IN PARTNERSHIP WITH:
Université Rennes 1

Activity Report 2018

Project-Team HYCOMES

Hybrid systems modeling & contract-based design for cyber-physical systems

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, from Oct 2018]

PhD Student
Christelle Kozaily [Inria, from Oct 2018]

Technical staff
Jean Hany [Inria, until May 2018]
Mathias Malandain [Inria, from Oct 2018]

Interns
Nathan Koskas [Ecole normale supérieure de Rennes, from Jun 2018 until Aug 2018]
Julien Morane [Inria, from Apr 2018 until Jul 2018]
Joan Thibault [Ecole Normale Supérieure Rennes, from Jul 2018 until Aug 2018]

Administrative Assistant
Armelle Mozziconacci [CNRS]

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 [11] 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], [5] and [14].

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 [5], [19], [15], [14]. 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 [1], a chapter of Simon Bliudze's PhD thesis [24], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [48].

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 [54], [41], [10]. 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. [43] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [25], [24] 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. 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. 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 [52]. 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 [6]. 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 [45], [32], [51], [13], [34]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [42]. A/G-contracts were advocated by the SPEEDS project [18]. They were further experimented in the framework of the CESAR project [37], with the additional consideration of *weak* and *strong* assumptions. This is still a very active research topic, with several recent contributions dealing with the timed [23] and probabilistic [28], [29] viewpoints in system design, and even mixed-analog circuit design [53].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch Input/Output Automata [50], [49]. Interface Automata [57], [56], [58], [30] 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 [3] inherit from both Interface Automata and the originally unrelated notion of Modal Transition System [47], [12], [26], [46]. 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 [59], [20],

[22], [39], [38], [21], probabilistic [28], [40] and energy-aware [31] 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 [55]. DOORS projects collecting requirements are poorly structured and cannot be considered a formal modeling framework today. They are nothing more than an informal 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 performed the development of the fly-by-wire and of the landing gear subsystems.

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 highlights of the year are:

- The start of two industrial collaborations of crucial importance for the Hycomes team: (i) the FUI ModeliScale project, in the context of which the Hycomes team design novel algorithms for the structural analysis of multimode DAE systems, with the objective of supporting a larger class of multimode Modelica models; and (ii) the Glose project, in collaboration with Safran Tech., on the topics of cyber-physical systems modeling and cosimulation.
- Albert Benveniste, Benoît Caillaud and co-authors have published a book on contract-based reasoning for cyber-physical systems design. This book is the result of more than 10 years of research on contract and interface theories.
- Albert Benveniste, Benoît Caillaud and co-authors have published a paper in *The Proceedings of the IEEE* on the design of Hybrid Systems modeling languages, based on our past work on ODE-based synchronous languages (namely the Zélus language).

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
- URL: http://www.irisa.fr/prive/Benoit.Caillaud/Benoit_Caillauds_Professional_homepage/Software/Entries/2017/12/31_Demodocos__A_test_and_flip_net_synthesis_tool_for_maintenance_and_surgical_process_.html

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. TnF-C++

FUNCTIONAL DESCRIPTION: TnF-C++ is a robust and portable re-implementation of Flipflop, developed in 2014 and integrated in the S3PM toolchain. Both software have been designed in the context of the S3PM project on surgical procedure modeling and simulation,

- Contact: Benoît Caillaud

6. New Results

6.1. Hybrid Systems Modeling and Verification

6.1.1. Building a Hybrid Systems Modeler on Synchronous Languages Principles

Participants: Albert Benveniste, Benoît Caillaud.

Hybrid systems modeling languages that mix discrete and continuous time signals and systems are widely used to develop Cyber-Physical systems where control software interacts with physical devices. Compilers play a central role, statically checking source models, generating intermediate representations for testing and verification, and producing sequential code for simulation and execution on target platforms. In [5], Albert Benveniste, Timothy Bourke (PARKAS team Inria/ENS Paris), Benoît Caillaud, Jean-Louis Colaço, Cédric Pasteur (ANSYS/Esterel Technologies, Toulouse) and Marc Pouzet (PARKAS team Inria/ENS Paris) propose a comprehensive study of hybrid systems modeling languages (formal semantics, causality analysis, compiler design, ...). This paper advocates a novel approach to the design and implementation of these languages, built on synchronous language principles and their proven compilation techniques. The result is a hybrid systems modeling language in which synchronous programming constructs can be mixed with Ordinary Differential Equations (ODEs) and zero-crossing events, and a runtime that delegates their approximation to an off-the-shelf numerical solver. We propose an ideal semantics based on non standard analysis, which defines the execution of a hybrid model as an infinite sequence of infinitesimally small time steps. It is used to specify and prove correct three essential compilation steps: (1) a type system that guarantees that a continuous-time signal is never used where a discrete-time one is expected and conversely; (2) a type system that ensures the absence of combinatorial loops; (3) the generation of statically scheduled code for efficient execution. Our approach has been evaluated in two implementations: the academic language Zélus, which extends a language reminiscent of Lustre with ODEs and zero-crossing events, and the industrial prototype Scade Hybrid, a conservative extension of Scade 6.

