IN PARTNERSHIP WITH:
Université Rennes 1

Activity Report 2017

Project-Team HYCOMES

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

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

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Embedded and Real-time Systems

Table of contents

Project-Team HYCOMES

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

Keywords:

Computer Science and Digital Science:

A2. - Software
A2.1. - Programming Languages
A2.1.1. - Semantics of programming languages
A2.1.5. - Constraint programming
A2.1.8. - 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. - 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. - 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. Personnel

Research Scientists

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

PhD Students

Ayman Aljarbouh [Inria, graduated Sep 2017, currently postdoctoral researcher at Verimag. Grenoble, France]
Aurélien Lamercerie [Univ de Rennes I, from Nov 2017]

Technical staff

Jean Hany [Inria, from Dec 2017]
Aurélien Lamercerie [Inria, until Oct 2017]

Intern

Youssouf Roudani [Univ de Rennes I, from Jul 2017 until Aug 2017]

Administrative Assistants

Angélique Jarnoux [Inria]
Armelle Mozziconacci [CNRS]

2. Overall Objectives

2.1. Overall Objectives

Hycomes is a team of the Rennes — Bretagne Atlantique Inria research center since 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 cyber-physical systems design are covered:

- 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.

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 [20], [1] and [16].

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

3.2. Background on non-standard analysis

Non-Standard analysis plays a central role in our research on hybrid systems modeling [1], [20], [17], [16]. 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 [25], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [47].

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

[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.

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 [53], [41], [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.

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 [26], [25] have also used non-standard analysis as a mathematical support for defining a system theory for hybrid systems. They discuss in detail the notion of “system” and investigate computability issues. The formalization they propose closely follows that of Turing machines, with a memory tape and a control mechanism.

3.3. 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 [51]. 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 [44], [33], [50], [15], [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 [19]. 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 [24] and probabilistic [29], [30] viewpoints in system design, and even mixed-analog circuit design [52].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch Input/Output Automata [49], [48]. Interface Automata [56], [55], [57], [31] 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 [46], [14], [27], [45]. 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 [58], [21], [23], [39], [38], [22], probabilistic [29], [40] and energy-aware [32] 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 [54]. 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. New Software and Platforms

4.1. 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://tinyurl.com/oql6f3y

4.2. 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/

4.3. 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

5. New Results

5.1. Semantics, Static or Runtime Analysis of Hybrid Systems

5.1.1. 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 [18] in April 2017.

5.1.2. Operational Models for Piecewise-Smooth Systems

In [7], we study ways of constructing meaningful operational models of piecewise-smooth systems (PWS). The systems we consider are described by polynomial vector fields defined on non-overlapping semi-algebraic sets, which form a partition of the state space. Our approach is to give meaning to motion in systems of this type by automatically synthesizing operational models in the form of hybrid automata (HA). Despite appearances, it is in practice often difficult to arrive at satisfactory HA models of PWS. The different ways of building operational models that we explore in our approach can be thought of as defining different semantics for the underlying PWS. These differences have a number of interesting nuances related to phenomena such as chattering, non-determinism, so-called mythical modes and sliding behaviour.

5.1.3. *Accelerated Simulation of Hybrid Systems: Method combining static analysis and runtime execution analysis*

Ayman Aljarbouh has defended his PhD [4] on September 13th 2017. His PhD has been 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 has been working on accelerated simulation techniques for hybrid systems. In particular, he has contributed, and implemented in a software prototype, a regularisation method transforming automatically at runtime a chattering behaviour into a semantics preserving smooth behaviour. He has also contributed a method for the approximation of Zeno behaviour. This method enables to jump past an accumulation of an infinite number of zero-crossing events, and to continue the simulation of a large class of Zeno hybrid systems, after accumulation points.

5.1.4. *A Type-based Analysis of Causality Loops in Hybrid Systems Modelers*

Explicit hybrid systems modelers like Simulink/Stateflow allow for programming both discrete- and continuous-time behaviors with complex interactions between them. A key issue in their compilation is the static detection of algebraic or causality loops. Such loops can cause simulations to deadlock and prevent the generation of statically scheduled code. In [5], we addresses this issue for a hybrid modeling language that combines synchronous data-flow equations with Ordinary Differential Equations (ODEs). We introduce the operator last(x) for the left-limit of a signal x. This operator is used to break causality loops and permits a uniform treatment of discrete and continuous state variables. The semantics relies on non-standard analysis, defining an execution as a sequence of infinitesimally small steps. A signal is deemed causally correct when it can be computed sequentially and only changes infinitesimally outside of announced discrete events like zero-crossings. The causality analysis takes the form of a type system that expresses dependences between signals. In well-typed programs, signals are provably continuous during integration provided that imported external functions are also continuous. The effectiveness of this system is illustrated with several examples written in Zélus, a Lustre-like synchronous language extended with hierarchical automata and ODEs.

