IN PARTNERSHIP WITH:
Université Rennes 1

Activity Report 2019

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)

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Embedded and Real-time Systems

Table of contents

Project-Team HYCOMES

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

Keywords:

Computer Science and Digital Science:

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, Visitors, External Collaborators

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

Post-Doctoral Fellow
Benoît Vernay [Inria, Post-Doctoral Fellow, until Mar 2019]

PhD Students
Christelle Kozaily [Inria, PhD Student]
Aurélien Lamercerie [Univ de Rennes I, PhD Student, also with SemLis team of IRISA]
Joan Thibault [Univ de Rennes I, PhD Student, from Sep 2019]

Technical staff
Mathias Malandain [Inria, Engineer]

Interns and Apprentices
Antoine Geimer [CNRS, from Jul 2019 until Aug 2019]
Joan Thibault [Ecole Normale Supérieure Rennes, from Feb 2019 until Jun 2019]

Administrative Assistant
Armelle Mozziconacci [CNRS, Administrative Assistant]

2. Overall Objectives

2.1. 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 [12] 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, can be tainted with uncertainty. A main source of difficulty lies in the failure to properly handle the discrete and the continuous parts of systems, and their interaction. How the propagation of mode changes and resets 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 [19], [15] and [16].

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 [15], [19], [17], [16]. The following text provides a brief summary of this theory and gives some hints on its usefulness in the context

[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

of hybrid systems modeling. This presentation is based on our paper [2], a chapter of Simon Bliudze's PhD thesis [25], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [49].

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 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 [58], [41], [11]. 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. [45] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [26], [25] 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:

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

where F is a system of n_e equations $\{f_1, \cdots, f_{n_e}\}$ and x is a finite list of n_v independent real-valued, smooth enough, functions $\{x_1, \cdots, x_{n_v}\}$ of the independent variable t. We use x' as a shorthand for the list of first-order time derivatives of x_j, $j = 1, \cdots, n_v$. 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 [55], 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 $C = \{c_1, \cdots, c_n\}$ such that system

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

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* [32] 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 [55]. 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 Σ-method

Albeit less renowned that the Pantelides method, Pryce's Σ-method [56] 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:

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

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 [44] or the Edmonds-Karp algorithm [43] 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 $(C, D) = (\{c_1, \cdots, c_n\}, \{d_1, \cdots, d_n\})$ 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 $\{(i, j_i)\}_{i\in\{1,\cdots,n\}}$:

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

From the results proved by Pryce in [56], 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 [31].

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 $x_j^{(d_j)}$, $j = 1 \cdots n$. 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 [10]. 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 type. 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 [53]. 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 tool sets 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 [46], [35], [52], [14], [37]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [42]. A/G-contracts were advocated in [18] and are is still a very active research topic, with several contributions dealing with the timed [24] and probabilistic [29], [30] viewpoints in system design, and even mixed-analog circuit design [54].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch Input/Output Automata [51], [50]. Interface Automata [61], [60], [62], [33] 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 [57] inherit from both Interface Automata and the originally unrelated notion of Modal Transition System [48], [13], [27], [47]. 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 [63], [21], [23], [39], [38], [22], probabilistic [29], [40] and energy-aware [34] 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 [59]. 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. Highlights of the Year

4.1. Highlights of the Year

The Hycomes team has reached in 2019 an important milestone in the team's research objectives: the design and implementation of an implicit structural analysis algorithm supporting multimode DAE systems. This method is based on an encoding of the varying structure of a multimode DAE as Boolean functions, represented with Binary Decision Diagrams (BDD). This enables a complete strucutral analysis of a multimode DAE system, without enumerating its modes.

5. New Software and Platforms

5.1. Demodocos

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.

- Participants: Aurélien Lamercerie and Benoît Caillaud
- Contact: Benoît Caillaud
- Publication: Surgical Process Mining with Test and Flip Net Synthesis

5.2. MICA

Model Interface Compositional Analysis Library

KEYWORDS: Modal interfaces - Contract-based desing

SCIENTIFIC DESCRIPTION: In Mica, systems and interfaces are represented by extension. However, a careful design of the state and event heap enables the definition, composition and analysis of reasonably large systems and interfaces. The heap stores states and events in a hash table and ensures structural equality (there is no duplication). Therefore complex data-structures for states and events induce a very low overhead, as checking equality is done in constant time.

Thanks to the Inter module and the mica interactive environment, users can define complex systems and interfaces using Ocaml syntax. It is even possible to define parameterized components as Ocaml functions.

