embedded Systems Part I: Methodology and Use Cases*, in "Proceedings of the IEEE", 2012

[35] A. BENVENISTE, J. RACLET, B. CAILLAUD, D. NICKOVIC, R. PASSERONE, A. SANGIOVANNI-VICENTELLI, T. HENZINGER, K. LARSEN. *Contracts for the Design of Embedded Systems Part II: Theory*, in "Proceedings of the IEEE", 2012

[36] B. BONAKDARPOUR, S. S. KULKARNI, F. ABUJARAD. *Symbolic synthesis of masking fault-tolerant distributed programs*, in "Distributed Computing", 2012, vol. 25, n^{o} 1

[37] S. BORKAR. *Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation*, in "IEEE Micro", 2005, vol. 25, n^{o} 6

[38] E. BRUNETON, T. COUPAYE, M. LECLERCQ, V. QUEMA, J.-B. STEFANI. *The Fractal Component Model and its Support in Java*, in "Software - Practice and Experience", 2006, vol. 36, n^{o} 11-12

[39] R. BRUNI, H. C. MELGRATTI, U. MONTANARI. *Theoretical foundations for compensations in flow composition languages*, in "32nd ACM Symposium on Principles of Programming Languages (POPL)", ACM, 2005

[40] T. CHOTHIA, D. DUGGAN. *Abstractions for fault-tolerant global computing*, in "Theor. Comput. Sci.", 2004, vol. 322, n^{o} 3

[41] R. DAVIS, A. BURNS. *A Survey of Hard Real-Time Scheduling for Multiprocessor Systems*, in "ACM Computing Surveys", 2011, vol. 43, n^o 4

[42] V. DE FLORIO, C. BLONDIA. *A Survey of Linguistic Structures for Application-Level Fault-Tolerance*, in "ACM Computing Surveys", 2008, vol. 40, n^o 2

[43] J. EKER, J. W. JANNECK, E. A. LEE, J. LIU, X. LIU, J. LUDVIG, S. NEUENDORFFER, S. SACHS, Y. XIONG. *Taming heterogeneity - the Ptolemy approach*, in "Proceedings of the IEEE", 2003, vol. 91, n^o 1

[44] J. FIELD, C. A. VARELA. *Transactors: a programming model for maintaining globally consistent distributed state in unreliable environments*, in "32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)", ACM, 2005

[45] A. GIRARD, G. PAPPAS. *Approximation metrics for discrete and continuous systems*, in "IEEE Trans. on Automatic Control", 2007, vol. 52, n^o 5, pp. 782–798

[46] A. GIRAULT, H. KALLA. *A Novel Bicriteria Scheduling Heuristics Providing a Guaranteed Global System Failure Rate*, in "IEEE Trans. Dependable Secure Comput.", December 2009, vol. 6, n^o 4, pp. 241–254, Research report Inria 6319, http://www.computer.org/portal/web/csdl/doi/10.1109/TDSC.2008.50

[47] D. GIZOPOULOS, M. PSARAKIS, S. V. ADVE, P. RAMACHANDRAN, S. K. S. HARI, D. SORIN, A. MEIXNER, A. BISWAS, X. VERA. *Architectures for Online Error Detection and Recovery in Multicore Processors*, in "Design Automation and Test in Europe (DATE)", 2011

[48] C. GLYMOUR, D. DANKS, B. GLYMOUR, F. EBERHARDT, J. RAMSEY, R. SCHEINES, P. SPIRTES, C. M. TENG, J. ZHANG. *Actual causation: a stone soup essay*, in "Synthese", 2010, vol. 175, n^o 2, pp. 169–192

[49] F. C. GÄRTNER. *Fundamentals of Fault-Tolerant Distributed Computing in Asynchronous Environments*, in "ACM Computing Surveys", 1999, vol. 31, n^o 1

[50] G. GÖSSLER, J.-B. STEFANI. *Fault Ascription in Concurrent Systems*, in "Proc. Trustworthy Global Computing - 10th International Symposium, TGC 2015", P. GANTY, M. LORETI (editors), LNCS, Springer, 2016, vol. 9533

[51] S. HAAR, E. FABRE. *Diagnosis with Petri Net Unfoldings*, in "Control of Discrete-Event Systems", Lecture Notes in Control and Information Sciences, Springer, 2013, vol. 433, chap. 15

[52] Y. HAIMES, L. LASDON, D. WISMER. *On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization*, in "IEEE Trans. Systems, Man, and Cybernetics", 1971, vol. 1, pp. 296–297

