
Activity Report 2013

# Team Hycomes

## Hybrid Modeling and Contract-Based Design for Multiphysics Embedded Systems

RESEARCH CENTER
**Rennes - Bretagne-Atlantique**

THEME
**Embedded and Real-time Systems**

# Table of contents

# Team Hycomes

**Keywords:** Hybrid Systems, Embedded Systems Design, Physical Modeling, Contract-based Design, Requirements Engineering

*Creation of the Team:* 2013 July 01.

# 1. Members

**Research Scientists**

Benoît Caillaud [Team leader, Inria, Researcher, HdR]
Albert Benveniste [Inria, Senior Researcher, HdR]

**PhD Student**

Ayman Aljarbouh [Inria, from Dec 2013, funded by the ITEA2 Modrio project and the Brittany Regional Council]

# 2. Overall Objectives

## 2.1. Highlights of the Year

Albert Benveniste has been elected IFAC Fellow [1] for his fundamental contributions to stochastic systems theory, and for connecting control, signal processing, and real-time software development.

# 3. Research Program

## 3.1. Introduction

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 S4 [2] team-project and of the SYNCHRONICS [3] large scale initiative. Two topics in embedded system design are covered: Hybrid modeling and contract-based design.

## 3.2. 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.

[1] http://www.ifac-control.org/awards/ifac-fellows
[2] http://www.inria.fr/equipes/s4
[3] http://synchronics.wiki.irisa.fr/

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 [4]. 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 [5]. A wider set of tools, both industrial and academic, now exists in this segment [6]. In the EDA sector, VHDL-AMS was developed as a standard [12].

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 [6], [9], [15].

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.

## 3.3. 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.

[4] http://www.lccc.lth.se/media/LCCC2012/WorkshopSeptember/slides/Astrom.pdf
[5] https://www.modelica.org/
[6] SimScape by Mathworks, Amesim by LMS International, now Siemens PLM, and more.

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 [44]. 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 [4]. 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 [37], [28], [43], [14], [29]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [35]. A/G-contracts were advocated by the SPEEDS project [16]. They were further experimented in the framework of the CESAR project [30], 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 [20] and probabilistic [24], [25] viewpoints in system design, and even mixed-analog circuit design [45].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch Input/Output Automata [42], [41]. Interface Automata [49], [48], [50], [26] 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 [5] inherit from both Interface Automata and the originally unrelated notion of Modal Transition System [39], [13], [23], [38]. 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 [51], [17],

[19], [32], [31], [18], probabilistic [24], [33] and energy-aware [27] 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 [47]. DOORS files 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 subsystem.

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.

## 3.4. Background on non-standard analysis

Non-Standard analysis plays a central role in our research on hybrid systems modeling [3], [9], [10], [6]. 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 [3], a chapter of Simon Bliudze's PhD thesis [21], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [40].

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 $\partial$ is an *infinitesimal* and ${}^*\mathbb{N}$ is the set of *non-standard integers* is such 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 [46], [34], [11]. Robinson's approach is axiomatic; he proposes adding three new axioms to the basic Zermelo-Fraenkel (ZFC) framework. There has been much debate in the mathematical community as to whether it is worth considering non-standard analysis instead of staying with the traditional one. We do not enter this debate. The important thing for us is that non-standard analysis allows the use of the non-standard discretization of continuous dynamics "as if" it was operational.

Not surprisingly, such an idea is quite ancient. Iwasaki et al. [36] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [22], [21] 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 [21] is very pleasant and we take the liberty to borrow it. This presentation was originally due to Lindstrøm, see [40]. 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 $\lim_{n\to\infty}(u_n - v_n) = 0$.

### *3.4.1. Motivation and intuitive introduction*

We begin with an intuitive introduction to the construction of the non-standard reals. The goal is to augment $\mathbb{R} \cup \{\pm\infty\}$ by adding, to each $x$ in the set, a set of elements that are "infinitesimally close" to it. We will call the resulting set ${}^*\mathbb{R}$. Another requirement is that all operations and relations defined on $\mathbb{R}$ should extend to ${}^*\mathbb{R}$.

