RESEARCH CENTRE

Rennes - Bretagne Atlantique

IN PARTNERSHIP WITH:

Université Rennes 1

2020
ACTIVITY REPORT

Project-Team
HYCOMES

Modélisation hybride & conception par contrats pour les systèmes embarqués multi-physiques

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Embedded and Real-time Systems

Contents

Project-Team HYCOMES

Creation of the Team: 2013 July 01, updated into Project-Team: 2016 September 01

Keywords

Computer sciences and digital sciences

A2. – Software

A2.1. – Programming Languages

A2.1.1. – Semantics of programming languages

A2.1.5. – Constraint programming

A2.1.9. – Synchronous languages

A2.1.10. – Domain-specific languages

A2.2. – Compilation

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. – Formal method for verification, reliability, certification

A2.4.1. – Analysis

A2.4.2. – Model-checking

A2.4.3. – Proofs

A2.5. – Software engineering

A2.5.1. – Software Architecture & Design

A2.5.2. – Component-based Design

A3. – Data and knowledge

A3.1. – Data

A3.1.1. – Modeling, representation

A6. – Modeling, simulation and control

A6.1. – Methods in mathematical modeling

A6.1.1. – Continuous Modeling (PDE, ODE)

A6.1.3. – Discrete Modeling (multi-agent, people centered)

A6.1.5. – Multiphysics modeling

A8.4. – Computer Algebra

Other research topics and application domains

B2. – Health

B2.4. – Therapies

B2.4.3. – Surgery

B4. – Energy

B4.4. – Energy delivery

B4.4.1. – Smart grids

B5. – Industry of the future

B5.2. – Design and manufacturing

B5.2.1. – Road vehicles

B5.2.2. – Railway

B5.2.3. – Aviation

B5.2.4. – Aerospace

B5.8. – Learning and training

B5.9. – Industrial maintenance

B7. – Transport and logistics

B7.1. – Traffic management

B7.1.3. – Air traffic

B8. – Smart Cities and Territories

B8.1. – Smart building/home

B8.1.1. – Energy for smart buildings

1 Team members, visitors, external collaborators

Research Scientists

- Benoît Caillaud [Team leader, Inria, Senior Researcher, HDR]
- Albert Benveniste [Inria, Emeritus, HDR]
- Khalil Ghorbal [Inria, Researcher]

PhD Students

- Christelle Kozaily [Inria]
- Aurélien Lamercerie [Univ de Rennes I]
- Joan Thibault [Univ de Rennes I]

Technical Staff

- Mathias Malandain [Inria, Engineer]
- Bertrand Provot [Inria, Engineer, from Oct 2020]

Interns and Apprentices

- Julien Duron [Univ de Rennes I, from Jun 2020 until Jul 2020]

Administrative Assistant

- Armelle Mozziconacci [CNRS]

2 Overall objectives

Hycomes was created a local team of the Rennes - Bretagne Atlantique Inria research center in 2013 and has been created as an Inria Project-Team in 2016. The team is focused on two topics in cyber-physical systems design:

- Hybrid systems modelling, with an emphasis on the design of modelling languages in which software systems, in interaction with a complex physical environment, can be modelled, simulated and verified. A special attention is paid to the mathematical rigorous semantics of these languages, and to the correctness (wrt. such semantics) of the simulations and of the static analyses that must be performed during compilation. The Modelica language is the main application field. The team aims at contributing language extensions facilitating the modelling of physical domains which are poorly supported by the Modelica language. The Hycomes team is also designing new structural analysis methods for hybrid (aka. multi-mode) Modelica models. New simulation and verification techniques for large Modelica models are also in the scope of the team.
- Contract-based design and interface theories, with applications to requirements engineering in the context of safety-critical systems design. The objective of our research is to bridge the gap between system-level requirements, often expressed in natural, constrained or semi-formal languages and formal models, that can be simulated and verified.

3 Research program

3.1 Hybrid Systems Modeling

Systems industries today make extensive use of mathematical modeling tools to design computer controlled physical systems. This class of tools addresses the modeling of physical systems with models that are simpler than usual scientific computing problems by using only Ordinary Differential Equations (ODE) and Difference Equations but not Partial Differential Equations (PDE). This family of tools first emerged in the 1980's with SystemBuild by MatrixX (now distributed by National Instruments) followed soon by Simulink by Mathworks, with an impressive subsequent development.

In the early 90's control scientists from the University of Lund (Sweden) realized that the above approach did not support component based modeling of physical systems with reuse [1]. For instance, it was not easy to draw an electrical or hydraulic circuit by assembling component models of the various devices. The development of the Omola language by Hilding Elmqvist was a first attempt to bridge this gap by supporting some form of Differential Algebraic Equations (DAE) in the models. Modelica quickly emerged from this first attempt and became in the 2000's a major international concerted effort with the Modelica Consortium [2]. A wider set of tools, both industrial and academic, now exists in this segment [3]. In the EDA sector, VHDL-AMS was developed as a standard [56] and also allows for differential algebraic equations. Several domain-specific languages and tools for mechanical systems or electronic circuits also support some restricted classes of differential algebraic equations. Spice is the historic and most striking instance of these domain-specific languages/tools [4]. The main difference is that equations are hidden and the fixed structure of the differential algebraic results from the physical domain covered by these languages.

Despite these tools are now widely used by a number of engineers, they raise a number of technical difficulties. The meaning of some programs, their mathematical semantics, is indeed ambiguous. A main source of difficulty is the correct simulation of continuous-time dynamics, interacting with discrete-time dynamics: How the propagation of mode switchings should be handled? How to avoid artifacts due to the use of a global ODE solver causing unwanted coupling between seemingly non interacting subsystems? Also, the mixed use of an equational style for the continuous dynamics with an imperative style for the mode changes and resets is a source of difficulty when handling parallel composition. It is therefore not uncommon that tools return complex warnings for programs with many different suggested hints for fixing them. Yet, these "pathological" programs can still be executed, if wanted so, giving surprising results — See for instance the Simulink examples in [27], [24] and [25].

Indeed this area suffers from the same difficulties that led to the development of the theory of synchronous languages as an effort to fix obscure compilation schemes for discrete time equation based languages in the 1980's. Our vision is that hybrid systems modeling tools deserve similar efforts in theory as synchronous languages did for the programming of embedded systems.

3.2 Background on non-standard analysis

Non-Standard analysis plays a central role in our research on hybrid systems modeling [24], [27, 26, 25]. The following text provides a brief summary of this theory and gives some hints on its usefulness in the context of hybrid systems modeling. This presentation is based on our paper [2], a chapter of Simon Bliudze's PhD thesis [33], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [62].

Non-standard numbers allowed us to reconsider the semantics of hybrid systems and propose a radical alternative to the *super-dense time semantics* developed by Edward Lee and his team as part of the Ptolemy II project, where cascades of successive instants can occur in zero time by using $\mathbb{R}_+ \times \mathbb{N}$ as a time index. In the non-standard semantics, the time index is defined as a set $\mathbb{T} = \{n\partial \mid n \in {}^*\mathbb{N}\}$, where ∂ is an *infinitesimal* and ${}^*\mathbb{N}$ is the set of *non-standard integers*. Remark that (1) $\mathbb{T}$ is dense in $\mathbb{R}_+$, making it "continuous", and (2) every $t \in \mathbb{T}$ has a predecessor in $\mathbb{T}$ and a successor in $\mathbb{T}$, making it "discrete". Although

[1] http://www.lccc.lth.se/media/LCCC2012/WorkshopSeptember/slides/Astrom.pdf

[2] https://www.modelica.org/

[3] SimScape by Mathworks, Amesim by LMS International, now Siemens PLM, and more.

[4] http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/MANUALS/spice3.html

it is not effective from a computability point of view, the *non-standard semantics* provides a framework that is familiar to the computer scientist and at the same time efficient as a symbolic abstraction. This makes it an excellent candidate for the development of provably correct compilation schemes and type systems for hybrid systems modeling languages.

Non-standard analysis was proposed by Abraham Robinson in the 1960s to allow the explicit manipulation of “infinitesimals” in analysis [71, 49, 45]. Robinson’s approach is axiomatic; he proposes adding three new axioms to the basic Zermelo-Fraenkel (ZFC) framework. There has been much debate in the mathematical community as to whether it is worth considering non-standard analysis instead of staying with the traditional one. We do not enter this debate. The important thing for us is that non-standard analysis allows the use of the non-standard discretization of continuous dynamics “as if” it was operational.

Not surprisingly, such an idea is quite ancient. Iwasaki et al. [57] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [34, 33] have also used non-standard analysis as a mathematical support for defining a system theory for hybrid systems. They discuss in detail the notion of “system” and investigate computability issues. The formalization they propose closely follows that of Turing machines, with a memory tape and a control mechanism.

