Activity Report 2016

Project-Team HYCOMES

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

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Embedded and Real-time Systems

Table of contents

1. Members 2
2. Overall Objectives 2
3. Research Program 2
3.1. Hybrid Systems Modeling 2
3.2. Background on non-standard analysis 3
3.3. Contract-Based Design, Interfaces Theories, and Requirements Engineering 4
4. Application Domains 6
4.1.1. Modelica 6
4.1.2. Co-modeling and co-simulation 7
4.1.3. Beyond simulation 7
4.1.4. Verification 7
5. Highlights of the Year 7
6. New Software and Platforms 8
6.1. SunDAE 8
6.2. Flipflop 8
6.3. MICA 8
7. New Results 9
7.1. Structural Analysis of Multi-Mode DAEs 9
7.2. Decoupling Abstractions 9
7.3. Formal Verification of the ACAS X System 9
7.4. Chattering-Free Simulation 10
8. Partnerships and Cooperations 10
8.1. Regional Initiatives 10
8.2. European Initiatives 10
9. Dissemination 11
9.1. Promoting Scientific Activities 11
9.1.1. Scientific Events Selection 11
9.1.1.1. Member of the Conference Program Committees 11
9.1.1.2. Reviewer 11
9.1.2. Invited Talks 11
9.1.3. Research Administration 11
9.2. Teaching - Supervision - Juries 12
9.2.1. Teaching 12
9.2.2. Supervision 12
9.2.3. Juries 12
10. Bibliography 12

Project-Team HYCOMES

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

Keywords:

Computer Science and Digital Science:

2. - Software
2.1. - Programming Languages
2.1.1. - Semantics of programming languages
2.1.5. - Constraint programming
2.1.8. - Synchronous languages
2.1.10. - Domain-specific languages
2.2. - Compilation
2.3. - Embedded and cyber-physical systems
2.3.1. - Embedded systems
2.3.2. - Cyber-physical systems
2.3.3. - Real-time systems
2.4. - Verification, reliability, certification
2.4.1. - Analysis
2.4.2. - Model-checking
2.4.3. - Proofs
2.5. - Software engineering
2.5.1. - Software Architecture & Design
2.5.2. - Component-based Design
3. - Data and knowledge
3.1. - Data
3.1.1. - Modeling, representation
6. - Modeling, simulation and control
6.1. - Mathematical Modeling
6.1.1. - Continuous Modeling (PDE, ODE)
6.1.3. - Discrete Modeling (multi-agent, people centered)
6.1.5. - Multiphysics modeling

Other Research Topics and Application Domains:

2. - Health
2.4. - Therapies
2.4.3. - Surgery
4. - Energy
5. - Industry of the future
5.2. - Design and manufacturing
5.2.1. - Road vehicles
5.2.2. - Railway
5.2.3. - Aviation
5.2.4. - Aerospace

5.8. - Learning and training
5.9. - Industrial maintenance
7. - Transport and logistics
7.1. - Traffic management
7.1.3. - Air traffic
8. - Smart Cities and Territories
8.1. - Smart building/home
8.1.1. - Energy for smart buildings

1. Members

Research Scientists

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

Engineer

Aurélien Lamercerie [Inria, from Nov 2016, funded by the Sunset collaborative project, Labex CominLabs]

PhD Student

Ayman Aljarbouh [Inria, partially granted by Conseil Régional de Bretagne and by the Modrio ITEA2 project]

Administrative Assistant

Angélique Jarnoux [Inria]

Other

Siham Rim Boudaoud [Univ. Rennes I, Master Intern, from Feb 2016 until Jun 2016]

2. Overall Objectives

2.1. Overall Objectives

Hycomes has been created as a new team of the Rennes — Bretagne Atlantique Inria research center in July 2013. The team builds upon the most promising results of the former S4 team-project and of the Synchronics large scale initiative. Two topics in embedded system design are covered:

- Hybrid systems modelling, with applications to the design of multi-physics embedded systems, often referenced as cyber-physical systems;
- Contract-based design and interface theories, with applications to requirements engineering in the context of safety-critical systems design.