A first idea is to represent such additional numbers as convergent sequences of reals. For example, elements infinitesimally close to the real number zero are the sequences $u_n = 1/n$, $v_n = 1/\sqrt{n}$ and $w_n = 1/n^2$. Observe that the above three sequences can be ordered: $v_n > u_n > w_n > 0$ where 0 denotes the constant zero sequence. Of course, infinitely large elements (close to $+\infty$) can also be considered, e.g., sequences $x_u = n$, $y_n = \sqrt{n}$, and $z_n = n^2$.

Unfortunately, this way of defining ${}^*\mathbb{R}$ does not yield a total order since two sequences converging to zero cannot always be compared: if $u_n$ and $u'_n$ are two such sequences, the three sets $\{n \mid u_n > u'_n\}$, $\{n \mid u_n = u'_n\}$, and $\{n \mid u_n < u'_n\}$ may even all be infinitely large. The beautiful idea of Lindstrøm is to enforce that *exactly one of the above sets is important and the other two can be neglected*. This is achieved by fixing once and for all a finitely additive positive measure $\mu$ over the set $\mathbb{N}$ of integers with the following properties: [7]

1. $\mu : 2^{\mathbb{N}} \to \{0, 1\}$;
2. $\mu(X) = 0$ whenever $X$ is finite;
3. $\mu(\mathbb{N}) = 1$.

Now, once $\mu$ is fixed, one can compare any two sequences: for the above case, exactly one of the three sets must have $\mu$-measure 1 and the others must have $\mu$-measure 0. Thus, say that $u > u', u = u'$, or $u < u'$, if $\mu(\{n \mid u_n > u'_n\} = 1)$, $\mu(\{n \mid u_n = u'_n\}) = 1$, or $\mu(\{n \mid u_n < u'_n\}) = 1$, respectively. Indeed, the same trick works for many other relations and operations on non-standard real numbers, as we shall see. We now proceed with a more formal presentation.

### *3.4.2. Construction of non-standard domains*

For $I$ an arbitrary set, a *filter* $\mathcal{F}$ over $I$ is a family of subsets of $I$ such that:

1. the empty set does not belong to $\mathcal{F}$,
2. $P, Q \in \mathcal{F}$ implies $P \cap Q \in \mathcal{F}$, and
3. $P \in \mathcal{F}$ and $P \subset Q \subseteq I$ implies $Q \in \mathcal{F}$.

Consequently, $\mathcal{F}$ cannot contain both a set $P$ and its complement $P^c$. A filter that contains one of the two for any subset $P \subseteq I$ is called an *ultra-filter*. At this point we recall Zorn's lemma, known to be equivalent to the axiom of choice:

**Lemma 1 (Zorn's lemma)** *Any partially ordered set $(X, \leq)$ such that any chain in $X$ possesses an upper bound has a maximal element.*

A filter $\mathcal{F}$ over $I$ is an ultra-filter if and only if it is maximal with respect to set inclusion. By Zorn's lemma, any filter $\mathcal{F}$ over $I$ can be extended to an ultra-filter over $I$. Now, if $I$ is infinite, the family of sets $\mathcal{F} = \{P \subseteq I \mid P^c \text{ is finite}\}$ is a *free* filter, meaning it contains no finite set. It can thus be extended to a free ultra-filter over $I$:

**Lemma 2** *Any infinite set has a free ultra-filter.*

[7] The existence of such a measure is non trivial and is explained later.

Every free ultra-filter $\mathcal{F}$ over $I$ uniquely defines, by setting $\mu(P) = 1$ if $P \in \mathcal{F}$ and otherwise 0, a finitely additive measure [8] $\mu : 2^I \mapsto \{0, 1\}$, which satisfies

$$\mu(I) = 1 \text{ and, if } P \text{ is finite, then } \mu(P) = 0.$$

Now, fix an infinite set $I$ and a finitely additive measure $\mu$ over $I$ as above. Let $\mathbb{X}$ be a set and consider the Cartesian product $\mathbb{X}^I = (x_i)_{i \in I}$. Define $(x_i) \approx (x'_i)$ iff $\mu\{i \in I \mid x_i \neq x'_i\} = 0$. Relation $\approx$ is an equivalence relation whose equivalence classes are denoted by $[x_i]$ and we define