3.3 Structural Analysis of DAE Systems

The Modelica language is based on Differential Algebraic Equations (DAE). The general form of a DAE is given by:

$$F(t,x,x',x'',\ldots) \tag{1}$$

where F is a system of n_e equations $\{f_1,\ldots,f_{n_e}\}$ and x is a finite list of n_v independent real-valued, smooth enough, functions $\{x_1,\ldots,x_{n_v}\}$ of the independent variable t. We use x' as a shorthand for the list of first-order time derivatives of x_j, $j=1,\ldots,n_v$. High-order derivatives are recursively defined as usual, and $x^{(k)}$ denotes the list formed by the k-th derivatives of the functions x_j. Each f_i depends on the scalar t and some of the functions x_j as well as a finite number of their derivatives.

Let $\sigma_{i,j}$ denote the highest differentiation order of variable x_j effectively appearing in equation f_i, or $-\infty$ if x_j does not appear in f_i. The *leading variables* of F are the variables in the set

$$\left\{x_j^{(\sigma_j)} \mid \sigma_j = \max_i \sigma_{i,j}\right\}$$

The *state variables* of F are the variables in the set

$$\left\{x_j^{(\nu_j)} \mid 0 \leq \nu_j < \max_i \sigma_{i,j}\right\}$$

A leading variable $x_j^{(\sigma_j)}$ is said to be *algebraic* if $\sigma_j = 0$ (in which case, neither x_j nor any of its derivatives are state variables). In the sequel, v and u denote the leading and state variables of F, respectively.

DAE are a strict generalization of *ordinary differential equations* (*ODE*), in the sense that it may not be immediate to rewrite a DAE as an explicit ODE of the form $v = G(u)$. The reason is that this transformation relies on the Implicit Function Theorem, requiring that the Jacobian matrix $\frac{\partial F}{\partial v}$ have full rank. This is, in general, not the case for a DAE. Simple examples, like the two-dimensional fixed-length pendulum in Cartesian coordinates [68], exhibit this behaviour.

For a square DAE of dimension n (i.e., we now assume $n_e = n_v = n$) to be solved in the neighborhood of some (v^*, u^*), one needs to find a set of non-negative integers $C = \{c_1,\ldots,c_n\}$ such that system

$$F^{(C)} = \{f_1^{(c_1)},\ldots,f_n^{(c_n)}\}$$

can locally be made explicit, i.e., the Jacobian matrix of $F^{(C)}$ with respect to its leading variables, evaluated at (v^*, u^*), is nonsingular. The smallest possible value of $\max_i c_i$ for a set C that satisfies this property is the *differentiation index* [39] of F, that is, the minimal number of time differentiations of all or part of the equations f_i required to get an ODE.

In practice, the problem of automatically finding a ”minimal” solution C to this problem quickly becomes intractable. Moreover, the differentiation index may depend on the value of (v^*, u^*). This is why, in lieu of numerical nonsingularity, one is interested in the *structural nonsingularity* of the Jacobian matrix, i.e., its almost certain nonsingularity when its nonzero entries vary over some neighborhood. In this framework, the *structural analysis* (*SA*) of a DAE returns, when successful, values of the c_i that are independent from a given value of (v^*, u^*).

A renowned method for the SA of DAE is the *Pantelides method*; however, Pryce's Σ-*method* is introduced also in what follows, as it is a crucial tool for our works.

3.3.1 Pantelides method

In 1988, Pantelides proposed what is probably the most well-known SA method for DAE [68]. The leading idea of his work is that the structural representation of a DAE can be condensed into a bipartite graph whose left nodes (resp. right nodes) represent the equations (resp. the variables), and in which an edge exists if and only if the variable occurs in the equation.

By detecting specific subsets of the nodes, called *Minimally Structurally Singular* (*MSS*) subsets, the Pantelides method iteratively differentiates part of the equations until a perfect matching between the equations and the leading variables is found. One can easily prove that this is a necessary and sufficient condition for the structural nonsingularity of the system.

The main reason why the Pantelides method is not used in our work is that it cannot efficiently be adapted to multimode DAE (mDAE). As a matter of fact, the adjacency graph of a mDAE has both its nodes and edges parametrized by the subset of modes in which they are active; this, in turn, requires that a parametrized Pantelides method must branch every time no mode-independent MSS is found, ultimately resulting, in the worst case, in the enumeration of modes.

3.3.2 Pryce's Sigma-method

Albeit less renowned that the Pantelides method, Pryce's Σ-method [69] is an efficient SA method for DAE, whose equivalence to the Pantelides method has been proved by the author. This method consists in solving two successive problems, denoted by primal and dual, relying on the Σ-*matrix*, or *signature matrix*, of the DAE F.

This matrix is given by:

$$\Sigma = (\sigma_{ij})_{1 \le i,j \le n} \tag{2}$$

where σ_{ij} is equal to the greatest integer k such that $x_j^{(k)}$ appears in f_i, or $-\infty$ if variable x_j does not appear in f_i. It is the adjacency matrix of a weighted bipartite graph, with structure similar to the graph considered in the Pantelides method, but whose edges are weighted by the highest differentiation orders. The $-\infty$ entries denote non-existent edges.

The *primal problem* consists in finding a *maximum-weight perfect matching* (*MWPM*) in the weighted adjacency graph. This is actually an assignment problem, for the solving of which several standard algorithms exist, such as the push-relabel algorithm [55] or the Edmonds-Karp algorithm [51] to only give a few. However, none of these algorithms are easily parametrizable, even for applications to mDAE systems with a fixed number of variables.

The *dual problem* consists in finding the component-wise minimal solution $(C, D) = (\{c_1, \dots, c_n\}, \{d_1, \dots, d_n\})$ to a given linear programming problem, defined as the dual of the aforementioned assignment problem. This is performed by means of a *fixpoint iteration* (*FPI*) that makes use of the MWPM found as a solution to the primal problem, described by the set of tuples $\{(i, j_i)\}_{i \in \{1,\dots,n\}}$:

1. Initialize $\{c_1, \dots, c_n\}$ to the zero vector.
2. For every $j \in \{1, \dots, n\}$,
$$d_j \leftarrow \max_i (\sigma_{ij} + c_i)$$
3. For every $i \in \{1, \dots, n\}$,
$$c_i \leftarrow d_{j_i} - \sigma_{i,j_i}$$
4. Repeat Steps 2 and 3 until convergence is reached.

From the results proved by Pryce in [69], it is known that the above algorithm terminates if and only if it is provided a MWPM, and that the values it returns are independent of the choice of a MWPM whenever there exist several such matchings. In particular, a direct corollary is that the Σ-method succeeds as long as a perfect matching can be found between equations and variables.

Another important result is that, if the Pantelides method succeeds for a given DAE F, then the Σ-method also succeeds for F and the values it returns for C are exactly the differentiation indices for the equations that are returned by the Pantelides method. As for the values of the d_j, being given by $d_j = \max_i(\sigma_{ij} + c_i)$, they are the differentiation indices of the leading variables in $F^{(C)}$.

Working with this method is natural for our works, since the algorithm for solving the dual problem is easily parametrizable for dealing with multimode systems, as shown in our recent paper [36].

3.3.3 Block triangular decomposition

Once structural analysis has been performed, system $F^{(C)}$ can be regarded, for the needs of numerical solving, as an algebraic system with unknowns $x_j^{(d_j)}$, $j = 1 \dots n$. As such, (inter)dependencies between its equations must be taken into account in order to put it into block triangular form (BTF). Three steps are required:

1. the *dependency graph* of system $F^{(C)}$ is generated, by taking into account the perfect matching between equations $f_i^{(c_i)}$ and unknowns $x_j^{(d_j)}$;

2. the *strongly connected components* (*SCC*) in this graph are determined: these will be the *equation blocks* that have to be solved;

3. the *block dependency graph* is constructed as the condensation of the dependency graph, from the knowledge of the SCC; a BTF of system $F^{(C)}$ can be made explicit from this graph.

3.4 Contract-Based Design, Interfaces Theories, and Requirements Engineering

System companies such as automotive and aeronautic companies are facing significant difficulties due to the exponentially raising complexity of their products coupled with increasingly tight demands on functionality, correctness, and time-to-market. The cost of being late to market or of imperfections in the products is staggering as witnessed by the recent recalls and delivery delays that many major car and airplane manufacturers had to bear in the recent years. The specific root causes of these design problems are complex and relate to a number of issues ranging from design processes and relationships with different departments of the same company and with suppliers, to incomplete requirement specification and testing.