6.1.2. Structural Analysis of Differential-Algebraic Equations (DAE), State-of-the-Art

Participants: Khalil Ghorbal, Mathias Malandain.

In a deliverable [5] for the FUI ModeliScale collaborative project, Mathias Malandain and Khalil Ghorbal discuss the state-of-the-art methods for performing what is called structural index reduction for differential-algebraic equations, that is equations involving both differential and algebraic equality constraints. Index reduction is one of the basic required methods implemented in any DAE-based modelling language (like Modelica). It is a mandatory step to perform prior to calling a numerical solver to effectively advance time by integrating the set of equations. We cover in particular a recent work that tackles extended models involving several modes, each of which is encoded as a standard DAE.

6.1.3. Multi-Mode DAE Models: Challenges, Theory and Implementation

Participants: Albert Benveniste, Benoît Caillaud, Khalil Ghorbal.

[5] Modeliscale project, deliverable M2.1.1 1, Structural Analysis of Differential-Algebraic Equations (DAE), State-of-the-Art.

The modeling and simulation of Cyber-Physical Systems (CPS) such as robots, vehicles, and power plants often require models with a time varying structure, due to failure situations or due to changes in physical conditions. These are called multi-mode models. In [17], Albert Benveniste, Benoît Caillaud, Hilding Elmqvist (Mogram AB, Lund, Sweden), Khalil Ghorbal, Martin Otter (DLR-SR, Oberpfaffenhofen, Germany) and Marc Pouzet (PARKAS team, Inria/ENS Paris) are interested in multi-domain, component-oriented modeling as performed, for example, with the modeling language Modelica that leads naturally to Differential Algebraic Equations (DAEs). This paper is thus about multi-mode DAE systems. In particular, new methods are introduced to overcome one key problem that was only solved for specific subclasses of systems before: How to switch from one mode to another one when the number of equations may change and variables may exhibit impulsive behavior? An evaluation is performed both with the experimental modeling and simulation system Modia, a domain specific language extension of the programming language Julia, and with SunDAE, a novel structural analysis library for multi-mode DAE systems.

6.1.4. Vector Barrier Certificates and Comparison Systems

Participant: Khalil Ghorbal.

Vector Lyapunov functions are a multi-dimensional extension of the more familiar (scalar) Lyapunov functions, commonly used to prove stability properties in systems of non-linear ordinary differential equations (ODEs). In [7], Kahlil Ghorbal and Andrew Sogokon (CMU, Pittsburgh, USA) explore an analogous vector extension for so-called barrier certificates used in safety verification. As with vector Lyapunov functions, the approach hinges on constructing appropriate comparison systems, i.e., related differential equation systems from which properties of the original system may be inferred. The paper presents an accessible development of the approach, demonstrates that most previous notions of barrier certificate are special cases of comparison systems, and discusses the potential applications of vector barrier certificates in safety verification and invariant synthesis.

6.2. Contract-based Reasoning for Cyper-Physical Systems Design

6.2.1. Contracts for Cyper-Physical Systems Design

Participants: Albert Benveniste, Benoît Caillaud.

Contract-based reasoningn has been proposed as an "orthogonal" approach that complements methodologies proposed so far to cope with the complexity of cyber-physical systems design. Contract-based reasoning provides a rigorous framework for the verification, analysis, abstraction/refinement, and even synthesis of cyber-physical systems. A number of results have been obtained in this domain but a unified treatment of the topic that can help put contract-based design in perspective was missing. In [6], Albert Benveniste, Benoît Caillaud and co-authors provide a unified theory where contracts are precisely defined and characterized so that they can be used in design methodologies with no ambiguity. This monograph gathers research results of the former S4 inria team. It identifies the essence of complex system design using contracts through a *mathematical meta-theory*, where all the properties of the methodology are derived from an abstract and generic notion of contract. We show that the meta-theory provides deep and enlightening links with existing contract and interface theories, as well as guidelines for designing new theories. Our study encompasses contracts for both software and systems, with emphasis on the latter. We illustrate the use of contracts with two examples: requirement engineering for a parking garage management, and the development of contracts for timing and scheduling in the context of the Autosar methodology in use in the automotive sector.