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 [13].

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 [21], [1] and [17].

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 [1], [21], [18], [17]. The following text provides a brief summary of this theory and gives some hints on its usefulness in the context of hybrid systems modeling. This presentation is based on our paper [1], a chapter of Simon Bliudze's PhD thesis [27], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [50].

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 [56], [42], [12]. 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.

[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.

Not surprisingly, such an idea is quite ancient. Iwasaki et al. [46] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [28], [27] 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.

The introduction to non-standard analysis in [27] is very pleasant and we take the liberty to borrow it. This presentation was originally due to Lindstrøm, see [50]. Its interest is that it does not require any fancy axiomatic material but only makes use of the axiom of choice — actually a weaker form of it. The proposed construction bears some resemblance to the construction of $\mathbb{R}$ as the set of equivalence classes of Cauchy sequences in $\mathbb{Q}$ modulo the equivalence relation $(u_n) \approx (v_n)$ iff $\lim_{n\to\infty} (u_n - v_n) = 0$.

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 [54]. 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 [2]. 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 [47], [36], [53], [15], [37]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [43]. A/G-contracts were advocated by the SPEEDS project [20]. They were further experimented in the framework of the CESAR project [38], 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 [25] and probabilistic [32], [33] viewpoints in system design, and even mixed-analog circuit design [55].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch Input/Output Automata [52], [51]. Interface Automata [59], [58], [60], [34] 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 [49], [14], [29], [48]. 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 [61], [22], [24], [40], [39], [23], probabilistic [32], [41] and energy-aware [35] 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 [57]. 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. Application Domains

4.1. Cyber-Physical Systems Design

Academic research and industry are currently witnessing several major revolutions: *Cyber-Physical Systems* (CPS), *Big-Data* and *Cloud Computing*, just to name a few. The Hycomes team is focused on CPS, and more precisely on CPS modeling with two targeted applications: The rigorous design of CPS and the optimal exploitation of CPS. Despite many engineers believe that *systems become too complex to be modeled in a faithfully*, the Hycomes team defends the opposite idea. We believe in the benefits of modeling, but acknowledge that the communities of researchers and tool developers are in part responsible for this defiance. The steep increase in the complexity of systems (e.g., public transportation systems, electric power grids) and of their models comes from composing smaller subsystems into complex architectures. As a matter of fact, these architectures are sparse, and subsystems interactions are confined to immediate surrounding neighborhoods. Thus, the dimension (number of state variables) of a system is not the most appropriate characterization of its complexity. It is rather the structure of a system and its combinatorics of modes of operation that encapsulate its complexity.

The main objective of the Hycomes team 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). We identify below the different axis we want to tackle.

4.1.1. Modelica

Modelica is a component-based modeling language initially designed for the modeling of multi-physics systems. The mathematical paradigm underlying Modelica, known as *Differential Algebraic Equations* (DAE). The key challenge is to be able to combine algebraic constraints, resulting from the laws of physics, in interaction with the nonsmooth behavior of some physical phenomena (e.g., impact laws), the multiple modes of operation of the system, and the intrinsically discrete behavior of software components. In essence, Modelica is based on the concept of multi-mode DAE, so that models can switch from one behavior to another when an event occurs, typically the crossing of a threshold. This approach is paramount to the modeling of large CPS. For instance, EDF has done a thorough modeling of the electric power grid of the Reunion island [4]. This was undertaken to gain a better understanding of this complex and notably unstable assembly of highly decentralized electric power plants: dams, small thermal power plants, wind and solar farms, and residential solar panels, just to name a few. This large model turned out to be intractable with state-of-the-art Modelica tools: because Modelica compilation techniques are not modular, the whole model has to be compiled as one unit, resulting in a very large simulation code. Parallel simulation of Modelica models is still in its infancy and gives poor results on very large models [44]; parallel/distributed techniques for networks of FMU components are not applicable to a monolithic model [45], [16]. Moreover, when simulating, for instance, thermal models of a building, the opening of a window or of a door impacts the whole simulation, despite it only has a local impact on the heat exchanges and temperatures. This is caused by the sudden change of stiffness in some part