We believe the most promising means to address the challenges in systems engineering is to employ structured and formal design methodologies that seamlessly and coherently combine the various viewpoints of the design space (behavior, space, time, energy, reliability, ...), that provide the appropriate abstractions to manage the inherent complexity, and that can provide correct-by-construction implementations. The following technology issues must be addressed when developing new approaches to the design of complex systems:

- The overall design flows for heterogeneous systems and the associated use of models across traditional boundaries are not well developed and understood. Relationships between different teams inside a same company, or between different stake-holders in the supplier chain, are not well supported by solid technical descriptions for the mutual obligations.

- System requirements capture and analysis is in large part a heuristic process, where the informal text and natural language-based techniques in use today are facing significant challenges [9]. Formal requirements engineering is in its infancy: mathematical models, formal analysis techniques and links to system implementation must be developed.

- Dealing with variability, uncertainty, and life-cycle issues, such as extensibility of a product family, are not well-addressed using available systems engineering methodologies and tools.

The challenge is to address the entire process and not to consider only local solutions of methodology, tools, and models that ease part of the design.

Contract-based design has been proposed as a new approach to the system design problem that is rigorous and effective in dealing with the problems and challenges described before, and that, at the same time, does not require a radical change in the way industrial designers carry out their task as it cuts across design flows of different types. Indeed, contracts can be used almost everywhere and at nearly all stages of system design, from early requirements capture, to embedded computing infrastructure and detailed design involving circuits and other hardware. Contracts explicitly handle pairs of properties, respectively representing the assumptions on the environment and the guarantees of the system under these assumptions. Intuitively, a contract is a pair $C = (A, G)$ of assumptions and guarantees characterizing in a formal way 1) under which context the design is assumed to operate, and 2) what its obligations are. Assume/Guarantee reasoning has been known for a long time, and has been used mostly as verification mean for the design of software [66]. However, contract based design with explicit assumptions is a philosophy that should be followed all along the design, with all kinds of models, whenever necessary. Here, specifications are not limited to profiles, types, or taxonomy of data, but also describe the functions, performances of various kinds (time and energy), and reliability. This amounts to enrich a component's interface with, on one hand, formal specifications of the behavior of the environment in which the component may be instantiated and, on the other hand, of the expected behavior of the component itself. The consideration of rich interfaces is still in its infancy. So far, academic researchers have addressed the mathematics and algorithmics of interfaces theories and contract-based reasoning. To make them a technique of choice for system engineers, we must develop:

- mathematical foundations for interfaces and requirements engineering that enable the design of frameworks and tools;
- a system engineering framework and associated methodologies and toolsets that focus on system requirements modeling, contract specification, and verification at multiple abstraction layers.

A detailed bibliography on contract and interface theories for embedded system design can be found in [3]. In a nutshell, contract and interface theories fall into two main categories:

Assume/guarantee contracts. By explicitly relying on the notions of assumptions and guarantees, A/G-contracts are intuitive, which makes them appealing for the engineer. In A/G-contracts, assumptions and guarantees are just properties regarding the behavior of a component and of its environment. The typical case is when these properties are formal languages or sets of traces, which includes the class of safety properties [59, 42, 65, 23, 44]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [50]. A/G-contracts were advocated in [28] and are is still a very active research topic, with several contributions dealing with the timed [32] and probabilistic [37, 38] viewpoints in system design, and even mixed-analog circuit design [67].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch's Input/Output Automata [64, 63]. Interface Automata [19, 18, 20, 40] focus primarily on parallel composition and compatibility: Two interfaces can be composed and are compatible if there is at least one environment where they can work together. The idea is that the resulting composition exposes as an interface the needed information to ensure that incompatible pairs of states cannot be reached. This can be achieved by using the possibility, for an Interface Automaton, to refuse selected inputs from the environment in a given state, which amounts to the implicit assumption that the environment will never produce any of the refused inputs, when the interface is in this state. Modal Interfaces [70] inherit from both Interface Automata and the originally unrelated notion of Modal Transition System [61, 22, 35, 60]. Modal Interfaces are strictly more expressive than Interface Automata by decoupling the I/O orientation of an event and its deontic modalities (mandatory, allowed or forbidden). Informally, a *must* transition is available in every component that realizes the modal interface, while a *may* transition needs not be. Research on interface theories is still very active. For instance, timed [21, 29, 31, 47, 46, 30], probabilistic [37, 48] and energy-aware [41] interface theories have been proposed recently.

Requirements Engineering is one of the major concerns in large systems industries today, particularly so in sectors where certification prevails [72]. Most requirements engineering tools offer a poor structuring of the requirements and cannot be considered as formal modeling frameworks today. They are nothing less, but nothing more than an informal structured documentation enriched with hyperlinks. As examples, medium size sub-systems may have a few thousands requirements and the Rafale fighter aircraft has above 250,000 of them. For the Boeing 787, requirements were not stable while subcontractors were working on the development of the fly-by-wire and of the landing gear subsystems, leading to a long and cahotic convergence of the design process.

We see Contract-Based Design and Interfaces Theories as innovative tools in support of Requirements Engineering. The Software Engineering community has extensively covered several aspects of Requirements Engineering, in particular:

- the development and use of large and rich *ontologies*; and
- the use of Model Driven Engineering technology for the structural aspects of requirements and resulting hyperlinks (to tests, documentation, PLM, architecture, and so on).

Behavioral models and properties, however, are not properly encompassed by the above approaches. This is the cause of a remaining gap between this phase of systems design and later phases where formal model based methods involving behavior have become prevalent—see the success of Matlab/Simulink/Scade technologies. We believe that our work on contract based design and interface theories is best suited to bridge this gap.

4 Application domains

The Hycomes team contributes to the design of mathematical modeling languages and tools, to be used for the design of cyberphysical systems. In a nutshell, two major applications can be clearly identified: (i) our work on the structural analysis of multimode DAE systems has a sizeable impact on the techniques to be used in Modelica tools; (ii) our work on the verification of dynamical systems has an impact on the design methodology for safety-critical cyberphysical systems. These two applications are detailed below.

4.1 Modelica

Mathematical modeling tools are a considerable business, with major actors such as MathWorks, with Matlab/Simulink, or Wolfram, with Mathematica. However, none of these prominent tools are suitable for the engineering of large systems. The Modelica language has been designed with this objective in mind, making the best of the advantages of DAEs to support a component-based approach. Several industries in the energy sector have adopted Modelica as their main systems engineering language.

Although multimode features have been introduced in version 3.3 of the language [52], proper tool support of multimode models is still lagging behind. The reason is not a lack of interest from tool vendors and academia, but rather that multimode DAE systems poses several fundamental difficulties, such as a proper definition of a concept of solutions for multimode DAEs, how to handle mode switchings that trigger a change of system structure, or how impulsive variables should be handled. Our work on multimode DAEs focuses on these crucial issues [6].

Thanks to the experimental coupling of Dymola (Dassault Systèmes' commercial implementation of the Modelica language) with our IsamDAE prototype (https://team.inria.fr/hycomes/software/isamdae/) [8, 17], that is being tested at the time of writing of this activity report, a larger class of Modelica models are expected to be compiled and simulated correctly. This should enable industrial users to have cleaner and simpler multimode Modelica models, with dynamically changing structure of cyberphysical systems. On the longer term, our ambition is to provide efficient code-generation techniques for the Modelica language, supporting, in full generality, multimode DAE systems, with dynamically changing differentiation index, structure and dimension.

4.2 Dynamical Systems Verification

In addition to well-defined operational semantics for hybrid systems, one often needs to provide formal guarantees about the behavior of some critical components of the system, or at least its main underlying logic. To do so, we are actively developing new techniques to automatically verify whether a hybrid system complies with its specifications, and/or to infer automatically the envelope within which the system behaves safely. The approaches we developed have been already successfully used to formally verify the intricate logic of the ACAS X, a mid-air collision avoidance system that advises the pilot to go upward or downward to avoid a nearby airplane which requires mixing the continuous motion of the aircraft with the discrete decisions to resolve the potential conflict [58]. This challenging example is nothing but an instance of the kind of systems we are targeting: autonomous smart systems that are designed to perform sophisticated tasks with an internal tricky logic. What is even more interesting perhaps is that such techniques can be often "reverted" to actually synthesize missing components so that some property holds, effectively helping the design of such complex systems.

5 Social and environmental responsibility

5.1 Impact of research results

The expected impact of our research is to allow both better designs and better exploitation of energy production units and distribution networks, enabling large-scale energy savings. At least, this is what we can observe in the context of the FUI ModeliScale collaborative project, which is focused on electric grids, urban heat networks and building thermal modeling.

