Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

An Interpretation of Concurrent Hybrid Time Systems over Multi-clock Systems

Marius C. Bujorianu, Manuela L. Bujorianu, Rom Langerak. *

* *Faculty of Computer Science, Unversity of Twente, The Netherlands*
e-mail: manuela@ ewi.utwente.nl

Abstract: In this paper, we present a multiclock model for real time abstractions of hybrid systems. We call Hybrid Time systems the resulting model, which is constructed using category theory. Such systems are characterized by heterogeneous timing, some components having discrete time and others continuous time. We define a timed (or clock) system as a functor from a category of states to a category of time values. We further define concurrent composition operators and bisimulation.

Keywords: concurrent hybrid systems, multiclock systems, real time, process algebra, category theory.

1. INTRODUCTION

It is widely accepted the definition of hybrid systems (abbreviated HS from now on) as systems whose behavior exhibits discrete and continuous state space evolutions. The most popular model is that of hybrid automata, where a discrete automaton controls several continuous dynamical systems. The main focus of the hybrid system research was in modeling and verification. But a very important problem is that of software development of these systems. A relevant characteristic of software is discreteness, which means that the continuous state spaces can not be represented directly. An important methodological step in constructing discrete abstractions (software models) of HS is given by real time systems. These systems that are characterized by a dense set of time values have now well established development methods. HS are abstracted into a subclass of real time systems that have some components working in discrete time and some components characterized by dense or real valued time. We call the systems of this subclass *hybrid time systems* (abbreviated HT systems).

It is desirable that the HT systems to preserve the main characteristics of the HS of interest, as concurrency, bisimulation and compositional semantics. In recent research these properties have been defined using category theory. For example, P. Tabuada, G. Pappas and coworkers in a series of papers Tabuada [2004], Pappas [2004] have defined and studied bisimulation of hybrid systems using open maps and compositional specification using category theory. A natural step to extend these concepts to RT systems is via category theory.

We introduce a new category of real time and hybrid systems. We define bisimulation of sytems in this category, and we show that it coincides with the concept introduced by Tabuada, Pappas and coworkers, in the case of hybrid systems. The advantages of the categorical approach are in the compositional semantics and the definition of bisimilarity. Using the compositionality of the semantics we define composition operators for RT systems, the most important being concurrent composition. We adhere to the interleaving philosophy of concurrency, specific to process algebra, and use the generic language of Winskel [1995]. This language is very general and it generalise the CCS and CSP, the most used concurrency languages.

The paper is structured as follows. In the next section we present a short categorical background (mainly to fix the notations). In section three the category of time systems and their bisimulation are introduced. Section four is dedicated to examples familiar from control theory. In section five a concurrent language for time systems is rigorously defined. In the final section, some conclusions are drawn and related work is discussed.

2. A QUICK TOUR IN CATEGORY THEORY

The paper makes use of category theory at an advanced level. The excellent monograph Barr [1990] covers all categorical background we use.

A category can be interpreted in many ways (as axiomatic structure, as a logic theory, as a type, etc Barr [1990]). In this paper, we deal mainly with higher order categories. Categories are related by functors. An endofunctor is a functor with the same domain and codomain. The functors themselves can be organised into a category having arrows natural transformations. These transformations can be thought of as functors between functor categories. This contruction can be iterated indefinitely, the resulted categories are called higher order. Functors get, in this way, a rich algebraic structure. They can be composed by: sequential composition, Cartesian products, tensor products, coproducts, pushouts and pullbacks. Functors will be interpreted as systems and, thus, the operations of the functorial algebra will provide composition operators for sytems. This is key point in achieving compositionality.

We inspire from Lawvere's functorial semantics of equational (or type) theories Lawvere [1963]. In his approach every element of interest is defined as a functor. In fact, the Yoneda lemma Barr [1990] guarantees a representation for every object in a category, as a special kind of functor called presheaf. This is essentially a set valued functor. In our approach, the functors are time valued. The representation of time as a set of values is insufficient since the time has an algebraic structure (discrete, continuous and combined). Therefore, functors representing RT systems need to be valued in a category of time values. We define an RT system as a functor from a category $\mathbf{P}$, of states, to a category $\mathbf{T}$ of time values. The category $\mathbf{P}$ can be thought of as modeling the structure of the *plant* and the category $\mathbf{T}$ models the structure of the *controller*. We further define a category of RT systems, where morphisms are functors between state spaces that preserve the timed control. The key point is that morphisms can be used to define discrete state space abstractions, whilst preserving the real time behavior.