FUNCTIONAL DESCRIPTION: Mica is an Ocaml library implementing the Modal Interface algebra. The purpose of Modal Interfaces is to provide a formal support to contract based design methods in the field of system engineering. Modal Interfaces enable compositional reasoning methods on I/O reactive systems.

- Participant: Benoît Caillaud
- Contact: Benoît Caillaud
- URL: http://www.irisa.fr/s4/tools/mica/

5.3. IsamDAE

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 for testing 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. Evaluation functions make it possible to return the lists of leading equations and leading variables, as well as the actual scheduling of blocks, in a specified mode.

IsamDAE is coded in OCaml, and uses the following packages: * MLBDD by Arlen Cox, * Menhir by François Pottier and Yann Régis-Gianas, * Pprint by François Pottier, * XML-Light by Nicolas Cannasse and Jacques Garrigue.

RELEASE FUNCTIONAL DESCRIPTION: Version 0.2: * MEL: ad hoc language for the declaration of variable dimension multi-mode DAE systems * automatic parsing, model checking and model allocation * XML output for the list of evaluation blocks (parameters, equations, unknowns to be computed) * new algorithms for the mode-dependent scheduling and the evaluation of the scheduling in a given mode

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.

- Authors: Benoît Caillaud and Mathias Malandain
- Contact: Benoît Caillaud

6. New Results

6.1. 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 $f(\dot{x}, x, u) = 0$. This facilitates modeling from first principles of the physics. This year we completed 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.

A big comprehensive document has been written, which will be finalized and submitted next year.

6.2. Structural analysis of multimode DAE systems

Participants: Albert Benveniste, Benoît Caillaud, Khalil Ghorbal, Mathias Malandain.

The Hycomes team has obtained two results related to the structural analysis of multimode DAE systems.

6.2.1. Impulsive behavior of multimode DAE systems

A major difficulty with multimode DAE systems are the commutations from one mode to another one when the number of equations may change and variables may exhibit impulsive behavior, meaning that not only the trajectory of the system may be discontinuous, but moreover, some variables may be Dirac measures at the instant of mode changes. In [7] , we compare two radically different approaches to the structural analysis problem of mode changes. The first one is a classical approach, for a restricted class of DAE systems, for which the existence and uniqueness of an impulsive state jump is proved. The second approach is based on nonstandard analysis and is proved to generalize the former approach, to a larger class of multimode DAE systems. The most interesting feature of the latter approach is that it defines the state-jump as the standardization of the solution of a system of system of difference equations, in the framework of nonstandard analysis.

6.2.2. An implicit structural analysis method for multimode DAE systems

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 [31] 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 [56] 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 750 equations and 10^{23} modes.

6.3. Functional Decision Diagrams: A Unifying Data Structure For Binary Decision Diagrams

Participants: Joan Thibault, Khalil Ghorbal.

Zero-suppressed binary Decision Diagram (ZDD) is a notable alternative data structure of Reduced Ordered Binary Decision Diagram (ROBDD) that achieves a better size compression rate for Boolean functions that evaluate to zero almost everywhere. Deciding *a priori* which variant is more suitable to represent a given Boolean function is as hard as constructing the diagrams themselves. Moreover, converting a ZDD to a ROBDD (or vice versa) often has a prohibitive cost. This observation could be in fact stated about almost all existing BDD variants as it essentially stems from the non-compatibility of the reduction rules used to build such diagrams. Indeed, they are neither interchangeable nor composable. In [8], we investigate a novel functional framework, termed Lambda Decision Diagram (LDD), that ambitions to classify the already existing variants as implementations of special LDD models while suggesting, in a principled way, new models that exploit application-dependant properties to further reduce the diagram's size. We show how the reduction rules we use locally capture the global impact of each variable on the output of the entire function. Such knowledge suggests a variable ordering that sharply contrasts with the static fixed global ordering in the already existing variants as well as the dynamic reordering techniques commonly used.

7. Bilateral Contracts and Grants with Industry

7.1. Glose: Globalisation for Systems Engineering

Participants: Benoît Caillaud, Benoît Vernay.

Glose is a bilateral collaboration between Inria and Safran Tech., the corporate research entity of Safran Group. It started late 2017 for a duration of 44 months. Three Inria teams are involved in this collaboration: Diverse (Inria Rennes), Hycomes and Kairos (Inria Sophia-Antipolis). The scope of the collaboration is systems engineering and co-simulation.