$$^*\mathbb{X} = \mathbb{X}^I / \approx \tag{1}$$

$\mathbb{X}$ is naturally embedded into $^*\mathbb{X}$ by mapping every $x \in \mathbb{X}$ to the constant tuple such that $x_i = x$ for every $i \in I$. Any algebraic structure over $\mathbb{X}$ (group, ring, field) carries over to $^*\mathbb{X}$ by almost point-wise extension. In particular, if $[x_i] \neq 0$, meaning that $\mu\{i \mid x_i = 0\} = 0$ we can define its inverse $[x_i]^{-1}$ by taking $y_i = x_i^{-1}$ if $x_i \neq 0$ and $y_i = 0$ otherwise. This construction yields $\mu\{i \mid y_i x_i = 1\} = 1$, whence $[y_i][x_i] = 1$ in $^*\mathbb{X}$. The existence of an inverse for any non-zero element of a ring is indeed stated by the formula: $\forall x \, (x = 0 \vee \exists y \, (xy = 1))$. More generally:

**Lemma 3 (Transfer Principle)** *Every first order formula is true over $^*\mathbb{X}$ iff it is true over $\mathbb{X}$.*

### 3.4.3. Non-standard reals and integers

The above general construction can simply be applied to $\mathbb{X} = \mathbb{R}$ and $I = \mathbb{N}$. The result is denoted $^*\mathbb{R}$; it is a field according to the transfer principle. By the same principle, $^*\mathbb{R}$ is totally ordered by $[u_n] \leq [v_n]$ iff $\mu\{n \mid u_n > v_n\} = 0$. We claim that, for any finite $[x_n] \in {}^*\mathbb{R}$, there exists a unique $st([x_n])$, call it the *standard part* of $[x_n]$, such that

$$st([x_n]) \in \mathbb{R} \quad \text{and} \quad st([x_n]) \approx [x_n] \, . \tag{2}$$

To prove this, let $x = \sup\{u \in \mathbb{R} \mid [u] \leq [x_n]\}$, where $[u]$ denotes the constant sequence equal to $u$. Since $[x_n]$ is finite, $x$ exists and we only need to show that $[x_n] - x$ is infinitesimal. If not, then there exists $y \in \mathbb{R}, y > 0$ such that $y < |x - [x_n]|$, that is, either $x < [x_n] - [y]$ or $x > [x_n] + [y]$, which both contradict the definition of $x$. The uniqueness of $x$ is clear, thus we can define $st([x_n]) = x$. Infinite non-standard reals have no standard part in $\mathbb{R}$.

It is also of interest to apply the general construction (1) to $\mathbb{X} = I = \mathbb{N}$, which results in the set $^*\mathbb{N}$ of *non-standard natural numbers*. The non-standard set $^*\mathbb{N}$ differs from $\mathbb{N}$ by the addition of *infinite natural numbers,* which are equivalence classes of sequences of integers whose essential limit is $+\infty$.

### 3.4.4. Integrals and differential equations: the standardization principle

Any sequence $(g_n)$ of functions $g_n : \mathbb{R} \mapsto \mathbb{R}$ point-wise defines a function $[g_n] : {}^*\mathbb{R} \mapsto {}^*\mathbb{R}$ by setting

$$[g_n]([x_n]) = [g_n(x_n)] \tag{3}$$

A function $^*\mathbb{R} \to {}^*\mathbb{R}$ so obtained is called *internal*. Properties of and operations on ordinary functions extend point-wise to internal functions of $^*\mathbb{R} \to {}^*\mathbb{R}$. The *non-standard version* of $g : \mathbb{R} \to \mathbb{R}$ is the internal function $^*g = [g, g, g, \cdots]$. The same notions apply to sets. An internal set $A = [A_n]$ is called *hyperfinite* if $\mu\{n \mid A_n \text{ finite}\} = 1$; the *cardinal* $|A|$ of $A$ is defined as $[|A_n|]$.

[8] Observe that, as a consequence, $\mu$ cannot be sigma-additive (in contrast to probability measures or Radon measures) in that it is *not* true that $\mu(\bigcup_n A_n) = \sum_n \mu(A_n)$ holds for an infinite denumerable sequence $A_n$ of pairwise disjoint subsets of $\mathbb{N}$.