5.2. Formal Verification of Hybrid Systems

5.2.1. *Formal Verification of Station Keeping Maneuvers for a Planar Autonomous Hybrid System*

In [9], we investigate the formal verification of a hybrid control law designed to perform a station keeping maneuver for a planar vehicle. Such maneuver requires that the vehicle reaches a neighborhood of its station in finite time and remains in it while waiting for further commands. We model the dynamics as well as the control law as a hybrid program and formally verify the reachability and safety properties involved. We highlight in particular the automated generation of invariant regions which turns out to be crucial in performing such verification. We use the hybrid system theorem prover KeymaeraX to formally check the parts of the proof that can be automatized in the current state of the tool.

5.2.2. *Formal verification of obstacle avoidance and navigation of ground robots*

In [6], we answer fundamental safety questions for ground robot navigation: Under which circumstances does a given control decision make a ground robot safely avoid obstacles? Unsurprisingly, the answer depends on the exact formulation of the safety objective as well as the physical capabilities and limitations of the robot and the obstacles. Because uncertainties about the exact future behavior of a robot's environment make this a challenging problem, we formally verify corresponding controllers and provide rigorous safety proofs justifying why they can never collide with the obstacle in the respective physical model. To account for ground robots in which different physical phenomena are important, we analyze a series of increasingly strong properties of controllers for increasingly rich dynamics and identify the impact that the additional model parameters have on the required safety margins. We analyze and formally verify: (i) static safety, which ensures that no collisions can happen with stationary obstacles, (ii) passive safety, which ensures that no collisions can happen with stationary or moving obstacles while the robot moves, (iii) the stronger passive friendly safety in which the robot further maintains sufficient maneuvering distance for obstacles to avoid collision as well, and (iv) passive orientation safety, which allows for imperfect sensor coverage of the robot, i. e., the robot is aware that not everything in its environment will be visible. We formally prove that safety can be guaranteed despite sensor uncertainty and actuator perturbation. We complement these provably correct safety properties with liveness properties: we prove that provably safe motion is flexible enough to let the robot navigate waypoints and pass intersections. In order to account for the mixed influence of discrete control decisions and the continuous physical motion of the ground robot, we develop corresponding hybrid system models and use differential dynamic logic theorem proving techniques to formally verify their correctness. Since these models identify a broad range of conditions under which control decisions are provably safe, our results apply to any control algorithm for ground robots with the same dynamics. As a demonstration, we, thus, also synthesize provably correct runtime monitor conditions that check the compliance of any control algorithm with the verified control decisions.

5.3. Synchronous Interfaces and Assume/Guarantee Contracts

In [10], we establish a link between the theory of Moore Interfaces proposed in 2002 by Chakraborty et al. as a specification framework for synchronous transition systems, and the Assume/Guarantee contracts as proposed in 2007 by Benveniste et al. as a simple and flexible contract framework. As our main result we show that the operation of saturation of A/G contracts (namely the mapping $(A, G) \to (A, G \vee \neg A)$), which was considered a drawback of this theory, is indeed implemented by the Moore Game of Chakraborty et al. We further develop this link and come up with some remarks on Moore Interfaces.

5.4. CominWeb project of the Labex CominLabs

Jean Hany and Albert Benveniste (together with William Dedzoe) were involved in this project.

CominWeb is a project supported by the Labex CominLabs since 2013. Its original objective was to equip CominLabs with Web infrastructures, tools, and services, that would allow to run the scientific activity of the Labex in an innovative way. Based on a study of the population of the CominLabs researchers, performed in year 2014-15 by the teams of CominLabs involved in social sciences, several services were investigated and prototyped. A short trial addressed the automatic generation of a scientific activity report, for a CominLabs project, from the material available from the publications ot the project team. This was suspended because such a service was not considered very useful by the community. A second trial (nicknamed “NSA”) consisted in monitoring the flows of email exchanges addressed to aliases of the CominLabs projects, with the objective of classifying the mails into: meeting announcements, mails with attachments of interest, and other mails. This would give to the CominLabs head a view on the project’s activities without asking for any specific contribution from the researchers. This was more interesting. Still, a difficulty was that researchers did not use the project aliases so much. For priority issues, this development was also suspended.