The rationale is as follows: system engineering models are meant to assess the correctness, safety and optimality of a system under design. However, system models are still useful after the system has been put in operation. This is especially true in the energy sector, where systems have an extremely long lifespan (for instance, more than 50 years for some nuclear power plants) and are upgraded periodically, to integrate new technologies. Exactly like in software engineering, where a software and its model co-evolve throughout the lifespan of the software, a co-evolution of the system and its physical models has to be maintained. This is required in order to maintan the safety of the system, but also its optimality.

Moreover, physical models can be instrumental to the optimal exploitation of a system. A typical example are model-predictive control (MPC) techniques, where the model is simulated, during the exploitation of the system, in order to predict system trajectories up to a bounded-time horizon. Optimal control inputs can then be computed by mathematical programming methods, possibly using multiple simulation results. This has been proved to be a practical solution [54], whenever classical optimal control methods are ineffective, for instance, when the system is non-linear or discontinuous. However, this requires the generation of high-performance simulation code, capable of simulating a system much faster than real-time.

The structural analysis techniques implemented in IsamDAE [8] generate a conditional block dependency graph, that can be used to generate high-performance simulation code : static code can be generated for each block of equations, and a scheduling of these blocks can be computed, at runtime, at each mode switching, thanks to an inexpensive topological sort algorithm. Contrarily to other approaches (such as [53]), no structural analysis, block-trangular decompositions, or automatic differentiation has to be performed at runtime.

6 Highlights of the year

The main highlights for 2020 are the two following achievements:

1. The publication of [6], a 47-pages long journal paper, detailing a comprehensive theory of multi-mode DAE systems. A particular attention is paid to the structural analysis of (possibly impulsive) mode switchings.

2. The development of the IsamDAE software (`https://team.inria.fr/hycomes/software/isamdae/`) became in 2020 a major undertaking for the Hycomes team. This software implements

structural analysis algorithms presented in [8, 17]. The development team has been strenghtened in October 2020 with the hiring of Bertrand Provot, a software engineer in charge of the consolidation, testing and documentation of the software.

7 New software and platforms

7.1 New software

7.1.1 Demodocos

Name: Demodocos (Examples to Generic Scenario Models Generator)

Keywords: Surgical process modelling, Net synthesis, Process mining

Scientific Description: Demodocos is used to construct a Test and Flip net (Petri net variant) from a collection of instances of a given procedure. The tool takes as input either standard XES log files (a standard XML file format for process mining tools) or a specific XML file format for surgical applications. The result is a Test and Flip net and its marking graph. The tool can also build a #SEVEN scenario for integration into a virtual reality environment. The scenario obtained corresponds to the generalization of the input instances, namely the instances synthesis enriched with new behaviors respecting the relations of causality, conflicts and competition observed.

Demodocos is a synthesis tool implementing a linear algebraic polynomial time algorithm. Computations are done in the Z/2Z ring. Test and Flip nets extend Elementary Net Systems by allowing test to zero, test to one and flip arcs. The effect of flip arcs is to complement the marking of the place. While the net synthesis problem has been proved to be NP hard for Elementary Net Systems, thanks to flip arcs, the synthesis of Test and Flip nets can be done in polynomial time. Test and flip nets have the required expressivity to give concise and accurate representations of surgical processes (models of types of surgical operations). Test and Flip nets can express causality and conflict relations. The tool takes as input either standard XES log files (a standard XML file format for process mining tools) or a specific XML file format for surgical applications. The output is a Test and Flip net, solution of the following synthesis problem: Given a finite input language (log file), compute a net, which language is the least language in the class of Test and Flip net languages, containing the input language.

Functional Description: The tool Demodocos allows to build a generic model for a given procedure from some examples of instances of this procedure. The generated model can take the form of a graph, a Test 'n Flip net or a SEVEN scenario (intended for integration into a virtual reality environment).

The classic use of the tool is to apply the summary operation to a set of files describing instances of the target procedure. Several file formats are supported, including the standard XES format for log events. As output, several files are generated. These files represent the generic procedure in different forms, responding to varied uses.

This application is of limited interest in the case of an isolated use, out of context and without a specific objective when using the model generated. It was developed as part of a research project focusing in particular on surgical procedures, and requiring the generation of a generic model for integration into a virtual reality training environment. It is also quite possible to apply the same method in another context.

Publication: hal-00872284

Authors: Benoît Caillaud, Aurélien Lamercerie

Contacts: Benoît Caillaud, Aurélien Lamercerie

Participants: Aurélien Lamercerie, Benoît Caillaud

7.1.2 IsamDAE

Name: Implicit Structural Analysis of Multimode DAE systems

Keywords: Structural analysis, Differential algebraic equations, Multimode, Scheduling

Scientific Description: Modeling languages and tools based on Differential Algebraic Equations (DAE) bring several specific issues that do not exist with modeling languages based on Ordinary Differential Equations. The main problem is the determination of the differentiation index and latent equations. Prior to generating simulation code and calling solvers, the compilation of a model requires a structural analysis step, which reduces the differentiation index to a level acceptable by numerical solvers.

The Modelica language, among others, allows hybrid models with multiple modes, mode-dependent dynamics and state-dependent mode switching. These Multimode DAE (mDAE) systems are much harder to deal with. The main difficulties are (i) the combinatorial explosion of the number of modes, and (ii) the correct handling of mode switchings.

The aim of the software is on the first issue, namely: How can one perform a structural analysis of an mDAE in all possible modes, without enumerating these modes? A structural analysis algorithm for mDAE systems has been designed and implemented, based on an implicit representation of the varying structure of an mDAE. It generalizes J. Pryce's Sigma-method to the multimode case and uses Binary Decision Diagrams (BDD) to represent the mode-dependent structure of an mDAE. The algorithm determines, as a function of the mode, the set of latent equations, the leading variables and the state vector. This is then used to compute a mode-dependent block-triangular decomposition of the system, that can be used to generate simulation code with a mode-dependent scheduling of the blocks of equations.

Functional Description: IsamDAE (Implicit Structural Analysis of Multimode DAE systems) is a software library implementing new structural analysis algorithms for multimode DAE systems, based on an implicit representation of incidence graphs, matchings between equations and variables, and block decompositions. The input of the software is a variable dimension multimode DAE system consisting in a set of guarded equations and guarded variable declarations. It computes a mode-dependent structural index reduction of the multimode system and produces a mode-dependent graph for the scheduling of blocks of equations. It also computes the differentiation order of the latent equations and leading variables, as functions of the modes.

IsamDAE is coded in OCaml, and uses (at least partially) the following packages: * MLBDD by Arlen Cox, * Menhir by François Pottier and Yann Régis-Gianas, * GuaCaml and Snowflake by Joan Thibault, * Pprint by François Pottier, * XML-Light by Nicolas Cannasse and Jacques Garrigue.

Release Contributions: Versions 0.3a to 0.3d (released between Mar. and Dec. 2020):

* Performance improvements: connection with the Snowflake package by Joan Thibault, based on his PhD works on RBTF (Reduced Block-Triangular Forms). The order in which variables and equations are declared in the model, and the way these declarations are grouped, has way less impact on performances when RBTF is active (now the default behaviour of IsamDAE). * New data structures were implemented in order to correct the inputs of equations blocks in the XML, text and graph outputs. Before this fix, when two or several derivatives of the same variable appeared in the same equation (as in the simple equation ‘der(x) + x = 0‘), the lower-order derivatives of this variable were ignored. * New examples: several examples have been added, in mechanics, electrodynamics and hydraulics. * Documentation: a comprehensive User and Developer manual is made available.

News of the Year: It has been possible to perform the structural analysis of systems with more than 750 equations and 10 to the power 23 modes, therefore demonstrating the scalability of the method.

URL: https://team.inria.fr/hycomes/software/isamdae/

Publication: hal-02476541

Authors: Benoît Caillaud, Mathias Malandain, Joan Thibault

Contacts: Benoît Caillaud, Mathias Malandain, Joan Thibault

8 New results

8.1 Mathematical Foundations of Physical Systems Modeling Languages

Participants Albert Benveniste, Benoît Caillaud, Mathias Malandain.

Modern modeling languages for general physical systems, such as Modelica or Simscape, rely on Differential Algebraic Equations (DAE), i.e., constraints of the form $f(\dot{x}, x, u) = 0$, when only first-order derivatives are considered. This facilitates modeling from first principles of the physics. This year we completed and published in the Annual Reviews in Control [6] the development of the mathematical theory needed to sound, on solid mathematical bases, the design of compilers and tools for DAE based physical modeling languages.

Unlike Ordinary Differential Equations (ODE, of the form $\dot{x} = g(x, u)$), DAE exhibit subtle issues because of the notion of *differentiation index* and related *latent equations*—ODE are DAE of index zero for which no latent equation needs to be considered. Prior to generating execution code and calling solvers, the compilation of such languages requires a nontrivial *structural analysis* step that reduces the differentiation index to a level acceptable by DAE solvers.