[4] http://www.ceser-reunion.fr/fileadmin/user_upload/tx_pubdb/archives/10.10.18_Rapport_electricite.pdf

of the model, that forces a change in discretization step size (assuming that a variable step solver is used for simulation), with the adverse effect that the simulation of the whole system is slowed down. The root cause of this phenomenon boils down to the fact that system models and numerical methods used to simulate them are not space adaptive — recall that such models are 0-D models, with ODEs/DAEs, with no Partial Differential Equations (PDEs).

4.1.2. Co-modeling and co-simulation

The emergence of the FMI standard [5] supporting co-modeling and co-simulation has contributed to the widespread belief that the co-simulation of a large number of models is achievable using FMI-based tools. This is unfortunately an illusion, as FMI does not guarantee the reproducibility and determinacy of simulations. There are several reasons for that. First, FMI offers no rollback mechanism [30], which makes the co-simulation to depend on the discretization policy. Second, as the standard is not formally specified, its various implementations by tool developers differ.

4.1.3. Beyond simulation

Many physical science engineers (mechanical, electrical, aeronautic, ...) develop models with the sole objective to simulate them, while it is known that models can be used for a variety of tasks, all contributing towards the safe design and operation of a CPS: validating a design model against a set of requirements, assess the robustness of a model, test implementations against a design model, perform state estimation during system operation, just to name a few.

Early stages of CPS design usually consist in the elicitation of system-level requirements that will be used later on to design detailed models that can be simulated. Most often, the design tasks are split among several suppliers. This calls for precise requirements to be passed to them, so that, as far as feasible, suppliers can work independently. Some of the requirements specify the allowed behavior of the sub-system to be design, while others specify the assumed behavior of the sub-system's environment.

During operation of a CPS, maintenance tasks play an ever-increasing role, to minimize the downtime of the system and, to maintain an extremely low probability of occurrence of catastrophic failures. *Diagnosis* enables to replace some routine inspections or precautionary replacements of critical parts (that are usually triggered by the number of hours of operation, or by calendar) by fewer maintenance operations, triggered by the estimated wear or aging of those parts. This helps to reduce immobilization times and maintenance costs. Design models could be reused to help the development of diagnosis software that will trigger maintenance operations, based on the output of *parity check* algorithms [26], capable of detecting slow or sudden changes of some parameters. Reusing design models in this context would be a genuine innovation, in comparison to the established practice, where diagnosis is designed by hand, from scratch.

4.1.4. Verification

Because of severe complexity or undecidability problems, CPS formal verification can be done only on partial and simplified models. When applicable, these techniques complement usefully simulations. Despite of the high level of expertise it requires, formal verification brings a level of confidence in the analyses that can not be compared with what can be obtained by simulation. Using formal verification makes sense only for the most critical parts of a CPS. A fine example is the formal correctness proof of a new generation of aircraft collision prevention system, the ACAS-X [6]. This proof has facilitated the certification of this system, according to the established aeronautic standards (DO-178C [6]).

5. Highlights of the Year

5.1. Highlights of the Year

[5] https://www.fmi-standard.org/

[6] http://www.adacore.com/gnatpro-safety-critical/avionics/do178c/

Team members have made a significant step towards the definition of a formal semantics of multimode DAE systems, their strucutral analysis and the generation of simulation code. In particular, impulsive behavior at mode changes are handled correctly [19] (see Section 7.1 for full details). This semantics has been implemented, in part, in the SunDAE prototype software (Section 6.1).