Notational Conventions. The symbols used to denote categories will be boldfaced. The objects of a category will be denoted by capital letters, and the arrows using small Greek letters. Functors will be, denoted using small letters or capital Greek letters. Given a category $\mathbf{C}$, we write $|\mathbf{C}|$ to denote its class of objects and $\mathbf{C}$ to denote its class of arrows. Composition in a category is written in diagrammatic order: given $\alpha : A \to B$ and $\beta : B \to C$ their composite is written as $\alpha; \beta : A \to C$. Application of an functor f to an argument A is denoted by $f[A]$ or by $f.A$.

When a category can be simulated in another category, we model this situation using a pair of functors called adjunction. Every adjunction gives rise to a monad, defined in the following.

Factorization Category

For every arrow $A \stackrel{\alpha}{\to} C \in \mathbf{C}$, we define its corresponding *factorization category* $\uparrow\alpha\downarrow$ having

- objects factorizations $A \stackrel{\beta}{\to} B \stackrel{\gamma}{\to} C$ of α in $\mathbf{C}$, and
- arrows between $A \stackrel{\beta}{\to} B \stackrel{\gamma}{\to} C$ and $A \stackrel{\beta'}{\to} B' \stackrel{\gamma'}{\to} C$ given by $\mathbf{C}-$arrows $B \stackrel{\delta}{\to} B'$ such that $\beta' = \beta; \delta$ and $\gamma' = \delta; \gamma$.

Category of Twisted Arrows

The MacLane's category of *twisted arrows* of a category $\mathbf{C}$, denoted by $\overleftrightarrow{\mathbf{C}}$, has

- objects the arrows of $\mathbf{C}$ and
- arrows between objects $A \stackrel{\alpha}{\to} B$ and $A' \stackrel{\beta}{\to} B'$ are pairs $< A \stackrel{\varphi}{\to} A, B' \stackrel{\psi}{\to} B >$, where $A, A', B, B' \in |\mathbf{C}|$, such that the diagram

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & B \\ \varphi\downarrow & & \uparrow\psi \\ A' & \xrightarrow[\beta]{} & B' \end{array}$$

commutes in $\mathbf{C}$. By $\mathbf{C}$ we denote the subcategoy of $\overleftrightarrow{\mathbf{C}}$ having arrows of the form $\langle id, \psi \rangle$.

Tensor Category

The *tensor category* $\mathbf{C} \otimes \mathbf{D}$ of two small categories $\mathbf{C}$ and $\mathbf{D}$ has

- objects given by pairs $A \otimes B$, with $A \in |\mathbf{C}|$ and $B \in |\mathbf{D}|$,
- arrows given by shuffled sequences of nonidentic composable arrows in $\mathbf{C}$ and $\mathbf{D}$

Monad over a Category

A *monad* over the category $\mathbf{C}$ is a triple (T, η, μ), where T is an endofunctor on $\mathbf{C}$ and $\eta : id_c \to T$ and $\mu : T^2 \to T$ are defined by

$$\begin{array}{ccc} & \mathbf{C} & \\ {}^{id_c}\uparrow & \stackrel{\eta}{\to} & \uparrow{}^{T} \\ & \mathbf{C} & \end{array} \qquad\qquad \begin{array}{ccc} & \mathbf{C} & \\ {}^{T^2}\uparrow & \stackrel{\mu}{\to} & \uparrow{}^{T} \\ & \mathbf{C} & \end{array}$$

are natural transformations, subject to the following conditions:

1. $\mu; (T; \mu) \equiv \mu; (\mu; T)$ (The associative law);
2. $\mu; (\eta; T) \equiv 1_{\mathbf{F}} \equiv \mu; (T; \eta)$ (The left and right unit laws).

3. A CATEGORICAL ACCOUNT OF TIME SYSTEMS

Let **Plant** and **Time** be two small categories.