Multimode DAE systems, having multiple modes with mode-dependent dynamics and state-dependent mode switching, are much harder to deal with. The main difficulty is the handling of the events of mode change. Unfortunately, the large literature devoted to the numerical analysis of DAEs does not cover the multimode case, typically saying nothing about mode changes. This lack of foundations causes numerous difficulties to the existing modeling tools. Some models are well handled, others are not, with no clear boundary between the two classes. Basically, no tool exists that performs a correct structural analysis taking multiple modes and mode changes into account.

In our work, we developed a comprehensive mathematical approach supporting compilation and code generation for this class of languages. Its core is the *structural analysis of multimode DAE systems,* taking both multiple modes and mode changes into account. As a byproduct of this structural analysis, we propose well sound criteria for accepting or rejecting models at compile time.

For our mathematical development, we rely on *nonstandard analysis,* which allows us to cast hybrid systems dynamics to discrete time dynamics with infinitesimal step size, thus providing a uniform framework for handling both continuous dynamics and mode change events.

8.2 An implicit structural analysis method for multimode DAE systems

Participants Albert Benveniste, Benoît Caillaud, Mathias Malandain, Joan Thibault.

Modeling languages and tools based on Differential Algebraic Equations (DAE) bring several specific issues that do not exist with modeling languages based on Ordinary Differential Equations. The main problem is the determination of the differentiation index and latent equations. Prior to generating simulation code and calling solvers, the compilation of a model requires a structural analysis step, which reduces the differentiation index to a level acceptable by numerical solvers.

The Modelica language, among others, allows hybrid models with multiple modes, mode-dependent dynamics and state-dependent mode switching. These multimode DAE (mDAE) systems are much harder to deal with. The main difficulties are (i) the combinatorial explosion of the number of modes, and (ii) the correct handling of mode switchings.

The focus of the paper [36] is on the first issue, namely: How can one perform a structural analysis of an mDAE in all possible modes, without enumerating these modes? A structural analysis algorithm for mDAE systems is presented, based on an implicit representation of the varying structure of an mDAE. It generalizes J. Pryce's Σ-method [69] to the multimode case and uses Binary Decision Diagrams (BDD) to represent the mode-dependent structure of an mDAE. The algorithm determines, as a function of the mode, the set of latent equations, the leading variables and the state vector. This is then used to compute a mode-dependent block-triangular decomposition of the system, that can be used to generate simulation code with a mode-dependent scheduling of the blocks of equations.

This method has been implemented in the IsamDAE software. This has allowed the Hycomes team to evaluate the performance and scalability of the method on several examples. In particular, it has been possible to perform the structural analysis of systems with more than 2300 equations and 10^{77} modes.

8.3 Ordered Functional Decision Diagrams: A Functional Semantics For Binary Decision Diagrams

Participants Joan Thibault, Khalil Ghorbal.

We introduce a novel framework, termed λDD, that revisits Binary Decision Diagrams from a purely functional point of view. The framework allows to classify the already existing variants, including the most recent ones like Chain-DD and ESRBDD, as implementations of a special class of ordered models. We enumerate, in a principled way, all the models of this class and isolate its most expressive model. This new model, termed λDD-O-NUCX, is suitable for both dense and sparse Boolean functions, and is moreover invariant by negation. The canonicity of λDD-O-NUCX is formally verified using the Coq proof assistant. We furthermore give bounds on the size of the different diagrams: the potential gain achieved by more expressive models can be at most linear in the number of variables n.

8.4 Functional Decision Diagrams: A Unifying Data Structure For Binary Decision Diagrams

Participants Joan Thibault, Khalil Ghorbal.

We present concise and canonical representations of Boolean functions akin to Binary Decision Diagrams, a versatile data structure with several applications beyond computer science. Our approach is functional: we encode the process that constructs the Boolean function of interest starting from the constant function zero (or False). This point of view makes the data structure more resilient to variable ordering, a well-known problem in standard representations. The experiments on both dense and sparse formulas are very encouraging and show not only a better compression rate of the final representation than all existing related variants but also a lower memory peak.

8.5 Characterizing Positively Invariant Sets: Inductive and Topological Methods

Participants Khalil Ghorbal.

Set positive invariance is an important concept in the theory of dynamical systems and one which also has practical applications in areas of computer science, such as formal verification, as well as in control theory. Great progress has been made in understanding positively invariant sets in continuous dynamical systems and powerful computational tools have been developed for reasoning about them; however, many of the insights from recent developments in this area have largely remained folklore and are not elaborated in existing literature. This article contributes an explicit development of modern methods for checking positively invariant sets of ordinary differential equations and describes two possible characterizations of positive invariants: one based on the real induction principle, and a novel alternative based on topological notions. The two characterizations, while in a certain sense equivalent, lead to two different decision procedures for checking whether a given semi-algebraic set is positively invariant under the flow of a system of polynomial ordinary differential equations.

8.6 Characterizing Q-matrices

Participants Khalil Ghorbal, Christelle Kozaily.

We show that the existence of solutions for linear complementarity problems amounts to a covering of the entire space by a set of finite cones defined by the involved vectors as well as the standard basis. We give several full characterizations for the case $n = 2$ and detail how these could be used to derive several necessary conditions for higher dimensions. The local existence of solutions is also investigated. It is shown that the positivity condition on the determinant, or equivalently, the orientation of the vectors forming the complementarity cones cannot be captured purely structurally.

9 Bilateral contracts and grants with industry

Glose (2018–2021) In the context of a framework agreement between Safran Tech. of the Safran aeronautic group and Inria, the Hycomes team, jointly with the KAIROS and DIVERSE teams, contributes to the Glose research grant funded by Safran. The contributions of the Hycomes team are structural analysis techniques for multimode DAE models resulting from the coupling of quasi-static models, expressed as non-linear equations, with dynamical systems, in the form of systems of ordinary differential equations. The multimode features of the model come from the dynamic changes of the system structures, possibly resulting from changes of mode of operation, or mechanical failure. Current work of the Hycomes team focuses on the definition of a component model, encapsulating multimode DAE systems, and on modular structural analysis methods, capable of characterizing, from a structural point of view only, the possible environments in which a component model may be correctly instantiated.

10 Partnerships and cooperations

10.1 International research visitors

The visit of Inigo Incer Romeo, PhD student at U. Berkeley, initially planned in the Summer 2020 had to be postponed to 2021. This visit is supported by a Chateaubriand Fellowship grant of the French Ministry of Foreign Affairs. The topics of the visit is on the use of Contract-based Reasoning to support the design of CPS systems.

10.2 National initiatives

10.2.1 Inria Project Lab (IPL): ModeliScale, Languages and Compilation for Cyber-Physical System Design

The project gathers researchers from three Inria teams (Hycomes, Parkas and Tripop), and from three other research labs in Paris area (ENSTA Paris-Tech, L2S-CNRS and LIX, École Polytechnique).

The main objective of ModeliScale is to advance modeling technologies (languages, compile-time analyses, simulation techniques) for CPS combining physical interactions, communication layers and software components. We believe that mastering CPS comprising thousands to millions of components requires radical changes of paradigms. For instance, modeling techniques must be revised, especially when physics is involved. Modeling languages must be enhanced to cope with larger models. This can only be done by combining new compilation techniques (to master the structural complexity of models) with new mathematical tools (new numerical methods, in particular).

ModeliScale gathers a broad scope of experts in programming language design and compilation (reactive synchronous programming), numerical solvers (nonsmooth dynamical systems) and hybrid systems modeling and analysis (guaranteed simulation, verification). The research program is carried out in close cooperation with the Modelica community as well as industrial partners, namely, Dassault Systèmes as a Modelica/FMI tool vendor, and EDF and Engie as end users.

In 2020, two general meetings have been organized by visioconference, with presentations of the partners on new results related to hybrid systems modeling and verification.

Two PhDs are funded by the ModeliScale IPL. Both started in October 2018:

- Christelle Kozaily has started a PhD, under the supervision of Vincent Acary (TRIPOP team at Inria Grenoble), Benoît Caillaud, Khalil Ghorbal on the structural and numerical analysis of non-smooth DAE systems. She is located in the Hycomes team at Inria Rennes.
- Ismail Lahkim-Bennani has started a PhD under the supervision of Goran Frehse (ENSTA ParisTech.) and Marc Pouzet (PARKAS team, INRIA/ENS Paris). His PhD topic is on random testing of hybrid systems, using techniques inspired by QuickCheck [43].

10.2.2 FUI ModeliScale: Scalable Modeling and Simulation of Large Cyber-Physical Systems

Participants Albert Benveniste, Benoît Caillaud, Mathias Malandain, Bertrand Provot.