The simulation of system-level models requires synchronizing, at simulation-time, physical models with software models. These models are developed and maintained by different stakeholders: physics engineers, control engineers and software engineers. Models designed by physics engineers are either detailed 3D finite-elements models, with partial differential equations (PDEs), or finite-dimension 0D models (obtained by model reduction techniques, or by empirical knowledge) expressed in modeling languages such as Simulink (with ordinary differential equations, or ODEs), Modelica (with differential algebraic equations, or DAEs), or directly as a C code embedding both the differential equations and its discretization scheme. Coupling together heterogeneous models and programs, so that they can be co-simulated, is not only a technological challenge, but more importantly raises several deep and difficult questions: Can we trust simulations? What about their reproducibility? Will it be possible to simulate large systems with hundreds to thousands of component models?

Co-simulation requires that models are provided with interfaces, specifying static and dynamic properties about the model and its expected environments. Interfaces are required to define how each model may synchronize and communicate, and how the model should be used. For instance, an interface should define (i) which variables are inputs, which are outputs, (ii) their data types, physical units, and sampling periods, but also (iii) the environmental assumptions under which the model is valid, and (iv) the causal dependencies between input and output variables and for continuous-time models, (v) the stiffness of the model, often expressed as a time-varying Jacobian matrix.

Formally, an interface is an abstraction of a model's behavior. A typical example of interface formalism for 0D continuous-time models is the FMI standard. Co-simulation also requires that a model of the system architecture is provided. This architectural model specifies how components are interconnected, how they communicate and how computations are scheduled. This is not limited to the topology of the architecture, and should also specify how components interact. For instance, variables in continuous-time models may have different data-types and physical units. Conversion may be required when continuous-time models are plugged together. Another fine example is the coupling of a 3D finite-element model to a 0D model: effort and flow fields computed in the 3D model must be averaged in a scalar value, before it can be sent to the 0D model, and conversely, scalar values computed by the 0D model must be distributed as a (vector) field along a boundary manifold of the 3D model. For discrete-time models (eg., software), components may communicate in many ways (shared variables, message passing, ...), and computations can be time- or event-triggered. All these features are captured as data-/behavior-coordination patterns, as exemplified by the GEMOC initiative [5].

In the Glose project, we propose to formalize the behavioral semantics of several modeling languages used at system-level. These semantics will be used to extract behavioral language interfaces supporting the definition of coordination patterns. These patterns, in turn, can systematically be used to drive the coordination of any model conforming to these languages. The co-simulation of a system-level architecture consists in an orchestration of hundreds to thousands of components. This orchestration is achieved by a master algorithm, in charge of triggering the communication and computation steps of each component. It takes into account the components' interfaces, and the data-/behavior-coordination patterns found in the system architecture model. Because simulation scalability is a major issue, the scheduling policy computed by the master algorithm should be optimal. Parallel or distributed simulations may even be required. This implies that the master algorithm should be hierarchical and possibly distributed.

[5] http://gemoc.org

In 2019, the Hycomes team has been working on the use of Quantized State System (QSS) nethods for the cosimulation of aeronautics system models. The aim is to design new distributed simulation protocols, capable of simulating large, but heterogeneous system models. The investigation is on the trade-offs between pessimistic simulation techniques, where no roll-back is required, and speculative methods, where roll-back may be required. The latter method can be beneficial to the performance and scalability of the simulation, provided roll-backs do not happen too often. The models under consideration are cyberphysical systems consisting in both Modelica models (for the physics) and discrete-time models expressed in a dedicated language (for the control).

In 2019, the Hycones team has delivered one report, detailing the state-of-the-art techniques for continuous systems cosimulation.

8. Partnerships and Cooperations

8.1. Regional Initiatives

Participants: Benoît Caillaud, Aurélien Lamercerie.

The Hycomes has been participating to the SUNSET project (2016–2019) of the CominLabs excellence laboratory [6]. This project focuses on the computation of surgical procedural knowledge models from recordings of individual procedures, and their execution [28]. The objective is to develop an enabling technology for procedural knowledge based computer assistance of surgery. In this project, we demonstrate its potential added value in nurse and surgeon training. The main contribution of the Hycomes team to this project has been the development of Demodocos, a process model synthesis tool, capable of generating models of a surgical procedure, from a few recordings of actual procedures. Demodocos has been interfaced to the `#SEVEN` virtual reality scenario modeling language and engine, developed in the Hybrid team at Inria Rennes. In 2019, the team has contributed to two publications presenting experimental results of the SUNSET project [9][6].

8.2. National Initiatives

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

The project gathers researchers from three Inria teams, and from three other research labs in Grenoble and Paris area.