6. New Software and Platforms

6.1. SunDAE

Structural analysis tool for multimode DAE systems
FUNCTIONAL DESCRIPTION

SunDAE is a multimode DAE (mDAE) structural analysis tool. Structural differentiation index is determined, impulsion analysis is performed and a BTF scheduling of the equations is performed, for each mode of a mDAE system. The input language consists in guarded equations. The output is a state-machine where states define continuous-time dynamics and transitions define resets. Both are defined by scheduled blocks of equations. SunDAE has been developed since 2016 by the Hycomes team and is distributed as an open-source software, under the CeCCIL Free Software Licensing Agreement.

- Contact: Benoit Caillaud

6.2. Flipflop

Test & Flip Net Synthesis Tool for the Inference of Technical Procedure Models
FUNCTIONAL DESCRIPTION

Flipflop is a Test and Flip net 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.

- Contact: Benoit Caillaud
- URL: http://tinyurl.com/oql6f3y

6.3. 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: Benoit Caillaud
- Contact: Benoit Caillaud
- URL: http://www.irisa.fr/s4/tools/mica/

7. New Results

7.1. Structural Analysis of Multi-Mode DAEs

Differential Algebraic Equation (DAE) systems constitute the mathematical model supporting physical modeling languages such as Modelica or Simscape. Unlike Ordinary Differential Equations, or ODEs, they exhibit subtle issues because of their implicit *latent equations* and related *differentiation index*. Multi-mode DAE (mDAE) systems are much harder to deal with, not only because of their mode-dependent dynamics, but essentially because of the events and resets occurring at mode transitions. Unfortunately, the large literature devoted to the numerical analysis of DAEs do not cover the multi-mode case. It typically says nothing about mode changes. This lack of foundations cause numerous difficulties to the existing modeling tools. Some models are well handled, others are not, with no clear boundary between the two classes. In [11], we develop a comprehensive mathematical approach to the *structural analysis* of mDAE systems which properly extends the usual analysis of DAE systems. We define a constructive semantics based on nonstandard analysis and show how to produce execution schemes in a systematic way. This work has been accepted for presentation at the HSCC 2017 conference [19] in April 2017.

7.2. Decoupling Abstractions

In [10], we investigated decoupling abstractions, by which we seek to simulate (i.e. abstract) a given system of ordinary differential equations (ODEs) by another system that features completely independent (i.e. uncoupled) sub-systems, which can be considered as separate systems in their own right. Beyond a purely mathematical interest as a tool for the qualitative analysis of ODEs, decoupling can be applied to verification problems arising in the fields of control and hybrid systems. Existing verification technology often scales poorly with dimension. Thus, reducing a verification problem to a number of independent verification problems for systems of smaller dimension may enable one to prove properties that are otherwise seen as too difficult. We show an interesting correspondence between Darboux polynomials and decoupling simulating abstractions of systems of polynomial ODEs and give a constructive procedure for automatically computing the latter.

7.3. Formal Verification of the ACAS X System

The *Next-Generation Airborne Collision Avoidance System* (ACAS X) is intended to be installed on all large aircraft to give advice to pilots and prevent mid-air collisions with other aircraft. It is currently being developed by the Federal Aviation Administration (FAA). In [6], we determine the geometric configurations under which the advice given by ACAS X is safe under a precise set of assumptions and formally verify these configurations using hybrid systems theorem proving techniques. We consider subsequent advisories and show how to adapt our formal verification to take them into account. We examine the current version of the real ACAS X system and discuss some cases where our safety theorem conflicts with the actual advisory given by that version, demonstrating how formal hybrid systems proving approaches are helping to ensure the safety of ACAS X. Our approach is general and could also be used to identify unsafe advice issued by other collision avoidance systems or confirm their safety.

7.4. Chattering-Free Simulation

Chattering is a fundamental phenomenon that is unique to hybrid systems, due to the complex interaction between discrete dynamics (in the form of discrete transitions) and continuous dynamics (in the form of time). In practice, simulating chattering hybrid systems is challenging in that simulation effectively halts near the chattering time point, as an infinite number of discrete transitions would need to be simulated. In [7], formal conditions are provided for when the simulated models of hybrid systems display chattering behavior, and methods are proposed for avoiding chattering "on the fly" in runtime. We utilize dynamical behavior analysis to derive conditions for detecting chattering without enumeration of modes. We also present a new iterative algorithm to allow for solutions to be carried past the chattering point, and we show by a prototypical implementation how to generate the equivalent chattering-free dynamics internally by the simulator in the main simulation loop.