Now, consider an infinite number $N \in {}^*\mathbb{N}$ and the set

$$T = \left\{ 0, \frac{1}{N}, \frac{2}{N}, \frac{3}{N}, \cdots \frac{N-1}{N}, 1 \right\} \tag{4}$$

By definition, if $N = [N_n]$, then $T = [T_n]$ with

$$T_n = \left\{ 0, \frac{1}{N_n}, \frac{2}{N_n}, \frac{3}{N_n}, \cdots \frac{N_n - 1}{N_n}, 1 \right\}$$

hence $|T| = [|T_n|] = [N_n + 1] = N + 1$. Now, consider an internal function $g = [g_n]$ and a hyperfinite set $A = [A_n]$. The *sum* of $g$ over $A$ can be defined:

$$\sum_{a \in A} g(a) \ =_{\text{def}} \ \left[ \sum_{a \in A_n} g_n(a) \right]$$

If $t$ is as above, and $f : \mathbb{R} \to \mathbb{R}$ is a standard function, we obtain

$$\sum_{t \in T} \frac{1}{N} {}^*f(t) = \left[ \sum_{t \in T_n} \frac{1}{N_n} f(t_n) \right] \tag{5}$$

Now, $f$ continuous implies $\sum_{t \in T_n} \frac{1}{N_n} f(t_n) \to \int_0^1 f(t)\, dt$, so,

$$\int_0^1 f(t)\, dt = st \left( \sum_{t \in T} \frac{1}{N} {}^*f(t) \right) \tag{6}$$

Under the same assumptions, for any $t \in [0, 1]$,

$$\int_0^t f(u)\, du = st \left( \sum_{u \in T, u \leq t} \frac{1}{N} {}^*f(t) \right) \tag{7}$$

Now, consider the following ODE:

$$\dot{x} = f(x, t), \quad x(0) = x_0 \tag{8}$$

Assume (8) possesses a solution $[0, 1] \ni t \mapsto x(t)$ such that the function $t \mapsto f(x(t), t)$ is continuous. Rewriting (8) in its equivalent integral form $x(t) = x_0 + \int_0^t f(x(u), u)\, du$ and using (7) yields

$$x(t) = st \left( x_0 + \sum_{u \in T, u \leq t} \frac{1}{N} {}^*f(x(u), u) \right) \tag{9}$$

The substitution in (9) of $\partial = 1/N$, which is positive and infinitesimal, yields $T = \{t_n = n\partial \mid n = 0, \cdots, N\}$. The expression in parentheses on the right hand side of (9) is the piecewise-constant right-continuous function ${}^*x(t), t \in [0, 1]$ such that, for $n = 1, \cdots, N$:

$$\begin{array}{rcl} {}^*x(t_n) & = & {}^*x(t_{n-1}) + \partial \times {}^*f({}^*x(t_{n-1}), t_{n-1}) \\ {}^*x(t_0) & = & x_0 \end{array} \tag{10}$$

By (9), the solutions $x$, of ODE (8), and ${}^*x$, as defined by recurrence (10), are related by $x = st({}^*x)$. Formula (10) can be seen as a *non-standard operational semantics* for ODE (8); one which depends on the choice of infinitesimal step parameter $\partial$. Property (9), though, expresses the idea that all these non-standard semantics are equivalent from the standard viewpoint regardless of the choice made for $\partial$. This fact is referred to as the *standardization principle*.

# 4. Software and Platforms

## 4.1. Mica: A Modal Interface Compositional Analysis Toolbox

**Participant:** Benoît Caillaud.

*http://www.irisa.fr/s4/tools/mica/*

Mica is an Ocaml library developed by Benoît Caillaud implementing the Modal Interface algebra published in [5], [4]. 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.

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.

Mica is available as an open-source distribution, under the CeCILL-C Free Software License Agreement (http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html).

## 4.2. Flipflop: A Test and Flip Net Synthesis Tool for Maintenance and Surgical Process Mining

**Participant:** Benoît Caillaud.

*http://tinyurl.com/oql6f3y*

Flipflop is a Test and Flip net synthesis tool implementing a linear algebraic polynomial time algorithm. Computations are done in the $\mathbb{Z}/2\mathbb{Z}$ 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.

This software has been designed in the context of the S3PM project (see Section 6.1).

# 5. New Results

## 5.1. Hybrid Systems Modeling

**Participants:** Albert Benveniste, Benoît Caillaud.

### *5.1.1. 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. We have addressed this issue for a hybrid modeling language that combines synchronous Lustre-like data-flow equations with Ordinary Differential Equations (ODEs) [6], [9]. 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 progresses by infinitesimal steps outside of discrete events. The causality analysis takes the form of a simple type system. In well-typed programs, signals are proved continuous during integration and can be translated into sequential code for integration with off-the-shelf ODE solvers. The effectiveness of this system is illustrated with several examples written in Zélus [9], a Lustre-like synchronous language extended with hierarchical automata and ODEs.