Name	*Team*	*Inria Center or Laboratory*
Vincent Acary Bernard Brogliato Alexandre Rocca	Tripop	Inria Grenoble Rhône Alpes
Albert Benveniste Benoît Caillaud Khalil Ghorbal Christelle Kozaily Mathias Malandain Benoît Vernay	Hycomes	Inria Rennes Bretagne Atlantique
Marc Pouzet Tim Bourke Imsail Lakhim-Bennani	Parkas	ENS & Inria Paris
Goran Frehse	SSH	ENSTA Paris-Tech.
Antoine Girard		L2S-CNRS, Saclay
Eric Goubault Sylvie Putot	Cosynus	LIX, École Polytechnique, Saclay

[6] http://www.s3pm.cominlabs.ueb.eu/

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).

MiodeliScale 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 2019, three general meetings have been organized, 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 [36].

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

Participants: Albert Benveniste, Benoît Caillaud, Khalil Ghorbal, Mathias Malandain.

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. One of the main contributions expected from Inria are:

- A novel structural analysis algorithms 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 2019, the effort has been put on the first objective, and two important milestones have been reached:

- The design of a novel algorithm for the structural analysis of multimode DAE systems. This algorithm is a generalization of the Pryce structural analysis method to the multimode case. The key feature of our method is that it works on implicit representations of the set of modes, and of the varying structure of the multimode DAE. In other words, it does not imply the enumeration of the system's modes. Performing the structural analysis at compile-time brings two decisive advantages: 1/ it allows to deliver to the user precise diagnostics about the model, and can be compared type-checking in programming languages; 2/ it is instrumental for the generation of efficient simulation code. Our algorithm is the first method enabling the compile-time analysis of systems with extremely large combinatorics of modes.

- Our multimode DAE structural analysis algorithm has been implemented in IsamDAE, a software comprizing an algorithmic library, to be used in modeling language compilers (Modelica tools) and a standalone tool, to be used independently of a complex Modelica toolset. IsamDAE has allowed to benchmark the method against several families of models, inspired by case-studies developed by industrial partners of the FUI ModeliScale project. Despite the tool is still under development, we have already been able to deal with models with up to 10^{23} modes.

On top of these two main results, the Hycomes team has started investigating the use of Quantized Space Systems (QSS), for the simulation of large DAE systems. QSSs simulation (QSS) was introduced in the early 2000's by F. Cellier and E. Kofman as an alternative to time-based simulation, which is the dominant approach to ODE/DAE systems simulation. Rather than linking QSS to Discrete Event Simulation, we propose to relate it to Synchronous Programming and its continuous time extension Zelus. In the deliverable [20], we expose our understanding of QSS and its variants, then we propose ideas toward a QSS-based cosimulation, by building on top of our knowledge on distributed executions of synchronous programs.

The plan for 2020 is to extend our structural analysis to cover impulsive mode changes and the consistent initialization problem, in the multimode case. A coupling of IsamDAE with Dymola (Dassault Système's commercial implementation of the Modelica language) is under development.

Another future development is to turn our structural analysis method to a compositional method, where large models could be considered by parts. This is a key problem in the Modelica language, as the compilation of a Modelica model is not modular.

Work on QSS methods will continue, and we envision to prototype a QSS-based distributed simulation method for hybrid ODE systems, based on the Zélus language.

8.3. International Initiatives

8.3.1. Inria International Partners

8.3.1.1. Informal International Partners

We have a long standing informal collaboration with Martin Otter (DLR, Munich, Germany) and Hilding Helmqvist (Mogram AB, Lund, Sweden). In 2019, this fruitful collaboration has resulted in one publication [7]. The publication draws links between two radically different, but equivalent approaches to the same problem: the impulsive behavior of some multimode DAE, when it is switching from one mode to another. The first approach relies on a transformation of the multimode DAE system to a special index one form, for which state-jumps are proved to be solution of a system of algebraic equations relating right limits to left limits. The second approach builds on the use of nonstandard analysis, combined with the heritage of synchronous programming languages, particularly on the concept of constructive semantics. This gives a formulation of the state-jumps, as a system of difference equations, with an infinitesimal time-step. The latter approach is more general than the former, in the sense that impulsive behavior can be characterized for a larger class of multimode DAE systems. Yet, both approaches coincide on a restricted class of multimode DAEs.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events: Selection

9.1.1.1. Member of the Conference Program Committees

- Albert Benveniste has served on the Program Committee of the Modelica Conference 2019.
- Khalil Ghorbal has served on the Program Committee of the VMCAI 2019 conference.
- Benoît Caillaud has served on the Steering Committee of the ACSD 2019 conference.