FUI ModeliScale is a French national collaborative project coordinated by Dassault Systèmes. The partners of this project are: EDF and Engie as main industrial users; DPS, Eurobios and PhiMeca are SME providing mathematical modeling expertise; CEA INES (Chambéry) and Inria are the academic partners. The project started January 2018, for a maximal duration of 42 months. Three Inria teams are contributing to the project : Hycomes, Parkas (Inria Paris / ENS) and Tripop (Inria Grenoble / LJK).

The focus of the project is on the scalable analysis, compilation and simulation of large Modelica models. The main contributions expected from Inria are:

- A novel structural analysis algorithm for multimode DAE systems, capable of handling large systems of guarded equations, that do not depend on the enumeration of a possibly exponential number of modes.
- The partitioning and high-performance distributed co-simulation of large Modelica models, based on the results of the structural analysis.

In 2020, the effort has been put on the first objective, and in particular the improvement of the scalability of the algorithms implemented in the IsamDAE software (https://team.inria.fr/hycomes/software/isamdae/). The performance of the tool has been improved by two orders of magnitude on some examples. This has allowed us to perform the structural analysis of multimode models of up to 2300 equations en 10^{77} modes.

A coupling of IsamDAE with Dymola (Dassault Système's commercial implementation of the Modelica language) has been implemented by Dassault Systèmes AB (Lund, Sweden), and is currently under test at the time of writing of this activity report.

11 Dissemination

11.1 Promoting scientific activities

11.1.1 Scientific events: organisation

General chair, scientific chair Khalil Ghorbal was the co-Chair of the NSAD Workshop (satellite of the SPLASH 2020 Event). https://2020.splashcon.org/home/nsad-2020

Abstract domains are a key notion in Abstract Interpretation theory and practice. They embed the semantic choices, data-structures and algorithmic aspects, and implementation decisions. The Abstract Interpretation framework provides constructive and systematic formal methods to design, compose, compare, study, prove, and apply abstract domains. Many abstract domains have been designed so far: numerical domains (intervals, congruences, polyhedra, polynomials, etc.), symbolic domains (shape domains, trees, etc.), but also domain operators (products, powersets, completions, etc.), and have been applied to several kinds of static analyses (safety, termination, probability, etc.) on a variety of systems (hardware, software, neural networks, etc.). The goal of NSAD workshop is to discuss work in progress, recent advances, novel ideas, experiences in the theory, practice, application, implementation, and experimentation related to abstract domains and/or their combination. This year's edition in particular welcomes abstract domains related and/or applied to analyzing neural networks, dynamical and hybrid systems.

11.1.2 Scientific events: selection

Member of the conference program committees Benoît Caillaud has served on the program committee of FDL'20, a workshop on the domain-specific languages. The workshop took place with both physical (in Kiel, Germany) and virtual attendance (by visioconference).

11.1.3 Journal

Reviewer - reviewing activities

- Benoît Caillaud has reviewed papers for ACM-TECS and IEEE-TAC;
- Khalil Ghorbal served as a reviewer for IEEE-TAC, IEEE-TECS, LITES, ICALP, Automatica;
- Mathias Malandain served as a reviewer for the Asian Modelica Conference 2020;
- Albert Benveniste served as a reviewer for the journals *Transactions on Software Engineering, Discrete Event Dynamic Systems, Automatica* and the *American Control Conference.*

11.1.4 Invited talks

- Benoît Caillaud has given a talk on switched DAE systems at the students seminar of the Department of Applied Mathematics at the ENS Paris-Saclay,
- Khalil Ghorbal was invited to give a talk at the PolySys Seminar (LIP6, Symbolic Computation),
- Khalil Ghorbal was invited to give a talk at the University of Perpignan (Computer Science, Mathematics and Physics departments),
- Khalil Ghorbal was invited to give a talk at the RWTH Aachen University (Computer Science and control departments).

11.1.5 Scientific expertise

- Albert Benveniste was chair of the Scientific Council of Orange Labs. His term terminated by end of february 2020.
- Albert Benveniste is member of the Scientific Council of Safran. In the period January to March 2020, together with Nikos Paragios (also member of the council), he prepared a report on the impacts and opportunities for AI in Safran, and the way forward, both internally and with the ecosystem.
- Albert Benveniste participated to the activities of the Academy of technologies (Pôle Numérique) since march 2020. More specifically, he started a Working Group on Crisis Management methods and tools targeting the COVID pandemia. The moto is: modeling should be extended beyond epidemiology. In 2020, long interviews with demos were held with Dassault-Systèmes, Thales, IBM, and the startup Causality Link. The report should be issued early in 2021.

11.1.6 Research administration

Benoît Caillaud is head of the *Programming Languages and Software Engineering* department of IRISA (UMR 6074). Part of his duties has been the preparation of the evaluation of IRISA, planned March 2021.

11.2 Teaching - Supervision - Juries

11.2.1 Teaching

- Master : Khalil Ghorbal, Category Theory, Monads, and Computation, M2, (enseignant principal), 30h EqTD, ENS Rennes, France
- Master : Khalil Ghorbal, Modeling Physics with Differential-Algebraic Equations, M2, (enseignant principal), 25h EqTD, Ecole Polytechnique, Palaiseau, France
- Licence : Mathias Malandain taught linear algebra and integration in multivariable calculus to 1st and 2nd-year students at the University of Rennes 1 (36 hours).

11.2.2 Supervision

- PhD: Christelle Kozaily, Structural analysis of nonsmooth dynamical systems, university of Rennes 1, co-supervised by Vincent Acary (Tripop [5] team at Inria Grenoble), Benoît Caillaud and Khalil Ghorbal, started October 2018.
- PhD: Aurélien Lamercerie, Formal analysis of cyber-physical systems requirements expressed in natural language, university of Rennes 1, co-supervised by par Benoît Caillaud et Annie Forêt (SemLIS [6] team of IRISA), started December 2017. His PhD defence is planned April 2021.
- PhD: Joan Thibault, Structural Analysis Techniques for Binary Decision Diagrams, university of Rennes 1, co-supervised by Benoît Caillaud and Khalil Ghorbal.
- Internship M1: Julien Duron, on Graph Combinatorial Optimization using Structural Analysis methods for Boolean Functions, ENS Rennes, co-supervised by Joan Thibault and Khalil Ghorbal.

12 Scientific production

12.1 Major publications

[1] A. Benveniste, T. Bourke, B. Caillaud, J.-L. Colaço, C. Pasteur and M. Pouzet. 'Building a Hybrid Systems Modeler on Synchronous Languages Principles'. In: *Proceedings of the IEEE*. Design Automation for Cyber-Physical Systems 106.9 (Sept. 2018), pp. 1568–1592. DOI: 10.1109/JPROC.2018.2858016. URL: https://hal.inria.fr/hal-01879026.

[2] A. Benveniste, T. Bourke, B. Caillaud and M. Pouzet. 'Non-standard semantics of hybrid systems modelers'. English. In: *Journal of Computer and System Sciences* 78.3 (2012). This work was supported by the SYNCHRONICS large scale initiative of INRIA, pp. 877–910. DOI: 10.1016/j.jcss.2011.08.009. URL: http://hal.inria.fr/hal-00766726.

[3] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger and K. G. Larsen. 'Contracts for System Design'. In: *Foundations and Trends in Electronic Design Automation* 12.2-3 (2018), pp. 124–400. DOI: 10.1561/1000000053. URL: https://hal.inria.fr/hal-01971429.

[4] J.-B. Jeannin, K. Ghorbal, Y. Kouskoulas, A. Schmidt, R. Gardner, S. Mitsch and A. Platzer. 'A Formally Verified Hybrid System for Safe Advisories in the Next-Generation Airborne Collision Avoidance System'. In: *International Journal on Software Tools for Technology Transfer* 19.6 (Nov. 2017), pp. 717–741. DOI: 10.1007/s10009-016-0434-1. URL: https://hal.archives-ouvertes.fr/hal-01232365.

[5] A. Sogokon, K. Ghorbal and T. T. Johnson. 'Operational Models for Piecewise-Smooth Systems'. In: *ACM Transactions on Embedded Computing Systems (TECS)* 16.5s (Oct. 2017), 185:1–185:19. DOI: 10.1145/3126506. URL: https://hal.inria.fr/hal-01658196.

[5] https://team.inria.fr/tripop/

[6] https://www-semlis.irisa.fr

12.2 Publications of the year

International journals

[6] A. Benveniste, B. Caillaud and M. Malandain. ‘The mathematical foundations of physical systems modeling languages’. In: *Annual Reviews in Control* (Dec. 2020). DOI: 10.1016/j.arcontrol.2020.08.001. URL: https://hal.inria.fr/hal-03045498.