### *5.1.2. Semantics of multi-mode DAE systems*

Hybrid systems modelers exhibit a number of difficulties related to the mix of continuous and discrete dynamics and sensitivity to the discretization scheme. Modular modeling, where subsystems models can be simply assembled with no rework, calls for using Differential Algebraic Equations (DAE). In turn, DAE are strictly more difficult than ODE. They require sophisticated pre-processing using various notions of index before they can be submitted to a solver. We have studied some fundamental issues raised by the modeling and simulation of hybrid systems involving DAEs [10]. The objective of this work is to serve for the evolution and the design of future releases of the Modelica language for such systems. We focus on the following questions:

- What is the proper notion of index for a hybrid DAE system?
- What are the primitive statements needed for a DAE hybrid systems modeler?

The differentiation index for DAE explicitly relies on everything being differentiable. Therefore, generalizations to hybrid systems must be done with caution. We propose to rely on non-standard analysis for this. Non-standard analysis formalizes differential equations as discrete step transition systems with infinitesimal time basis. We can thus bring hybrid DAE systems to their non-standard form, where the notion of difference index can be firmly used. From this study, general hints for future releases of Modelica can be drawn.

## 5.2. Surgical Process Mining with Test and Flip Net Synthesis

**Participant:** Benoît Caillaud.

Surgical process modeling aims at providing an explicit representation of surgical procedural knowledge. *Surgical process models* are inferred from a set of surgical procedure recordings, and represent in a concise manner concurrency, causality and conflict relations between actions. In the context of the S3PM project (Section 6.1), we have investigated the use of *test and flip* nets, a mild extension of flip-flop nets, to represent surgical process models. A test and flip net synthesis algorithm, based on linear algebraic methods in the $Z/2Z$ ring is detailed. Experimental results regarding the use of this synthesis algorithm to automate the construction of simple surgical process models are also presented.

# 6. Partnerships and Cooperations

## 6.1. Regional Initiatives

- Ayman Aljarbooh's PhD is partially funded by a ARED grant of the Brittany Regional Council.

[9] http://zelus.di.ens.fr

- Benoît Caillaud is participating to the S3PM project of the CominLabs excellence laboratory [10]. This project focuses on the computation of surgical procedural knowledge models from recordings of individual procedures, and their execution [7]. 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.

## 6.2. National Initiatives

Program:« Briques génériques du logiciel embarqué » (Embedded Software Generic Building-Blocks)

Project acronym: Sys2soft

Project title: Physics Aware Software

Duration: June 2012 – April 2016

Coordinator: Dassault Systèmes (France)

Other partners: Thales TGS / TRT / TAS, Alstom Transport, Airbus, DPS, Obeo, Soyatec

Abstract: The Sys2soft project aims at developping methods and tools supporting the design of embedded software interacting with a complex physical environment. The project advocates a methodology where both physics and software are co-modeled and co-simulated early in the design process and embedded code is generated automatically from the joint physics and software models. Extensions of the Modelica language with synchronous programming features are being investigated, as a unified framework where interacting physical and software artifacts can be modeled.

## 6.3. European Initiatives

### *6.3.1. Collaborations in European Programs, except FP7*

Program: ITEA2

Project acronym: Modrio

Project title: Model Driven Physical Systems Operation

Duration: September 2012 – November 2015

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).

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.

[10] http://www.cominlabs.ueb.eu/projects/

## 6.4. International Initiatives

### *6.4.1. Informal International Partners*

Beyond the Modrio and Sys2soft collaborative projects, we have an informal but sustained collaboration with the Dassault Systèmes team developping the Dymola tool, located in Lund, Sweden, and with the DLR in Munich, Germany, which are both prominent actors of the Modelica association. This collaboration has allowed us to have an impact on the recent evolution of the Modelica language: Version 3.3 of the language integrates several of our contributions related to the introduction of language constructs inherited from synchronous programming languages [11].

# 7. Dissemination

## 7.1. Scientific Animation

Benoît Caillaud has served in the steering and program committees of the International Conference on Application of Concurrency to System Design (ACSD'13) and of the Applications of Regions Theory (ART'13) satellite workshop. He is serving on the Evaluation Committee of INRIA.