A *real time system* (*RT system* for short) over a small category **Time** is a functor

$$\Lambda : \ \mathbf{Plant} \to \mathbf{Time}$$

satisfying the so called *Lawvere condition* Lawvere [1986]: for every configuration $P \in \mathbf{Plant}$ and $T_0, T_1 \in \mathbf{Time}$, if $\Lambda.P = T_0 \cdot T_1$ in **Time** then there exist unique configurations P_0, P_1 in **Plant** for which

$$P = P_0; P_1 \text{ and } \Lambda.P_0 = T_0 \text{ and } \Lambda.P_1 = T_1$$

The category $\mathbf{Time}_{\Uparrow}$ has:

- objects: RT systems over **Time** and
- arrows given by functors $\mathbf{Plant} \stackrel{f}{\to} \mathbf{Plant}'$ such that $\Lambda = \Lambda'; f$.

The **Plant** can be, for example, the category of models of a formal specification Bujorianu [2004].

We denote by $\mathbf{Time}_{\uparrow}$ the category of functors $\mathbf{Plant} \stackrel{f}{\to} \mathbf{Plant}'$ for which the Lawvere condition is omitted.

It is possible to define a faithful functor from $\overleftrightarrow{\mathbf{T}}$ to $\mathbf{T}_{\Uparrow}$. This functor sends:

- each $\overleftrightarrow{\mathbf{T}}$ -object $A \stackrel{\alpha}{\to} B$ to the $\mathbf{T}_{\Uparrow}$ -object defined by $\alpha_{\mathbf{T}} : \uparrow\alpha\downarrow \to \mathbf{T}$, where the RT system $\alpha_{\mathbf{T}}$ maps $A \stackrel{\alpha}{\to} B \stackrel{\beta}{\to} C$ into B and
- each $< A \stackrel{\varphi}{\to} A, B' \stackrel{\psi}{\to} B >$ into the functor $f_{\varphi,\psi} : \uparrow\alpha\downarrow \to \uparrow\beta\downarrow$ that associates to (α, β) the pair $(\varphi; \alpha, \beta; \psi)$.

This embedding says, essentially, that $\mathbf{Time}_{\Uparrow}$ is the free cocompletion of $\overleftrightarrow{\mathbf{T}}$ with respect to the pushouts of the form

$$\begin{array}{ccccc} & & id_B & & \\ & \overset{(\alpha, id_B)}{\swarrow} & | & \overset{(id_B,\beta)}{\searrow} & \\ \alpha & & h & & \beta \\ & \underset{(id_A,\beta)}{\searrow} & \downarrow & \underset{(\alpha, id_C)}{\swarrow} & \\ & & \alpha;\beta & & \end{array}$$

where $A \overset{\alpha}{\to} B \overset{\beta}{\to} C$ in $\mathbf{C}$.

We can also consider the obvious embedding of $\overleftarrow{\mathbf{T}}$ in $\mathbf{T}_{\Uparrow}$.

For every RT system $\Lambda : \ \mathbf{Plant} \to \mathbf{Time}$ we define the category

$$\mathbf{Time}_{\Uparrow}^{\Lambda},$$

having

- objects given by triples $(\mathbf{X}, \Omega, f)$ where $\mathbf{X}$ is a category, $\Omega : \mathbf{X} \to \mathbf{Time}$ is an RT system and $f : \mathbf{Plant} \to \mathbf{X}$ is a functor
- arrows $(\mathbf{X}, \Omega, f) \overset{g}{\to} (\mathbf{X}', \Omega', f')$ given by functors $g : \mathbf{X} \to \mathbf{X}'$ such that $\Omega'; g = \Omega$ and $f' = f; g$.

Assumption 1. From now on we suppose that, for each arrow α the category $\uparrow\alpha\downarrow$ is a linear preorder.