9.1.2. Invited Talks

Albert Benveniste has given an invited talk titled "Why considering nonstandard semantics for hybrid systems and how to reconcile it with superdense time semantics" at the Oded Maler Memorial workshop at the HSCC'19 conference in Montreal, Canada.

9.1.3. Leadership within the Scientific Community

Albert Benveniste is member of the French Académie des Technologies.

9.1.4. Scientific Expertise

- In 2019, Benoît Caillaud has reviewed collaborative research project proposals submitted to the French national funding agency ANR. As an Inria Evaluation Committee member, he has served on several Inria hiring and promotion committees.
- Albert Benveniste is president of the Scientific Council of Orange and member of the Scientific Council of Safran.

9.1.5. Research Administration

- Albert Benveniste is member of the Burex (Executive Bureau) of the Cominlabs Labex [7].
- Benoît Caillaud is in charge of the IPL ModeliScale [8] national initiative funded by Inria. He is also head of the Programming Languages & Software Engineering department [9] of IRISA.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master : Khalil Ghorbal, *Analyse et Conception Formelles*, M1, (chargé de TD), 22h EqTD, University Rennes 1 and ENS Rennes, France

Master : Khalil Ghorbal, Solvers Principle and Architectures, 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

9.2.2. Supervision

PhD: Christelle Kozaily, Structural analysis of nonsmooth dynamical systems, university of Rennes 1, co-supervised by Vincent Acary (Tripop [10] team at Inria Grenoble), Benoît Caillaud and Kahlil 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 [11] team of IRISA), started December 2017.

PhD: Joan Thibault, Structural Analysis Techniques for Binary Decision Diagrams, university of Rennes 1, co-supervised by Benoît Caillaud and Khalil Ghorbal.

9.2.3. Juries

Benoît Caillaud has served as president of the jury for Hugo Bazille's PhD defense, at the University of Rennes 1.

9.3. Popularization

9.3.1. Internal action

The Hycomes team has hosted short internships for secondary school students. This has been an opportunity to promote women in computing, since three female students visited the team for four days, to discover what scientific research is, and what research in computer science could mean. All team members contributed to the initiative.

[7] https://cominlabs.u-bretagneloire.fr/governance
[8] https://team.inria.fr/modeliscale/
[9] http://www.irisa.fr/en/departments/d4-language-and-software-engineering
[10] https://team.inria.fr/tripop/
[11] https://www-semlis.irisa.fr

10. Bibliography

Major publications by the team in recent years

[1] A. BENVENISTE, T. BOURKE, B. CAILLAUD, J.-L. COLAÇO, C. PASTEUR, M. POUZET. *Building a Hybrid Systems Modeler on Synchronous Languages Principles*, in "Proceedings of the IEEE", September 2018, vol. 106, n° 9, pp. 1568–1592 [*DOI :* 10.1109/JPROC.2018.2858016], https://hal.inria.fr/hal-01879026

[2] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *Non-standard semantics of hybrid systems modelers*, in "Journal of Computer and System Sciences", 2012, vol. 78, n° 3, pp. 877-910, This work was supported by the SYNCHRONICS large scale initiative of Inria [*DOI :* 10.1016/J.JCSS.2011.08.009], 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, K. G. LARSEN. *Contracts for System Design*, in "Foundations and Trends in Electronic Design Automation", 2018, vol. 12, n° 2-3, pp. 124-400 [*DOI :* 10.1561/1000000053], https://hal.inria.fr/hal-01971429

[4] J.-B. JEANNIN, K. GHORBAL, Y. KOUSKOULAS, A. SCHMIDT, R. GARDNER, S. MITSCH, 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", November 2017, vol. 19, n° 6, pp. 717-741 [*DOI :* 10.1007/S10009-016-0434-1], https://hal.archives-ouvertes.fr/hal-01232365

[5] A. SOGOKON, K. GHORBAL, T. T. JOHNSON. *Operational Models for Piecewise-Smooth Systems*, in "ACM Transactions on Embedded Computing Systems (TECS)", October 2017, vol. 16, n° 5s, pp. 185:1–185:19 [*DOI :* 10.1145/3126506], https://hal.inria.fr/hal-01658196

Publications of the year

Articles in International Peer-Reviewed Journals

[6] M.-S. BRACQ, E. MICHINOV, B. ARNALDI, B. CAILLAUD, B. GIBAUD, V. GOURANTON, P. JANNIN. *Learning procedural skills with a virtual reality simulator An acceptability study*, in "Nurse Education Today", August 2019, vol. 79, pp. 153-160 [*DOI :* 10.1016/J.NEDT.2019.05.026], https://hal-univ-rennes1.archives-ouvertes.fr/hal-02150192