The main result of this project is thus the service called *LookinLabs*, deployed in two different versions: http://lookinlabs4halinria.cominlabs.ueb.eu/ and http://www.lookinlabs.cominlabs.ueb.eu/. The former is a more advanced version of LookinLabs, developed for the whole Inria community, by exploiting the HAL publication archive. LookinLabs for HAL-Inria allows the user to find, among teams/individuals/publications taken from all the Inria teams, those best matching a query consisting of a list of keywords or a short text. The tool exploits, as data, HAL-Inria archives, in combination with the Inria Activity reports (the Raweb), and the internal data base of Inria teams called BASTRI. Active teams/individuals are shown in boldface. Teams/individuals shown in gray are no longer active at Inria. If team TEAM0 is no longer active, the mention: TEAM0 → (TEAM1,TEAM2) indicates follow-up active teams, if any. In LookinLabs, no ontology is used. No data need to be manually entered (besides the users’ queries). The tool uses *Elasticsearch* (https://www.elastic.co/fr/products/elasticsearch) as its core algorithm. This means that the matching is based on a distance between the query and the set of data attached, in HAL, to each team/individual/publication. Ranking is performed accordingly. Explanations are given for each returned item. Correlation graphs are given, allowing to navigate through teams or individuals that share common interests (they may or may not be co-authors).

LookinLabs is deployed in two versions. LookinLabs4HALInria is the one we just described. The other version is in operation since 2016 and addresses the scientific community of CominLabs researchers. The data used are up to 10 standard bibliographical data bases (Dblp, IEEE Explore, Arxiv, HAL, and more) for which links have been collected from the researchers (this was the only data they were asked for). Results are returned in the form of individuals and publications, not teams.

6. Bilateral Contracts and Grants with Industry

6.1. GLOSE

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. Control engineers favor Matlab/Simulink, mainly because of its toolboxes and ease of use. Computer scientists program or model real-time reactive software, either with a dedicated language, for instance SCADE, hierarchical state machines or sequence/activity diagrams (as in UML/SysML) or directly in C. 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?

The objective of the GLOSE project is to address these objectives, and propose both sound foundations and practical technological solutions to system level modeling and simulation. The GLOSE project has started in December 2017 and is funded by Safran, in the realm of the DESIR joint Safran-Academia research network. The academic teams contributing to GLOSE are the Hycomes, Diverse and Kairos Inria teams, and IRIT/CNRS in Toulouse.

7. Partnerships and Cooperations

7.1. Regional Initiatives

- Ayman Aljarbouh's PhD (see Section 5.1.3) was 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 [4]. This project focuses on the computation of surgical procedural knowledge models from recordings of individual procedures, and their execution [28]. The objective is to develop an enabling technology for procedural knowledge based computer assistance of surgery. In this project, we demonstrate its potential added value in nurse and surgeon training [36], [35]. In 2017, Benoît Caillaud and Aurélien Lamercerie have released Demodocos, a software synthesizing surgical process models from instances of surgical procedures.

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

7.2. National Initiatives

7.2.1. 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	Bipop	Inria Grenoble Rhône Alpes
Albert Benveniste Benoît Caillaud Khalil Ghorbal	Hycomes	Inria Rennes Bretagne Atlantique
Marc Pouzet Tim Bourke	Parkas	ENS Inria Paris
Goran Frehse	Tempo	Verimag-univ. Grenoble Alpes
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).

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.

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific Events Organisation

8.1.1.1. General Chair, Scientific Chair

Benoît Caillaud has organized the Synchron'17 open workshop on *Synchronous Programming Languages* [5] that took place at Inria Rennes from November 27th-30th 2017.

8.1.2. Scientific Events Selection

8.1.2.1. Member of the Conference Program Committees

Khalil Ghorbal served as a PC member in the Repeatability Evaluation Committee of HSCC (Hybrid Systems: Computation and Control) 2017.

Albert Benveniste served as a PC member of the International Modelica Conference 2017.

Benoît Caillaud has served on the Steering and Programme Committees of the ACSD'17 conference.

[5] https://synchron17.inria.fr

8.1.2.2. Reviewer

Khalil Ghorbal reviewed a paper for the IEEE Conference on Decision and Control 2017.

Albert Benveniste reviewed a paper for FoSSaCS (International Conference on Foundations of Software Science and Computation Structures) 2017.

Benoît Caillaud has reviewed one paper for the LICS'17 conference.

8.1.3. Journal

8.1.3.1. Reviewer - Reviewing Activities

Khalil Ghorbal reviewed a journal paper for the IEEE Transactions on Automatic Control.

Albert Benveniste reviewed a journal paper for the Science of Computer Programming journal.

Benoît Caillaud has reviewed papers for th IEEE Transactions on Control Systems Technology.

8.1.4. Invited Talks

Khalil Ghorbal was invited by Saman Zonouz. Rutgers University, NJ, USA.