6.2.2. Cyber-Physical Systems Design: from Natural Language Requirements

In his current PhD work, co-supervised by Benoît Caillaud and Annie Forêt (SemLIS, IRISA, Rennes, France), Aurélien Lamercerie explores the construction of formal representations of natural language texts. The mapping from a natural language to a logical representation is realized with a grammatical formalism, linking the syntactic analysis of the text to a semantic representation. In [44], Aurélien Lamercerie targets behavioral specifications of cyber-physical systems, ie any type of system in which software components interact closely with a physical environment. The objective is the simulation and formal verification, by automatic or assisted methods, of system level requirements expressed in a controled fragment of a natural language.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

7.1.1. 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 [6].

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

[6] http://gemoc.org

be optimal. Parallel or distributed simulations may even be required. This implies that the master algorithm should be hierarchical and possibly distributed.

8. Partnerships and Cooperations

8.1. Regional Initiatives

Participant: Benoît Caillaud.

Benoît Caillaud is contributing to the SUNSET projects of the CominLabs excellence laboratory [7]. This project focuses on the computation of surgical procedural knowledge models from recordings of individual procedures, and their execution [27]. 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 [36], [35]. In 2018, Benoît Caillaud and Aurélien Lamercerie (SemLIS team of IRISA) have maintained and enhanced the Demodocos prototype software. This software is synthesizing surgical process models (expressed in the ♯Seven language developed in the Hybrid team, Inria Rennes) from instances of surgical procedures. These models can be executed in a virtual reality environment developed by the Hybrid team.

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

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).

[7] https://s3pm.cominlabs.u-bretagneloire.fr/fr

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 2018, three general meetings have been organized, with presentations of the partners on new results related to hybrid systems modeling and verification. A two days workshop open to a larger community of researchers and engineers has been organized, with a focus on model-based system diagnosis [8]. The programme of the workshop comprized invited talks by Erik Frisk and Mattias Krysander on the use of DAE Structural Analysis methods to generated automatically embedded diagnosers from a system model.

Two PhDs funded by the ModeliScale IPL have 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 Paris-Tech.) and Marc Pouzet (PARKAS team, Inria/ENS Paris). His PhD topic is on random testing of hybrid systems, using techniques inspired by QuickCheck [33].

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 2018, two reports have been delivered: the first one is a state of the art on structural analysis methods for DAE systems [9], while the second details a structural analysis algorithm for multimode DAE systems [10]. It is an improvement of the algorithm presented in [16].

8.3. International Initiatives

8.3.1. Informal International Partners

The Hycomes team has a continued collaboration with Martin Otter (DLR, Munich, Germany) and Hilding Elmqvist (Mogram AB, Lund, Sweden), on the structural analysis and compilation of the Modelica language [17]. The team is also establishing a collaboration with John Pryce from the University of Cardiff (UK), on the structural analysis of DAE systems.

[8] https://team.inria.fr/modeliscale/workshop-on-diagnostics-25-26-january-2018/
[9] Modeliscale project, deliverable M2.1.1 1, Structural Analysis of Differential-Algebraic Equations (DAE), State-of-the-Art.
[10] Modeliscale project, deliverable M2.1.2 1, Algorithms for the structural Analysis of Multi-Mode DAE Systems.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

Prof. Jean-Baptiste Jeannin, from the University of Michigan (Ann Arbor, Mi, USA) has visited the Hycomes team at the beginning of Summer 2018. He has collaborated with Kahlil Ghorbal and Benoît Caillaud on the topics cyber-physical systems modeling and contract-based reasoning.

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 Programme Committee of the American Modelica Conference 2018.
- Benoît Caillaud has served on the Steering and Programme Committees of the ACSD'18 conference.
- Khalil Ghorbal has served on the Programme Committee of the Japanese Modelica Conference 2018.

9.1.1.2. Reviewer

- Benoî Caillaud has reviewed papers for the following conferences : ACSD'18,
- Khalil Ghorbal has reviewed papers for the following conferences : Japanese Modelica Conference 2018,

9.1.2. Leadership within the Scientific Community

Albert Benveniste has given a lecture on the Signal synchronous language at Collège de France [9], hosted by Gérard Berry, in the realm of his chair in Computer Science.

9.1.3. Scientific Expertise

- Albert Benveniste is president of the Scientific Council of Orange and member of the Scientific Council of Safran. He has also evaluated grant proposals submitted to the European Research Council.
- Benoît Caillaud has evaluated a grant proposal submitted to the European Research Council. As an Evaluation Committee member, he has served on several Inria hiring and promotion committees (in particular, Senior Researcher at a national level and Junior Researcher in Lillle).

9.1.4. Research Administration

- Albert Benveniste is member of the Burex (Executive Bureau) of the Cominlabs Labex [11].
- Benoît Caillaud is in charge of the IPL ModeliScale [12] national initiative funded by Inria. He is also head of the Programming Languages & Software Engineering department [13] of IRISA.