Scientific Books (or Scientific Book chapters)

[7] A. BENVENISTE, B. CAILLAUD, H. ELMQVIST, K. GHORBAL, M. OTTER, M. POUZET. *Multi-Mode DAE Models - Challenges, Theory and Implementation*, in "Computing and Software Science: State of the Art and Perspectives", Lecture Notes in Computer Science, Springer, October 2019, vol. 10000, pp. 283-310 [*DOI :* 10.1007/978-3-319-91908-9_16], https://hal.inria.fr/hal-02333603

Research Reports

[8] J. THIBAULT, K. GHORBAL. *Functional Decision Diagrams: A Unifying Data Structure For Binary Decision Diagrams*, Inria Rennes - Bretagne Atlantique and University of Rennes 1, France, November 2019, n° RR-9306, https://hal.inria.fr/hal-02369112

Other Publications

[9] M.-S. BRACQ, E. MICHINOV, B. ARNALDI, A. AUDINOT, B. CAILLAUD, B. GIBAUD, V. GOURANTON, P.-L. HENAUX, A. LAMERCERIE, B. NOGUES, P. JANNIN. *Modeling, Simulation and Training Procedural Skills: User experience and acceptability of a virtual reality simulator for scrub nurses in neurosurgery*, January 2019, vol. 36, 1 p. , 19th International Meeting on Simulation in Healthcare (IMSH), Poster, https://hal.archives-ouvertes.fr/hal-02123682

[10] A. LAMERCERIE. *Une algèbre des automates d'acceptation propositionnelle déterministes comme théorie d'interface pour la conception de systèmes cyberphysiques*, November 2019, 1 p. , MSR 2019 - 12ème Colloque sur la Modélisation des Systèmes Réactifs, Nov 2019, Angers, France, Poster, https://hal.archives-ouvertes.fr/hal-02432696

References in notes

[11] N. J. CUTLAND (editor). *Nonstandard analysis and its applications*, Cambridge Univ. Press, 1988

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

[13] A. ANTONIK, M. HUTH, K. G. LARSEN, U. NYMAN, A. WASOWSKI. *20 Years of Modal and Mixed Specifications*, in "Bulletin of European Association of Theoretical Computer Science", 2008, vol. 1, n^{o} 94

[14] C. BAIER, J.-P. KATOEN. *Principles of Model Checking*, MIT Press, Cambridge, 2008

[15] A. BENVENISTE, T. BOURKE, B. CAILLAUD, J.-L. COLAÇO, C. PASTEUR, M. POUZET. *Building a Hybrid Systems Modeler on Synchronous Languages Principles*, in "Proceedings of the IEEE", September 2018, vol. 106, n^{o} 9, pp. 1568–1592 [*DOI :* 10.1109/JPROC.2018.2858016], https://hal.inria.fr/hal-01879026

[16] A. BENVENISTE, T. BOURKE, B. CAILLAUD, B. PAGANO, M. POUZET. *A Type-Based Analysis of Causality Loops In Hybrid Systems Modelers*, December 2013, Deliverable D3.1_1 v 1.0 of the Sys2soft collaborative project "Physics Aware Software", https://hal.inria.fr/hal-00938866

[17] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *Semantics of multi-mode DAE systems*, August 2013, Deliverable D.4.1.1 of the ITEA2 Modrio collaborative project, https://hal.inria.fr/hal-00938891

[18] A. BENVENISTE, B. CAILLAUD, A. FERRARI, L. MANGERUCA, R. PASSERONE, 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)", Amsterdam, The Netherlands, Revised Lectures, Lecture Notes in Computer Science, Springer, October 2008, vol. 5382

[19] A. BENVENISTE, B. CAILLAUD, B. PAGANO, M. POUZET. *A type-based analysis of causality loops in hybrid modelers*, in "HSCC '14: International Conference on Hybrid Systems: Computation and Control", Berlin, Germany, Proceedings of the 17th international conference on Hybrid systems: computation and control (HSCC '14), ACM Press, April 2014, 13 p. [*DOI :* 10.1145/2562059.2562125], https://hal.inria.fr/hal-01093388

[20] A. BENVENISTE, M. POUZET, M. MALANDAIN. *Bibliography report on Quantized Space Systems simulation — Proposals for QSS-based co-simulation of large DAE systems*, Inria, January 2020, n° M2.2.2_1

[21] N. BERTRAND, A. LEGAY, S. PINCHINAT, J.-B. RACLET. *A Compositional Approach on Modal Specifications for Timed Systems*, in "11th International Conference on Formal Engineering Methods (ICFEM'09)", Rio de Janeiro, Brazil, LNCS, Springer, December 2009, vol. 5885, pp. 679-697, http://hal.inria.fr/inria-00424356/en