Consider a functor $\Lambda : \ \mathbf{P} \to \mathbf{T}$. We define the category $\overline{\mathbf{P}}$ having

- objects: pairs (T, π) where $T \in |\mathbf{T}|$ and $\pi \in \mathbf{T}_{\uparrow}$ is a morphism between (pseudo)systems $(id_T)_{\mathbf{T}} : \uparrow id_T\downarrow \to \mathbf{T}$ and $\Lambda : \ \mathbf{P} \to \mathbf{T}$.
- arrows between (T, π) and $(T'\pi')$ are given by pairs (t, κ) where $t : T \to T' \in \mathbf{T}$ and κ is morphism between systems $t_{\mathbf{T}} : \uparrow t\downarrow \to \mathbf{T}$ and $\Lambda : \ \mathbf{P} \to \mathbf{T}$ such that $\pi = f_{id_T, t}; \kappa$ and $\pi' = f_{id_{T'}, t}; \kappa$
- the identities are $id_{(T,\pi)} = (id_T, \pi)$ and arrow composition given by pushout

Proposition 1. The embedding $\mathbf{Time}_{\Uparrow} \hookrightarrow \mathbf{Time}_{\uparrow}$ admits a right adjoint $\mathcal{F} : \mathbf{Time}_{\uparrow} \longrightarrow \mathbf{Time}_{\Uparrow}$

Hint for proof: define $\mathcal{F}[\Lambda]$, for $\Lambda : \ \mathbf{P} \to \mathbf{T}$, as the first projection functor $\overline{\mathbf{P}} \to \mathbf{T}$.

Consider two arbitrary RT systems $\Lambda : \ \mathbf{Plant} \to \mathbf{Time}$ and $\Lambda' : \ \mathbf{Plant}' \to \mathbf{Time}'$ and a functor $\mathbf{Plant} \overset{f}{\to} \mathbf{Plant}'$.

A *simulation* Winskel [1995] between Λ and Λ' is a relation ρ between $|\mathbf{Plant}|$ and $|\mathbf{Plant}'|$ such that:

if $S_1(\rho)S_1'$ then, for every arrow $S_1 \overset{\alpha}{\to} S_2$ in $\mathbf{Plant}$ there is an arrow $S_1' \overset{\alpha'}{\to} S_2'$ in $\mathbf{Plant}'$ with $S_2(\rho)S_2'$ and $\Lambda.\alpha = \Lambda'.\alpha'$.

Two states $S \in |\mathbf{Plant}|$ and $S' \in |\mathbf{Plant}'|$ with $\Lambda.S = \Lambda'.S'$ are called *open bisimilar* if there is a span of open functors Winskel [1996] $\mathbf{Plant} \overset{f}{\leftarrow} \mathbf{Plant}^{\$} \overset{f'}{\to} \mathbf{Plant}'$ such that there is $S^{\$} \in |\mathbf{Plant}^{\$}|$ with $f[S^{\$}] = S$ and $f'[S^{\$}] = S'$.

Proposition 2. The functor f is open w.r.t. the embedding of $\overleftarrow{\mathbf{T}}$ in $\mathbf{T}_{\Uparrow}$ if and only if for each $S \in |\mathbf{Plant}|$ and $f[S] \overset{\alpha'}{\to} S'$ in $\mathbf{Plant}'$ there exists $S \overset{\alpha}{\to} S^{\$}$ in $\mathbf{Plant}$ with $f[\alpha] = \alpha'$.

This proposition, adapted from Bunge [2000], states that the concepts of bisimilarity and open bisimilarity coincide for RT systems.

4. EXAMPLES

In this section we consider the main instantiations of the theory of RT systems from the previous section. The $\mathbf{Time}_{\Uparrow}$ can be instantiated with categories of states modelling both discrete and continuous (time) plants, by simply taking $\mathbf{Time}$ to be $\mathbf{Z}_{+}$, the nonnegative integers or $\mathbf{R}_{+}$, the nonnegative real numbers (which are simple monoidal categories). An important instantiation of $\mathbf{Time}$ is the category $[A]$ generated by the free monoid $(A^*, \bullet, \epsilon)$ formed with symbols from the set A.

The category $\mathbf{Time}_{\Uparrow}^{\epsilon_{\mathbf{Time}}}$ is the important category of transition systems with initial state (the labelled transition systems).

As the categorical theory of transition systems with discrete time is better understood Winskel [1995], in the following we detail the structure of continuous time systems.

A *duration* Lawvere [1986] is a functor $RT : \mathbf{Plant} \to \mathbf{R}_{+}$ such that, for every $P \in \mathbf{Plant}$ and $t_0, t_1 \in \mathbf{R}_{+}$, if $RT[P] = t_0 + t_1$ in $\mathbf{Time}$ then there exists a unique factorisation $P = P_0 \cdot P_1$ in $\mathbf{Plant}$ for which $f(P_0) = t_0$ and $f(P_1) = t_1$.