8. Partnerships and Cooperations

8.1. Regional Initiatives

- Ayman Aljarbouh's PhD is partially funded by an ARED grant of the Brittany Regional Council. His doctoral work took place in the context of the Modrio (completed in 2016) and Sys2Soft (completed in 2015) projects on hybrid systems modeling. Ayman Aljarbouh is working on accelerated simulation techniques for hybrid systems. In particular, he is focusing on the regularisation, at runtime, of chattering behaviour and the approximation of Zeno behaviour.
- Benoît Caillaud and Aurélien Lamercerie are participating to the S3PM and 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 [31]. 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 [9], [5].

8.2. European Initiatives

8.2.1. Collaborations in European Programs, Except FP7 & H2020

Program: ITEA2

Project acronym: Modrio

Project title: Model Driven Physical Systems Operation

Duration: September 2012 – May 2016

Coordinator: EDF (France)

Other partners: ABB (Sweden), Ampère Laboratory / CNRS (France), Bielefeld University (Germany), Dassault Systèmes (Sweden), Dassault Aviation (France), DLR (Germany), DPS (France), EADS (France), Equa Simulation (Sweden), IFP (France), ITI (Germany), Ilmenau University (Germany), Katholic University of Leuven (Belgium), Knorr-Bremse (Germany), LMS (France and Belgium), Linköping University (Sweden), MathCore (Sweden), Modelon (Sweden), Pöry (Finland), Qtronic (Germany), SICS (Sweden), Scania (Sweden), Semantum (Finland), Sherpa Engineering (France), Siemens (Germany and Sweden), Simpack (Germany), SKF (Sweden), Supmeca (France), Triphase (Belgium), University of Calabria (Italy), VTT (Finland), Vattenfall (Sweden), Wapice (Finland).

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

Abstract: Modelling and simulation are efficient and widely used tools for system design. But they are seldom used for systems operation. However, most functionalities for system design are beneficial for system operation, provided that they are enhanced to deal with real operating situations. Through open standards the benefits of sharing compatible information and data become obvious: improved cooperation between the design and the operation communities, easier adaptation of operation procedures wrt. design evolutions. Open standards also foster general purpose technology. The objective of the ITEA 2 MODRIO project is to extend modelling and simulation tools based on open standards from system design to system operation.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events Selection

9.1.1.1. Member of the Conference Program Committees

Benoî Caillaud has served on the program committee of ACSD 2016 (http://acsd2016.mat.umk.pl), a conference on the applications of concurrency in system design. He is a member of the steering committe of ACSD since 2006.

9.1.1.2. Reviewer

Benoît Caillaud has reviewed papers submitted to the ACSD 2016 and ACC 2016 conferences.

Khalil Ghorbal reviewed two regular research papers for the Hybrid Systems: Computation and Control Conference.

Khalil Ghorbal reviewed two journal papers for the IEEE Transactions on Automatic Control.

Khalil Ghorbal reviewed a journal paper for the Computer Journal (Oxford Journals, Science and Mathematics).

Khalil Ghorbal reviewed a journal paper for the Information and Computation journal (Elsevier).

9.1.2. Invited Talks

Benoît Caillaud has given an invited talk on *Time Domains in Hybrid Systems Modeling* at the SHARC 2016 workshop and ALROB meetingthat took place in Brest in June 2016 (http://lab-sticc.univ-brest.fr/~goulven/sharc2016/program/index.html).

In May 13, 2016, Khalil Ghorbal gave an invited talk about the invariant generation for polynomial ordinary differential equations during the Effective Algebraic Geometry Seminar, IRMAR, Rennes, France.