Albert Benveniste gave an invited talk at the Laboratory for Information & Decision Systems, MIT, Cambridge, MA, USA.

8.1.5. Scientific Expertise

Albert Benveniste was a reviewer for the ERC Advanced Grant proposals 2017.

8.1.6. Research Administration

Benoît Caillaud is head of the *Language and Software Engineering Department* of IRISA (UMR 6074). The department is composed of 9 research teams and about 120 researchers and students, in Brest, Rennes and Vannes.

8.2. Teaching - Supervision - Juries

8.2.1. 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, *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

8.2.2. Supervision

PhD : Ayman Aljarbouh, *Accelerated Simulation of Hybrid Systems: Method combining static analysis and runtime execution analysis*, University of Rennes 1, defended 13/09/2017, supervised by Benoît Caillaud.

PhD : Guillaume Baudart, *A Synchronous Approach to Quasi-Periodic Systems*, Ecole Normale Superieure (Paris), defended 13/03/2017, co-supervised by Albert Benveniste.

8.2.3. Juries

Benoît Caillaud has been president of PhD defence jury of Mohamed Amine Aouadhi, on 29 September 2017, at LS2N, the University of Nantes, France.

Albert Benveniste participated in the jury of the PhD thesis of Guillaume Baudart.

9. 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

Doctoral Dissertations and Habilitation Theses

[4] A. ALJARBOUH. *Accelerated Simulation of Hybrid Systems : Method combining static analysis and run-time execution analysis*, Université de Rennes 1, France, September 2017, https://hal.inria.fr/tel-01614081

Articles in International Peer-Reviewed Journals

[5] A. Benveniste, T. Bourke, B. Caillaud, B. Pagano, M. Pouzet. *A Type-based Analysis of Causality Loops in Hybrid Systems Modelers*, in "Nonlinear Analysis: Hybrid Systems", November 2017, vol. 26, pp. 168–189 [*DOI :* 10.1016/j.nahs.2017.04.004], https://hal.inria.fr/hal-01549183

[6] S. Mitsch, K. Ghorbal, D. Vogelbacher, A. Platzer. *Formal verification of obstacle avoidance and navigation of ground robots*, in "International Journal of Robotics Research", 2017, vol. 36, n^{o} 12, pp. 1312–1340 [*DOI :* 10.1177/0278364917733549], https://hal.inria.fr/hal-01658197

[7] 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^{o} 5s, pp. 185:1–185:19 [*DOI :* 10.1145/3126506], https://hal.inria.fr/hal-01658196

International Conferences with Proceedings

[8] 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

[9] B. Martin, K. Ghorbal, E. Goubault, S. Putot. *Formal Verification of Station Keeping Maneuvers for a Planar Autonomous Hybrid System*, in "FVAV 2017 - 1st Formal Verification of Autonomous Vehicles Workshop", Turin, Italy, L. Bulwahn, M. Kamali, S. Linker (editors), FVAV@iFM 2017, September 2017, vol. 257, pp. 91–104 [*DOI :* 10.4204/EPTCS.257.9], https://hal.archives-ouvertes.fr/hal-01657848

Scientific Books (or Scientific Book chapters)

[10] A. Benveniste, B. Caillaud. *Synchronous Interfaces and Assume/Guarantee Contracts*, in "Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday", L. Aceto, G. Bacci, G. Bacci, A. Ingólfsdóttir, R. Mardare (editors), Theoretical Computer Science and General Issues, Springer, July 2017, vol. 10460, pp. 233-248 [*DOI :* 10.1007/978-3-319-63121-9_12], https://hal.inria.fr/hal-01616369

Research Reports

[11] A. Benveniste, B. Caillaud, H. Elmqvist, K. Ghorbal, M. Otter, M. Pouzet. *Structural Analysis of Multi-Mode DAE Systems*, Inria, 2017, n^{o} RR-8933, pp. 1-23, 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. Benveniste, T. Bourke, B. Caillaud, B. Pagano, M. Pouzet. *A Type-Based Analysis of Causality Loops In Hybrid Systems Modelers*, December 2013, Deliverable D3.1_1 v 1.0 of the Sys2soft collaborative project ”Physics Aware Software”, https://hal.inria.fr/hal-00938866

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

[18] A. Benveniste, B. Caillaud, 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

[19] 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

[20] 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

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

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

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

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

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

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

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

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

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

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

[31] 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

[32] 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

[33] 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

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

[45] 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

[46] 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

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

[48] 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

[49] 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

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

[51] 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

[52] 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

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

[54] 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

[55] 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

[56] 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

[57] 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

[58] 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