The category $\mathbf{Dur}$ has

- objects: durations (flows) and
- arrows: functors $\mathbf{Plant} \overset{G}{\to} \mathbf{Plant}'$ such that $RT = RT'; G$.

The most important examples of durations are provided by the solutions of differential equations (the flows). Let us consider the differential equation

$$\frac{dx}{dt} = a \qquad \text{(eq)}$$

whose initial value problem with initial condition $x(0) = s$ has unique solution

$$\widetilde{s}(t) = a.t + s.$$

It can be regarded as the flow $\mathbf{Plant} \to \mathbf{R}_{+}$ mapping arrows $s_0 \overset{t}{\to} s_1$ to $t > 0$, where

$$|\mathbf{Plant}| \overset{def}{=} \mathbf{R}$$

and, for $t \in \mathbf{R}_{+}$, we have $s_0 \overset{t}{\to} s_1$in $\mathbf{S}$ if and only if there exists a solution of (eq) $\sigma : I \to \mathbf{R}$ and $t_0 \leq t_1$ in I such that

$$\sigma(t_0) = s_0, \ \sigma(t_1) = s_1, \text{ and } t_1 - t_0 = t$$

(i.e., if and only if $\widetilde{s}_0(t) = s_1$).

The *category of continuous paths*, denoted by $\overleftrightarrow{\mathbf{R}}$,has

- the set of positive reals as objects and

- an arrow $t \to t'$ a way of placing an interval of length t within an interval of length t' , i.e. pairs (x, y) in $\mathbf{R}_+$ such that $x + t + y = t'$.

The category $\overleftarrow{\mathbf{R}}$ is the subcategory of $\overleftrightarrow{\mathbf{R}}$ arrows of the form $(0, y)$.

There is an embedding $\overleftrightarrow{\mathbf{R}} \hookrightarrow \mathbf{Dur}$ that sends a non-negative real number t to the difference flow

$$|t| \to \mathbf{R}_+ \ , \ x \leq y \to y - x \ ,$$

where $|t|$ is the interval poset $([0, t], \leq)$. We can consider further the faithful functors $\overleftarrow{\mathbf{R}} \to \overleftrightarrow{\mathbf{R}} \hookrightarrow \mathbf{Dur}$

5. COMPOSITIONS OF RT SYSTEMS

The composition operators unary or binary. They apply to RT systems and produce new RT systems. As any system is a functor, it results that composition operators are functors from functor categories to functor categories. When the composition operator is binary, we deal with bifunctors. In defining composition operators, one can expect then many tricky compositions from functorial algebra: products, coproducts, pullbacks, pushouts, tensor product, etc. We use both prefix and infix notations and the position of the arguments is indicated by underscores.

Control translation

We define two unary operators that describe two different ways to translate RT systems when translating the control category.

Let $\theta : \mathbf{Time} \to \mathbf{Time}'$ be a functor.

Define the functor $\Gamma_{\Upsilon\theta} : \mathbf{Time}_{\Uparrow} \to \mathbf{Time}'_{\Uparrow}$ that maps $\Lambda : \ \mathbf{Plant} \to \mathbf{Time}$ into $\Lambda; \theta_{\Upsilon} : \ \mathbf{Plant} \to \mathbf{Time}'$.

The functor $\digamma_{\theta} : \mathbf{Time}_{\Uparrow} \to \mathbf{Time}'_{\Uparrow}$ that maps Λ : $\mathbf{Plant} \to \mathbf{Time}$ into $\digamma[\Lambda; \theta_{\Upsilon}] : \ \mathbf{Plant} \to \mathbf{Time}'$.

Restriction and pullback composition of RT systems

Consider two arbitrary RT systems $\Lambda : \ \mathbf{Plant} \to \mathbf{Time}$ and $\Lambda' : \ \mathbf{Plant} \to \mathbf{Time}'$ and a functor $\mathbf{Time} \xrightarrow{\Upsilon} \mathbf{Time}'$.