[7] F. Lécuyer, V. Gouranton, A. Lamercerie, A. Reuzeau, B. Arnaldi and B. Caillaud. ‘Unveiling the implicit knowledge, one scenario at a time’. In: *Visual Computer* (2020), pp. 1–12. DOI: 10.1007/s00371-020-01904-7. URL: https://hal.inria.fr/hal-02879083.

International peer-reviewed conferences

[8] B. Caillaud, M. Malandain and J. Thibault. ‘Implicit structural analysis of multimode DAE systems’. In: HSCC 2020 - 23rd ACM International Conference on Hybrid Systems: Computation and Control. Sydney New South Wales Australia, France, 21st Apr. 2020, pp. 1–11. DOI: 10.1145/3365365.3382201. URL: https://hal.inria.fr/hal-02572879.

[9] A. Lamercerie and B. Caillaud. ‘An Algebra of Deterministic Propositional Acceptance Automata (DPAA)’. In: FDL 2020 - Forum on specification & Design Languages. Kiel, Germany, 15th Sept. 2020, pp. 1–8. URL: https://hal.archives-ouvertes.fr/hal-02971772.

Conferences without proceedings

[10] A. Lamercerie. ‘ARES : un extracteur d’exigences pour la modélisation de systèmes’. In: EGC 2020 - Extraction et Gestion des Connaissances (Atelier - Fouille de Textes - Text Mine). Bruxelles, Belgium, 27th Jan. 2020, pp. 1–4. URL: https://hal.archives-ouvertes.fr/hal-02971727.

[11] A. Lamercerie. ‘Transduction sémantique pour la modélisation de système’. In: PFIA 2020 - Plate-Forme de l’Intelligence Artificielle (PFIA), rencontres RJCIA. Angers, France, 29th June 2020, pp. 1–6. URL: https://hal.archives-ouvertes.fr/hal-02971742.

Reports & preprints

[12] A. Benveniste, B. Caillaud and M. Malandain. *Structural Analysis of Multimode DAE Systems: summary of results*. Inria Rennes – Bretagne Atlantique, 8th Jan. 2021, p. 27. URL: https://hal.inria.fr/hal-03104030.

[13] A. Benveniste, B. Caillaud and M. Malandain. *The Mathematical Foundations of Physical Systems Modeling Languages*. Inria, Apr. 2020, p. 112. URL: https://hal.inria.fr/hal-02521747.

[14] A. Benveniste, K. G. Larsen and J.-B. Raclet. *Mixed Nondeterministic-Probabilistic Interfaces*. Inria Rennes Bretagne Atlantique; Aalborg University; Université de Toulouse 3 Paul Sabatier, 2nd Nov. 2020, p. 40. URL: https://hal.inria.fr/hal-02985273.

[15] B. Caillaud, M. Malandain and J. Thibault. *Implicit Structural Analysis of Multimode DAE Systems*. Inria Rennes - Bretagne Atlantique; IRISA, Université de Rennes, 12th Feb. 2020. URL: https://hal.inria.fr/hal-02476541.

[16] J. Thibault and K. Ghorbal. *Ordered Functional Decision Diagrams: A Functional Semantic For Binary Decision Diagrams*. Inria, 2020. URL: https://hal.inria.fr/hal-02512117.

Other scientific publications

[17] B. Caillaud, M. Malandain and J. Thibault. *Demo: IsamDAE, an Implicit Structural Analysis Tool for Multimode DAE Systems*. Sydney, Australia, 21st Apr. 2020. URL: https://hal.inria.fr/hal-02545380.

12.3 Cited publications

[18] L. de Alfaro. 'Game Models for Open Systems'. In: *Verification: Theory and Practice*. Vol. 2772. Lecture Notes in Computer Science. Springer, 2003, pp. 269–289.

[19] L. de Alfaro and T. A. Henzinger. 'Interface automata'. In: *Proc. of the 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'01)*. ACM Press, 2001, pp. 109–120.

[20] L. de Alfaro and T. A. Henzinger. 'Interface-based design'. In: *In Engineering Theories of Software Intensive Systems, proceedings of the Marktoberdorf Summer School*. Kluwer, 2004.

[21] L. de Alfaro, T. A. Henzinger and M. Stoelinga. 'Timed Interfaces'. In: *Proc. of the 2nd International Workshop on Embedded Software (EMSOFT'02)*. Vol. 2491. Lecture Notes in Computer Science. Springer, 2002, pp. 108–122.

[22] A. Antonik, M. Huth, K. G. Larsen, U. Nyman and A. Wasowski. '20 Years of Modal and Mixed Specifications'. In: *Bulletin of European Association of Theoretical Computer Science* 1.94 (2008).

[23] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, Cambridge, 2008.

[24] A. Benveniste, T. Bourke, B. Caillaud, J.-L. Colaço, C. Pasteur and M. Pouzet. 'Building a Hybrid Systems Modeler on Synchronous Languages Principles'. In: *Proceedings of the IEEE*. Design Automation for Cyber-Physical Systems 106.9 (Sept. 2018), pp. 1568–1592. DOI: 10.1109/JPROC.2018.2858016. URL: https://hal.inria.fr/hal-01879026.

[25] A. Benveniste, T. Bourke, B. Caillaud, B. Pagano and M. Pouzet. *A Type-Based Analysis of Causality Loops In Hybrid Systems Modelers*. Deliverable D3.1_1 v 1.0 of the Sys2soft collaborative project "Physics Aware Software". Dec. 2013. URL: https://hal.inria.fr/hal-00938866.

[26] A. Benveniste, T. Bourke, B. Caillaud and M. Pouzet. *Semantics of multi-mode DAE systems*. Deliverable D.4.1.1 of the ITEA2 Modrio collaborative project. Aug. 2013. URL: https://hal.inria.fr/hal-00938891.

[27] A. Benveniste, B. Caillaud, B. Pagano and M. Pouzet. 'A type-based analysis of causality loops in hybrid modelers'. In: *HSCC '14: International Conference on Hybrid Systems: Computation and Control*. Proceedings of the 17th international conference on Hybrid systems: computation and control (HSCC '14). Berlin, Germany: ACM Press, Apr. 2014, p. 13. DOI: 10.1145/2562059.2562125. URL: https://hal.inria.fr/hal-01093388.

[28] A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone and C. Sofronis. 'Multiple Viewpoint Contract-Based Specification and Design'. In: *Proceedings of the Software Technology Concertation on Formal Methods for Components and Objects (FMCO'07)*. Vol. 5382. Revised Lectures, Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Oct. 2008.

[29] N. Bertrand, A. Legay, S. Pinchinat and J.-B. Raclet. 'A Compositional Approach on Modal Specifications for Timed Systems'. In: *11th International Conference on Formal Engineering Methods (ICFEM'09)*. Vol. 5885. LNCS. Rio de Janeiro, Brazil: Springer, Dec. 2009, pp. 679–697. URL: http://hal.inria.fr/inria-00424356/en.

[30] N. Bertrand, A. Legay, S. Pinchinat and J.-B. Raclet. 'Modal event-clock specifications for timed component-based design'. In: *Science of Computer Programming* (2011). DOI: 10.1016/j.scico.2011.01.007. URL: http://dx.doi.org/10.1016/j.scico.2011.01.007.

[31] N. Bertrand, S. Pinchinat and J.-B. Raclet. 'Refinement and Consistency of Timed Modal Specifications'. In: *3rd International Conference on Language and Automata Theory and Applications (LATA'09)*. Vol. 5457. LNCS. Tarragona, Spain: Springer, Apr. 2009, pp. 152–163. DOI: 10.1007/978-3-642-00982-2_13. URL: http://hal.inria.fr/inria-00424283/en.

[32] P. Bhaduri and I. Stierand. 'A proposal for real-time interfaces in SPEEDS'. In: *Design, Automation and Test in Europe (DATE'10)*. IEEE, 2010, pp. 441–446.

[33] S. Bliudze. 'Un cadre formel pour l'étude des systèmes industriels complexes: un exemple basé sur l'infrastructure de l'UMTS'. PhD thesis. Ecole Polytechnique, 2006.

[34] S. Bliudze and D. Krob. 'Modelling of Complex Systems: Systems as Dataflow Machines'. In: *Fundam. Inform.* 91.2 (2009), pp. 251–274.

[35] G. Boudol and K. G. Larsen. 'Graphical Versus Logical Specifications'. In: *Theor. Comput. Sci.* 106.1 (1992), pp. 3–20.

[36] B. Caillaud, M. Malandain and J. Thibault. 'Implicit Structural Analysis of Multimode DAE Systems'. In: *23rd ACM International Conference on Hybrid Systems: Computation and Control (HSCC 2020)*. to appear. Sydney, Australia, Apr. 2020.