[22] N. BERTRAND, A. LEGAY, S. PINCHINAT, J.-B. RACLET. *Modal event-clock specifications for timed component-based design*, in "Science of Computer Programming", 2011, http://dx.doi.org/10.1016/j.scico.2011.01.007

[23] N. BERTRAND, S. PINCHINAT, J.-B. RACLET. *Refinement and Consistency of Timed Modal Specifications*, in "3rd International Conference on Language and Automata Theory and Applications (LATA'09)", Tarragona, Spain, LNCS, Springer, April 2009, vol. 5457, pp. 152-163 [*DOI :* 10.1007/978-3-642-00982-2_13], http://hal.inria.fr/inria-00424283/en

[24] P. BHADURI, I. STIERAND. *A proposal for real-time interfaces in SPEEDS*, in "Design, Automation and Test in Europe (DATE'10)", IEEE, 2010, pp. 441-446

[25] S. BLIUDZE. *Un cadre formel pour l'étude des systèmes industriels complexes: un exemple basé sur l'infrastructure de l'UMTS*, Ecole Polytechnique, 2006

[26] S. BLIUDZE, D. KROB. *Modelling of Complex Systems: Systems as Dataflow Machines*, in "Fundam. Inform.", 2009, vol. 91, n° 2, pp. 251–274

[27] G. BOUDOL, K. G. LARSEN. *Graphical Versus Logical Specifications*, in "Theor. Comput. Sci.", 1992, vol. 106, n° 1, pp. 3-20

[28] B. CAILLAUD. *Surgical Process Mining with Test and Flip Net Synthesis*, in "Application of Region Theory (ART)", Barcelona, Spain, R. BERGENTHUM, J. CARMONA (editors), July 2013, pp. 43-54, http://hal.inria.fr/hal-00872284

[29] B. CAILLAUD, B. DELAHAYE, K. G. LARSEN, A. LEGAY, M. L. PEDERSEN, A. WASOWSKI. *Compositional design methodology with constraint Markov chains*, in "QEST 2010", Williamsburg, Virginia, United States, September 2010 [*DOI :* 10.1109/QEST.2010.23], http://hal.inria.fr/inria-00591578/en

[30] B. CAILLAUD, B. DELAHAYE, K. G. LARSEN, A. LEGAY, M. L. PEDERSEN, A. WASOWSKI. *Constraint Markov Chains*, in "Theoretical Computer Science", May 2011, vol. 412, n° 34, pp. 4373-4404 [*DOI :* 10.1016/J.TCS.2011.05.010], http://hal.inria.fr/hal-00654003/en

[31] B. CAILLAUD, M. MALANDAIN, J. THIBAULT. *Implicit Structural Analysis of Multimode DAE Systems*, in "23rd ACM International Conference on Hybrid Systems: Computation and Control (HSCC 2020)", Sydney, Australia, April 2020, to appear

[32] S. L. CAMPBELL, C. W. GEAR. *The index of general nonlinear DAEs*, in "Numerische Mathematik", dec 1995, vol. 72, n° 2, pp. 173–196, http://dx.doi.org/10.1007/s002110050165

[33] A. CHAKRABARTI. *A Framework for Compositional Design and Analysis of Systems*, EECS Department, University of California, Berkeley, Dec 2007, http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-174.html

[34] A. CHAKRABARTI, L. DE ALFARO, T. A. HENZINGER, M. STOELINGA. *Resource Interfaces*, in "EMSOFT", R. ALUR, I. LEE (editors), Lecture Notes in Computer Science, Springer, 2003, vol. 2855, pp. 117-133

[35] E. Y. CHANG, Z. MANNA, A. PNUELI. *Characterization of Temporal Property Classes*, in "ICALP", W. KUICH (editor), Lecture Notes in Computer Science, Springer, 1992, vol. 623, pp. 474-486

[36] K. CLAESSEN, 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.", M. ODERSKY, P. WADLER (editors), ACM, 2000, pp. 268–279, https://doi.org/10.1145/351240.351266

[37] E. CLARKE, O. GRUMBERG, D. PELED. *Model Checking*, MIT Press, 1999

[38] A. DAVID, K. G. LARSEN, A. LEGAY, U. NYMAN, 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

[39] A. DAVID, K. G. LARSEN, A. LEGAY, U. NYMAN, 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