## 7.2. Teaching - Supervision - Juries

### *7.2.1. Teaching*

Benoît Caillaud has contributed to the training programme for the computer-science option of the *agrégation* in mathematics, at ENS Cachan-Ker Lann.

### *7.2.2. Supervision*

PhD in progress Ayman Aljarbooh, *Scalable Simulation of Hybrid Systems: Language Design and Compilation*, started December 2013, supervised by Benoît Caillaud

### *7.2.3. Juries*

Benoît Caillaud has participated to the jury for the defense of Florent Avellaneda's PhD thesis, *Verification of stateful Petri-nets under a partial order semantics*, December 10th 2013, Aix-Marseille University. He has also served on the junior researcher hiring committee of Inria Sophia Antipolis - Méditerranée

# 8. Bibliography

## Major publications by the team in recent years

[1] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *A hybrid synchronous language with hierarchical automata: static typing and translation to synchronous code*, in "Proceedings of the 11th International Conference on Embedded Software, EMSOFT 2011", S. CHAKRABORTY, A. JERRAYA, S. K. BARUAH, S. FISCHMEISTER (editors), ACM, 2011, pp. 137-148, http://doi.acm.org/10.1145/2038642.2038664

[2] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *Divide and recycle: types and compilation for a hybrid synchronous language*, in "Proceedings of the ACM SIGPLAN/SIGBED 2011 conference on Languages, compilers, and tools for embedded systems, LCTES 2011, Chicago, IL, USA, April 11-14, 2011", J. VITEK, B. DE SUTTER (editors), ACM, 2011, pp. 61-70, http://doi.acm.org/10.1145/1967677.1967687

[11] See acknowledgements in section E.1.3, page 261 of https://www.modelica.org/documents/ModelicaSpec33.pdf

[3] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *Non-standard semantics of hybrid systems modelers*, in "Journal of Computer and System Sciences", 2012, vol. 78, n° 3, pp. 877-910, This work was supported by the SYNCHRONICS large scale initiative of Inria [*DOI :* 10.1016/J.JCSS.2011.08.009], http://hal.inria.fr/hal-00766726

[4] 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° RR-8147, 65 p. , http://hal.inria.fr/hal-00757488

[5] J.-B. RACLET, E. BADOUEL, A. BENVENISTE, B. CAILLAUD, A. LEGAY, R. PASSERONE. *A Modal Interface Theory for Component-based Design*, in "Fundamenta Informaticae", 2011, vol. 108, n° 1-2, pp. 119-149, http://dx.doi.org/10.3233/FI-2011-416

## Publications of the year

### International Conferences with Proceedings

[6] A. BENVENISTE, T. BOURKE, B. CAILLAUD, B. PAGANO, M. POUZET. *A Type-based Analysis of Causality Loops in Hybrid Systems Modelers*, in "17th International Conference on Hybrid Systems: Computation and Control (HSCC 2014)", Berlin, Germany, April 2014, http://hal.inria.fr/hal-00939947

[7] 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

### Research Reports

[8] A. BENVENISTE, D. NICKOVIC, T. HENZINGER. , *Compositional Contract Abstraction for System Design*, Inria, January 2014, n° RR-8460, http://hal.inria.fr/hal-00938854

### Other Publications

[9] 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", http://hal.inria.fr/hal-00938866

[10] 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, http://hal.inria.fr/hal-00938891

## References in notes

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

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

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

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

[15] A. BENVENISTE, T. BOURKE, B. CAILLAUD, M. POUZET. *Non-Standard Semantics of Hybrid Systems Modelers*, in "Journal of Computer and System Sciences (JCSS)", 2012, vol. 78, n$^{o}$ 3, pp. 877–910, http://dx.doi.org/10.1016/j.jcss.2011.08.009

[16] 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

[17] 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

[18] 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

[19] 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

[20] 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

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

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

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

[24] 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

[25] 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

[26] 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

[27] 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

[28] 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

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

[30] 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

[31] 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

[32] 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

[33] 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

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

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

[36] 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

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

[38] 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

[39] 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

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

[41] 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

[42] 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

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

[44] 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

[45] 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

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

[47] 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

[48] 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

[49] 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

[50] 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

[51] 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