[11] https://cominlabs.u-bretagneloire.fr/governance
[12] https://team.inria.fr/modeliscale/
[13] http://www.irisa.fr/en/departments/d4-language-and-software-engineering

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 [14] 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 [15] team of IRISA), started December 2017.

9.2.3. Juries

Benoît Caillaud has been external examiner of Nikolaos Kekatos' PhD, defended at the University of Grenoble Alpes in December 2018. He has also served on the jury of Etienne André's habilitation, defended in June 2018 at University Paris 13.

10. Bibliography

Major publications by the team in recent years

[1] 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

[2] 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

[3] 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° 1-2, pp. 119-149, http://dx.doi.org/10.3233/FI-2011-416

[4] 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

[14] https://team.inria.fr/tripop/
[15] https://www-semlis.irisa.fr

Publications of the year

Articles in International Peer-Reviewed Journals

[5] 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

[6] 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, https://hal.inria.fr/hal-01971429

Conferences without Proceedings

[7] A. T. SOGOKON, K. GHORBAL, Y. K. TAN, A. PLATZER. *Vector Barrier Certificates and Comparison Systems*, in "FM 2018 - 22nd International Symposium on Formal Methods", Oxford, United Kingdom, LNCS, Springer, July 2018, vol. 10951, pp. 418-437 [*DOI :* 10.1007/978-3-319-95582-7_25], https://hal.archives-ouvertes.fr/hal-01969800

Research Reports

[8] W. DEDZOE, J. HANY, A. BENVENISTE. *Competencies mining with LookinLabs*, Inria Rennes Bretagne Atlantique, March 2018, n° RR-9158, pp. 1-27, https://hal.inria.fr/hal-01739845

Other Publications

[9] A. BENVENISTE, T. GAUTIER. *The Signal synchronous language: the principles beyond the language and how to exploit and extend them*, March 2018, pp. 1-68, Lecture, https://hal.archives-ouvertes.fr/hal-01929567

References in notes

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

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

[12] 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° 94

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

[14] 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

[15] 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

[16] A. BENVENISTE, B. CAILLAUD, H. ELMQVIST, K. GHORBAL, M. OTTER, M. POUZET. *Structural Analysis of Multi-Mode DAE Systems*, in "Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, HSCC 2017", Pittsburgh, PA, United States, April 2017 [*DOI :* 10.1145/3049797.3049806], https://hal.inria.fr/hal-01521918

[17] A. BENVENISTE, B. CAILLAUD, H. ELMQVIST, K. GHORBAL, M. OTTER, M. POUZET. *Multi-Mode DAE Models: Challenges, Theory and Implementation*, in "Lecture Notes in Computer Science Celebrates 10,000th Manuscript!", Lectures Notes in Computer Science, Springer, 2019, vol. 10000, to appear

[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] 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

[21] 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

[22] 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

[23] 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

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

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

[26] G. BOUDOL, K. G. LARSEN. *Graphical Versus Logical Specifications*, in "Theor. Comput. Sci.", 1992, vol. 106, n^{o} 1, pp. 3-20

[27] 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

[28] 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

[29] 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^{o} 34, pp. 4373-4404 [*DOI :* 10.1016/J.TCS.2011.05.010], http://hal.inria.fr/hal-00654003/en

[30] 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

[31] 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

[32] 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

[33] 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

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

[35] G. CLAUDE, V. GOURANTON, B. CAILLAUD, B. GIBAUD, B. ARNALDI, P. JANNIN. *Synthesis and Simulation of Surgical Process Models*, in "Studies in Health Technology and Informatics", 2016, vol. 220, pp. 63–70 [*DOI :* 10.3233/978-1-61499-625-5-63], https://hal.archives-ouvertes.fr/hal-01300990

[36] G. CLAUDE, V. GOURANTON, B. CAILLAUD, B. GIBAUD, P. JANNIN, B. ARNALDI. *From Observations to Collaborative Simulation: Application to Surgical Training*, in "ICAT-EGVE 2016 - International Conference on Artificial Reality and Telexistence, Eurographics Symposium on Virtual Environments", Little Rock, Arkansas, United States, December 2016, https://hal.archives-ouvertes.fr/hal-01391776

[37] W. DAMM, E. THADEN, I. STIERAND, T. PEIKENKAMP, H. HUNGAR. *Using Contract-Based Component Specifications for Virtual Integration and Architecture Design*, in "Proceedings of the 2011 Design, Automation and Test in Europe (DATE'11)", March 2011

[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] 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

[44] A. LAMERCERIE. *Formal analysis of natural language requirements for the design of cyber-physical systems*, in "Conférence TALN 2018", Rennes, France, May 2018, https://hal.inria.fr/hal-01970134

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

[46] 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

[47] 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

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

[49] 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

[50] 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

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

[52] 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

[53] 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

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

[55] 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

[56] 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

[57] 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

[58] 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

[59] 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