[40] B. DELAHAYE, J.-P. KATOEN, K. G. LARSEN, A. LEGAY, M. L. PEDERSEN, F. SHER, A. WASOWSKI. *Abstract Probabilistic Automata*, in "VMCAI", R. JHALA, D. A. SCHMIDT (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6538, pp. 324-339

[41] F. DIENER, G. REEB. *Analyse non standard*, Hermann, 1989

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

[43] J. EDMONDS, R. M. KARP. *Theoretical improvements in algorithmic efficiency for network flow problems*, in "Journal of the ACM", 1972, vol. 19, n^{o} 2, pp. 248–264, http://dx.doi.org/10.1145/321694.321699

[44] A. V. GOLDBERG, 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, http://dx.doi.org/10.1145/12130.12144

[45] Y. IWASAKI, A. FARQUHAR, V. SARASWAT, D. BOBROW, V. GUPTA. *Modeling Time in Hybrid Systems: How Fast Is "Instantaneous"?*, in "IJCAI", 1995, pp. 1773–1781

[46] L. LAMPORT. *Proving the Correctness of Multiprocess Programs*, in "IEEE Trans. Software Eng.", 1977, vol. 3, n^{o} 2, pp. 125-143

[47] K. G. LARSEN, U. NYMAN, A. WASOWSKI. *On Modal Refinement and Consistency*, in "Proc. of the 18th International Conference on Concurrency Theory (CONCUR'07)", Springer, 2007, pp. 105–119

[48] K. G. LARSEN, 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

[49] T. LINDSTRØM. *An Invitation to Nonstandard Analysis*, in "Nonstandard Analysis and its Applications", N. J. CUTLAND (editor), Cambridge Univ. Press, 1988, pp. 1–105

[50] N. A. LYNCH. *Input/Output Automata: Basic, Timed, Hybrid, Probabilistic and Dynamic*, in "CONCUR", R. M. AMADIO, D. LUGIEZ (editors), Lecture Notes in Computer Science, Springer, 2003, vol. 2761, pp. 187-188

[51] N. A. LYNCH, E. W. STARK. *A Proof of the Kahn Principle for Input/Output Automata*, in "Inf. Comput.", 1989, vol. 82, n^{o} 1, pp. 81-92

[52] Z. MANNA, A. PNUELI. *Temporal verification of reactive systems: Safety*, Springer, 1995

[53] B. MEYER. *Applying "Design by Contract"*, in "Computer", October 1992, vol. 25, n^{o} 10, pp. 40–51, http://dx.doi.org/10.1109/2.161279

[54] P. NUZZO, A. L. SANGIOVANNI-VINCENTELLI, X. SUN, A. PUGGELLI. *Methodology for the Design of Analog Integrated Interfaces Using Contracts*, in "IEEE Sensors Journal", Dec. 2012, vol. 12, n^{o} 12, pp. 3329–3345

[55] C. PANTELIDES. *The consistent initialization of differential-algebraic systems*, in "SIAM J. Sci. Stat. Comput.", 1988, vol. 9, n^{o} 2, pp. 213–231

[56] J. D. PRYCE. *A Simple Structural Analysis Method for DAEs*, in "BIT Numerical Mathematics", March 2001, vol. 41, n^{o} 2, pp. 364–394, http://dx.doi.org/10.1023/a:1021998624799

[57] J.-B. RACLET, E. BADOUEL, A. BENVENISTE, B. CAILLAUD, A. LEGAY, R. PASSERONE. *A Modal Interface Theory for Component-based Design*, in "Fundamenta Informaticae", 2011, vol. 108, n^{o} 1-2, pp. 119-149 [*DOI :* 10.3233/FI-2011-416], http://hal.inria.fr/inria-00554283/en

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

[59] E. SIKORA, B. TENBERGEN, K. POHL. *Industry needs and research directions in requirements engineering for embedded systems*, in "Requirements Engineering", 2012, vol. 17, pp. 57–78, http://link.springer.com/article/10.1007/s00766-011-0144-x

[60] L. DE ALFARO. *Game Models for Open Systems*, in "Verification: Theory and Practice", Lecture Notes in Computer Science, Springer, 2003, vol. 2772, pp. 269-289

[61] L. DE ALFARO, 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

[62] L. DE ALFARO, T. A. HENZINGER. *Interface-based design*, in "In Engineering Theories of Software Intensive Systems, proceedings of the Marktoberdorf Summer School", Kluwer, 2004

[63] L. DE ALFARO, T. A. HENZINGER, M. STOELINGA. *Timed Interfaces*, in "Proc. of the 2nd International Workshop on Embedded Software (EMSOFT'02)", Lecture Notes in Computer Science, Springer, 2002, vol. 2491, pp. 108–122