In May 23, 2016, Ayman Aljarbouh presented a talk at the Embassy of Sweden in Tokyo for the first Japanese Modelica Conference (MODELICA2016), May 23-24, 2016, Tokyo, JAPAN.

In July 2016, Ayman Aljarbouh presented a poster during the French-American Doctoral Exchange Seminar (FADEx) 2016: Systèmes Cyber-Physiques, July 04-08, 2016, Grenoble, FRANCE.

In November, 5-12, Albert Benveniste was invited at the Systems Research Center, a center of excellence of the University of Maryland at College Park, USA.

9.1.3. Research Administration

Benoît Caillaud is head of the *Languages and Software Engineering* department of IRISA (http://www.irisa.fr/en/departments/d4-language-and-software-engineering). He has been in charge of presenting the departement during the evaluation seminar of IRISA by HCERES in January 2016.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master : Benoît Caillaud is teaching with Marc Pouzet a first year master degree course on *hybrid systems modeling*. The course is open to the students registered to the computer science research and innovation curriculum of the university of Rennes 1 and ENS Rennes, France.

Master : Khalil Ghorbal was "Chargé de TD" (20h Eq TD) for the "Analyse et Conception Formelles" module open for students registered to the computer science master degree of the university of Rennes 1 and ENS Rennes, France.

9.2.2. Supervision

PhD in progress : Ayman Aljarbouh, *Accelerated Simulation of Hybrid Systems*, started january 2014, supervised by Benoît Caillaud. Ayman Aljarbouh is expected to defend his PhD in MArch 2017.

9.2.3. Juries

Khalil Ghorbal was reviewer in the PhD defense committee of Sameh Mohamed, "Une Méthode Topologique pour la Recherche d'Ensembles Invariants de Systèmes Continus et á Commutation", defended in October 17th, 2016, Univ. Paris Saclay (ENS Cachan).

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^{o} 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] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. L. Sangiovanni-Vincentelli, W. Damm, T. A. Henzinger, K. G. Larsen. *Contracts for System Design*, Inria, November 2012, n^{o} RR-8147, 65 p. , http://hal.inria.fr/hal-00757488

[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^{o} 1-2, pp. 119-149, http://dx.doi.org/10.3233/FI-2011-416

Publications of the year

Articles in International Peer-Reviewed Journals

[4] G. Baudart, A. Benveniste, T. Bourke. *Loosely Time-Triggered Architectures*, in "ACM Transactions on Embedded Computing Systems (TECS)", August 2016, vol. 15, Article 71 [*DOI :* 10.1145/2932189], https://hal.inria.fr/hal-01408224

[5] 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

[6] 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", October 2016 [*DOI :* 10.1007/S10009-016-0434-1], https://hal.archives-ouvertes.fr/hal-01232365

Invited Conferences

[7] A. ALJARBOUH, B. CAILLAUD. *Chattering-Free Simulation of Hybrid Dynamical Systems with the Functional Mock-Up Interface 2.0*, in "The First Japanese Modelica Conferences", Tokyo, Japan, Linköping University Electronic Press, Linköpings universitet, May 2016, vol. 124, n^{o} 013, pp. 95-105 [*DOI :* 10.3384/ECP1612495], https://hal.archives-ouvertes.fr/hal-01247008

[8] A. ALJARBOUH, A. DURACZ, Y. ZENG, B. CAILLAUD, W. TAHA. *Chattering-Free Simulation for Hybrid Dynamical Systems: Semantics and Prototype Implementation*, in "2016 IEEE International Conference on Computational Science and Engineering, IEEE International Conference on Embedded and Ubiquitous Computing, and International Symposium on Distributed Computing and Applications to Business, Engineering and Science", Paris, France, Proceedings of the 19th IEEE International Conference on Computational Science and Engineering, the 14th IEEE International Conference on Embedded and Ubiquitous Computing, the 15th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, IEEE Computer Society, August 2016 [*DOI :* 10.1109/CSE-EUC-DCABES.2016.217], https://hal.archives-ouvertes.fr/hal-01365875