[53] J. Y. HALPERN. *A Modification of the Halpern-Pearl Definition of Causality*, in "Proc. Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015", Q. YANG, M. WOOLDRIDGE (editors), AAAI Press, 2015, pp. 3022–3033, http://ijcai.org/Abstract/15/427

[54] J. HALPERN, J. PEARL. *Causes and Explanations: A Structural-Model Approach. Part I: Causes*, in "British Journal for the Philosophy of Science", 2005, vol. 56, n^o 4, pp. 843-887

[55] D. HARMANCI, V. GRAMOLI, P. FELBER. *Atomic Boxes: Coordinated Exception Handling with Transactional Memory*, in "25th European Conference on Object-Oriented Programming (ECOOP)", Lecture Notes in Computer Science, 2011, vol. 6813

[56] T. HENZINGER, J. SIFAKIS. *The Embedded Systems Design Challenge*, in "Formal Methods 2006", Lecture Notes in Computer Science, Springer, 2006, vol. 4085

[57] I. HWANG, S. KIM, Y. KIM, C. E. SEAH. *A Survey of Fault Detection, Isolation and Reconfiguration Methods*, in "IEEE Trans. on Control Systems Technology", 2010, vol. 18, n^{o} 3

[58] V. IZOSIMOV, P. POP, P. ELES, Z. PENG. *Scheduling and Optimization of Fault-Tolerant Embedded Systems with Transparency/Performance Trade-Offs*, in "ACM Trans. Embedded Comput. Syst.", 2012, vol. 11, n^{o} 3, 61 p.

[59] C. M. KIRSCH, A. SOKOLOVA. *The Logical Execution Time Paradigm*, in "Advances in Real-Time Systems (to Georg Färber on the occasion of his appointment as Professor Emeritus at TU München after leading the Lehrstuhl für Realzeit-Computersysteme for 34 illustrious years)", 2012, pp. 103–120

[60] R. KÜSTERS, T. TRUDERUNG, A. VOGT. *Accountability: definition and relationship to verifiability*, in "ACM Conference on Computer and Communications Security", 2010, pp. 526-535

[61] I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *Reversing Higher-Order Pi*, in "21th International Conference on Concurrency Theory (CONCUR)", Lecture Notes in Computer Science, Springer, 2010, vol. 6269

[62] E. A. LEE, A. L. SANGIOVANNI-VINCENTELLI. *Component-based design for the future*, in "Design, Automation and Test in Europe, DATE 2011", IEEE, 2011

[63] P. MENZIES. *Counterfactual Theories of Causation*, in "Stanford Encyclopedia of Philosophy", E. ZALTA (editor), Stanford University, 2009, http://plato.stanford.edu/entries/causation-counterfactual

[64] M. MOORE. *Causation and Responsibility*, Oxford, 1999

[65] J. PEARL. *Causal inference in statistics: An overview*, in "Statistics Surveys", 2009, vol. 3, pp. 96-146

[66] P. RAMADGE, W. WONHAM. *Supervisory Control of a Class of Discrete Event Processes*, in "SIAM Journal on control and optimization", January 1987, vol. 25, n^{o} 1, pp. 206–230

[67] G. RAMALINGAM, K. VASWANI. *Fault tolerance via idempotence*, in "40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)", ACM, 2013

[68] B. RANDELL. *System Structure for Software Fault Tolerance*, in "IEEE Trans. on Software Engineering", 1975, vol. 1, n^{o} 2

[69] J. RUSHBY. *Partitioning for Safety and Security: Requirements, Mechanisms, and Assurance*, NASA Langley Research Center, 1999, n^{o} CR-1999-209347

[70] J.-B. STEFANI. *Components as Location Graphs*, in "11th International Symposium on Formal Aspects of Component Software", Bertinoro, Italy, Lecture Notes in Computer Science, September 2014, vol. 8997, https://hal.inria.fr/hal-01094208

[71] P. TABUADA. *Verification and Control of Hybrid Systems - A Symbolic Approach*, Springer, 2009

[72] K. TINDELL. *Using offset information to analyse static priority pre-emptively scheduled task sets*, Technical report YCS 182, University of York, Department of Computer Science, 1992, https://books.google.fr/books?id=qARQHAAACAAJ

[73] D. WALKER, L. W. MACKEY, J. LIGATTI, G. A. REIS, D. I. AUGUST. *Static typing for a faulty lambda calculus*, in "11th ACM SIGPLAN International Conference on Functional Programming (ICFP)", ACM, 2006

[74] F. YAO, A. DEMERS, S. SHENKER. *A scheduling model for reduced CPU energy*, in "Proceedings of lEEE Annual Foundations of Computer Science", 1995, pp. 374–382