[37] B. Caillaud, B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen and A. Wasowski. 'Compositional design methodology with constraint Markov chains'. In: *QEST 2010*. Williamsburg, Virginia, United States, Sept. 2010. DOI: 10.1109/QEST.2010.23. URL: http://hal.inria.fr/inria-00591578/en.

[38] B. Caillaud, B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen and A. Wasowski. 'Constraint Markov Chains'. In: *Theoretical Computer Science* 412.34 (May 2011), pp. 4373–4404. DOI: 10.1016/j.tcs.2011.05.010. URL: http://hal.inria.fr/hal-00654003/en.

[39] S. L. Campbell and C. W. Gear. 'The index of general nonlinear DAEs'. In: *Numerische Mathematik* 72.2 (Dec. 1995), pp. 173–196. DOI: 10.1007/s002110050165. URL: http://dx.doi.org/10.1007/s002110050165.

[40] A. Chakrabarti. 'A Framework for Compositional Design and Analysis of Systems'. PhD thesis. EECS Department, University of California, Berkeley, Dec. 2007. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-174.html.

[41] A. Chakrabarti, L. de Alfaro, T. A. Henzinger and M. Stoelinga. 'Resource Interfaces'. In: *EMSOFT*. Ed. by R. Alur and I. Lee. Vol. 2855. Lecture Notes in Computer Science. Springer, 2003, pp. 117–133.

[42] E. Y. Chang, Z. Manna and A. Pnueli. 'Characterization of Temporal Property Classes'. In: *ICALP*. Ed. by W. Kuich. Vol. 623. Lecture Notes in Computer Science. Springer, 1992, pp. 474–486.

[43] K. Claessen and J. Hughes. 'QuickCheck: a lightweight tool for random testing of Haskell programs'. In: *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00), Montreal, Canada, September 18-21, 2000*. Ed. by M. Odersky and P. Wadler. ACM, 2000, pp. 268–279. DOI: 10.1145/351240.351266. URL: https://doi.org/10.1145/351240.351266.

[44] E. Clarke, O. Grumberg and D. Peled. *Model Checking*. MIT Press, 1999.

[45] N. J. Cutland, ed. *Nonstandard analysis and its applications*. Cambridge Univ. Press, 1988.

[46] A. David, K. G. Larsen, A. Legay, U. Nyman and A. Wasowski. 'ECDAR: An Environment for Compositional Design and Analysis of Real Time Systems'. In: *Automated Technology for Verification and Analysis - 8th International Symposium, ATVA 2010, Singapore, September 21-24, 2010. Proceedings*. 2010, pp. 365–370.

[47] A. David, K. G. Larsen, A. Legay, U. Nyman and A. Wasowski. 'Timed I/O automata: a complete specification theory for real-time systems'. In: *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, April 12-15, 2010*. 2010, pp. 91–100.

[48] B. Delahaye, J.-P. Katoen, K. G. Larsen, A. Legay, M. L. Pedersen, F. Sher and A. Wasowski. 'Abstract Probabilistic Automata'. In: *VMCAI*. Ed. by R. Jhala and D. A. Schmidt. Vol. 6538. Lecture Notes in Computer Science. Springer, 2011, pp. 324–339.

[49] F. Diener and G. Reeb. *Analyse non standard*. Hermann, 1989.

[50] D. L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits*. ACM Distinguished Dissertations. MIT Press, 1989.

[51] J. Edmonds and R. M. Karp. 'Theoretical improvements in algorithmic efficiency for network flow problems'. In: *Journal of the ACM* 19.2 (1972), pp. 248–264. DOI: 10.1145/321694.321699. URL: http://dx.doi.org/10.1145/321694.321699.

[52] H. Elmqvist, S. E. Mattsson and M. Otter. 'Modelica extensions for Multi-Mode DAE Systems'. In: *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Linköping University Electronic Press, Mar. 2014. DOI: 10.3384/ecp14096183.

[53] H. Elmqvist, A. Neumayr and M. Otter. 'Modia-dynamic modeling and simulation with julia'. In: *Juliacon'18*. University College London, UK, Aug. 2018.

[54] H. J. Ferreau, S. Almér, H. Peyrl, J. L. Jerez and A. Domahidi. 'Survey of industrial applications of embedded model predictive control'. In: *2016 European Control Conference (ECC)*. 2016, pp. 601–601. DOI: 10.1109/ECC.2016.7810351.

[55] A. V. Goldberg and R. E. Tarjan. 'A new approach to the maximum flow problem'. In: *Proceedings of the eighteenth annual ACM symposium on Theory of computing (STOC'86)*. 1986. DOI: 10.1145/12130.12144. URL: http://dx.doi.org/10.1145/12130.12144.

[56] *IEEE Standard VHDL Analog and Mixed-Signal Extensions, Std 1076.1-1999*. 1999. DOI: 10.1109/IEEESTD.1999.90578. URL: http://dx.doi.org/10.1109/IEEESTD.1999.90578.

[57] Y. Iwasaki, A. Farquhar, V. Saraswat, D. Bobrow and V. Gupta. 'Modeling Time in Hybrid Systems: How Fast Is "Instantaneous"?' In: *IJCAI*. 1995, pp. 1773–1781.

[58] J.-B. Jeannin, K. Ghorbal, Y. Kouskoulas, R. Gardner, A. Schmidt, E. Zawadzki and A. Platzer. 'Formal verification of ACAS X, an industrial airborne collision avoidance system'. In: *2015 International Conference on Embedded Software, EMSOFT 2015, Amsterdam, Netherlands, October 4-9, 2015*. Ed. by A. Girault and N. Guan. IEEE, 2015, pp. 127–136.

[59] L. Lamport. 'Proving the Correctness of Multiprocess Programs'. In: *IEEE Trans. Software Eng.* 3.2 (1977), pp. 125–143.

[60] K. G. Larsen, U. Nyman and A. Wasowski. 'On Modal Refinement and Consistency'. In: *Proc. of the 18th International Conference on Concurrency Theory (CONCUR'07)*. Springer, 2007, pp. 105–119.

[61] K. G. Larsen and B. Thomsen. 'A Modal Process Logic'. In: *Proceedings of the Third Annual Symposium on Logic in Computer Science (LICS'88)*. IEEE, 1988, pp. 203–210.

[62] T. Lindstrøm. 'An Invitation to Nonstandard Analysis'. In: *Nonstandard Analysis and its Applications*. Ed. by N. J. Cutland. Cambridge Univ. Press, 1988, pp. 1–105.

[63] N. A. Lynch. 'Input/Output Automata: Basic, Timed, Hybrid, Probabilistic and Dynamic'. In: *CONCUR*. Ed. by R. M. Amadio and D. Lugiez. Vol. 2761. Lecture Notes in Computer Science. Springer, 2003, pp. 187–188.

[64] N. A. Lynch and E. W. Stark. 'A Proof of the Kahn Principle for Input/Output Automata'. In: *Inf. Comput.* 82.1 (1989), pp. 81–92.

[65] Z. Manna and A. Pnueli. *Temporal verification of reactive systems: Safety*. Springer, 1995.

[66] B. Meyer. 'Applying "Design by Contract"'. In: *Computer* 25.10 (Oct. 1992), pp. 40–51. DOI: 10.1109/2.161279. URL: http://dx.doi.org/10.1109/2.161279.

[67] P. Nuzzo, A. L. Sangiovanni-Vincentelli, X. Sun and A. Puggelli. 'Methodology for the Design of Analog Integrated Interfaces Using Contracts'. In: *IEEE Sensors Journal* 12.12 (Dec. 2012), pp. 3329–3345.

[68] C. Pantelides. 'The consistent initialization of differential-algebraic systems'. In: *SIAM J. Sci. Stat. Comput.* 9.2 (1988), pp. 213–231.

[69] J. D. Pryce. 'A Simple Structural Analysis Method for DAEs'. In: *BIT Numerical Mathematics* 41.2 (Mar. 2001), pp. 364–394. DOI: 10.1023/a:1021998624799. URL: http://dx.doi.org/10.1023/a:1021998624799.

[70] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay and R. Passerone. 'A Modal Interface Theory for Component-based Design'. In: *Fundamenta Informaticae* 108.1-2 (2011), pp. 119–149. DOI: 10.3233/FI-2011-416. URL: http://hal.inria.fr/inria-00554283/en.

[71] A. Robinson. *Non-Standard Analysis*. ISBN 0-691-04490-2. Princeton Landmarks in Mathematics, 1996.

[72] E. Sikora, B. Tenbergen and K. Pohl. 'Industry needs and research directions in requirements engineering for embedded systems'. In: *Requirements Engineering* 17 (2012), pp. 57–78. DOI: 10.1007/s00766-011-0144-x. URL: http://link.springer.com/article/10.1007/s00766-011-0144-x.