International Conferences with Proceedings

[9] 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

[10] A. T. SOGOKON, K. GHORBAL, T. T. JOHNSON. *Decoupling Abstractions of Non-linear Ordinary Differential Equations*, in "Formal Methods", Limassol, Cyprus, FM 2016: Formal Methods, November 2016, vol. Lecture Notes in Computer Science, n^{o} 9995, pp. 628-644 [*DOI :* 10.1007/978-3-319-48989-6_38], https://hal.archives-ouvertes.fr/hal-01374899

Research Reports

[11] A. BENVENISTE, B. CAILLAUD, M. POUZET, H. ELMQVIST, M. OTTER. *Structural Analysis of Multi-Mode DAE Systems*, Inria, July 2016, n^{o} RR-8933, 32 p. , https://hal.inria.fr/hal-01343967

References in notes

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

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

[14] 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

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

[16] A. Ben Khaled, M. E. M. Ben Gaïd, N. Pernet, D. Simon. *Fast multi-core co-simulation of Cyber-Physical Systems : application to internal combustion engines*, in "Simulation Modelling Practice and Theory", September 2014, vol. 47, n^{o} September, pp. 79-91 [*DOI :* 10.1016/j.simpat.2014.05.002], https://hal-ifp.archives-ouvertes.fr/hal-01018348

[17] 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

[18] 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

[19] A. Benveniste, B. Caillaud, H. Elmqvist, K. Ghorbal, M. Otter, M. Pouzet. *Structural Analysis of Multi-Mode DAE Systems*, in "Proc. of the 20th ACM International Conference on Hybrid Systems: Computation and Control, HSCC'17", Pittsburgh, PA, USA, April 2017, to appear

[20] 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

[21] 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

[22] 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

[23] 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

[24] 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

[25] 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

[26] G. Biswas, A. Bregon, X. Koutsoukos, B. Pulido. *Analytic Redundancy, Possible Conflicts, and TCG-based Fault Signature Diagnosis applied to Nonlinear Dynamic Systems*, in "IFAC Proceedings Vol-

umes", 2009, vol. 42, n^{o} 8, pp. 1486 - 1491 [*DOI :* 10.3182/20090630-4-ES-2003.00242], http://www.sciencedirect.com/science/article/pii/S1474667016359857

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

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

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

[30] D. BROMAN, L. GREENBERG, E. A. LEE, M. MASIN, S. TRIPAKIS, M. WETTER. *Requirements for hybrid cosimulation standards*, in "Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control, HSCC'15, Seattle, WA, USA, April 14-16, 2015", A. GIRARD, S. SANKARANARAYANAN (editors), ACM, 2015, pp. 179–188, http://doi.acm.org/10.1145/2728606.2728629

[31] 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

[32] 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

[33] 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

[34] 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

[35] 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

[36] 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

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

[38] 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

[39] 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

[40] 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

[41] 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

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

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

[44] H. ELMQVIST, S. E. MATSSON, H. OLSSON. *Parallel Model Execution on Many Cores*, in "Proceedings of the 10th International Modelica Conference; March 10-12; 2014; Lund; Sweden", Linköping University Electronic Press; Linköpings universitet, 2014, pp. 363-370

[45] V. GALTIER, S. VIALLE, C. DAD, J.-P. TAVELLA, J.-P. LAM-YEE-MUI, G. PLESSIS. *FMI-Based Distributed Multi-Simulation with DACCOSIM*, in "Symposium on Theory of Modeling and Simulation - TMS'15", Alexandria, VA, United States, April 2015, pp. 804-811, https://hal-supelec.archives-ouvertes.fr/hal-01155707

[46] 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

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

[48] 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

[49] 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

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

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

[52] 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

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

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

[55] 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$^{\rm o}$ 12, pp. 3329–3345

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

[57] 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

[58] 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

[59] 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

[60] 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

[61] 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