For any morphism of RT systems $\mathbf{Plant} \xrightarrow{f} \mathbf{Plant}'$ define

- the functor $(\_)_{|f} : \mathbf{Time}^{\Lambda}_{\Uparrow} \to \mathbf{Time}^{\Lambda'}_{\Uparrow}$ that maps $(\mathbf{X}, \Omega, g)$ to $(\mathbf{X}, \Omega, f; g)$.
- the *pullback functor* $\widehat{f}$, along f, as the functor $\widehat{f}$: $\mathbf{Time}'_{\Uparrow} \to \mathbf{Time}_{\Uparrow}$ that maps the RT system Λ : $\mathbf{Plant} \to \mathbf{Time}$ into $\widehat{f}\Lambda : \ \widehat{f}.\mathbf{Plant} \to \mathbf{Time}$, where $\widehat{f}.\mathbf{Plant}$ is the subcategory of $\mathbf{Time} \times \mathbf{Plant}$ with arrows $(T, S) \xrightarrow{(\lambda, \alpha)} (T', S')$ such that $\Upsilon.\lambda = \Lambda.\alpha$.

Process algebra

We recall the basic operators of the process algebra from Winskel [1995].

Process Terms

$P =: nil \mid P_{\restriction i} \mid P\{\lambda\} \mid P \times P' \mid P \| P' \mid P + P' \mid l.P$

Process Equations

$Eq =: P = P'$ where P, P' are process terms.

A *specification* consists of a signature and a set of equations.

In the following, we construct a semantics of this process algebra where every term (process) is a RT system.

Sum and product of RT systems

The operations of sum $(+)$ and product $(\times)$ between functors generate RT system operations.

The *sum functor* is defined by

$$\_ + \_ : \mathbf{Time}_{\Uparrow} \times \mathbf{Time}'_{\Uparrow} \to (\mathbf{Time} + \mathbf{Time}')_{\Uparrow}$$
$$\Lambda + \Lambda' : \mathbf{Plant} + \mathbf{Plant}' \to \mathbf{Time} + \mathbf{Time}'$$

The fibred sum $\oplus$ is defined by $\mathbf{Time}_{\Uparrow} \times \mathbf{Time}_{\Uparrow} \xrightarrow{+} (\mathbf{Time} + \mathbf{Time})_{\Uparrow} \xrightarrow{\Gamma_{[id, id]}} \mathbf{Time}_{\Uparrow}$

The *product functor* is defined by

$$\_ \times \_ : \mathbf{Time}_{\Uparrow} \times \mathbf{Time}'_{\Uparrow} \to (\mathbf{Time} \times \mathbf{Time}')_{\Uparrow}$$
$$\Lambda \times \Lambda' : \mathbf{Plant} \times \mathbf{Plant}' \to \mathbf{Time} \times \mathbf{Time}'$$

Tensor product and pushout composition of RT systems

The *tensor product* of two systems $\Lambda : \ \mathbf{Plant}_1 \to [S]$ and $\Gamma : \ \mathbf{Plant}_2 \to [T]$ is induced by the tensor category of two monoids, which is their coproduct in the category of monoids:

$$[S \cup T] = [S] \otimes [T]$$

Therefore, we can consider a functor $\_ \otimes \_ : [S]_{\Uparrow} \times [T]_{\Uparrow} \to [S \cup T]_{\Uparrow}$ given by

$$\Lambda \otimes \Gamma : \mathbf{Plant}_1 \otimes \mathbf{Plant}_2 \to [S] \otimes [T]$$

Consider two arbitrary RT systems $\Lambda_1 : \ \mathbf{P_1} \to \mathbf{T}$ and $\Lambda_2 : \ \mathbf{P_2} \to \mathbf{T}$ and $\mathbf{P_1} \xrightarrow{p} \mathbf{P_2}$ in $\mathbf{T}_{\Uparrow}$.

The *pushout functor*, along p, as the functor $p \odot \_ : \mathbf{T}^{\Lambda 1}_{\Uparrow} \to \mathbf{T}^{\Lambda 2}_{\Uparrow}$ that maps $(\mathbf{X}, \Omega, f)$ into $(\mathbf{P_2} \odot \mathbf{X}, \Lambda_2 \odot \Omega, p \odot f)$ where $\mathbf{P_2} \odot \mathbf{X}$, $\Lambda_2 \odot \Omega$ and $p \odot f$ are given by the following commutative diagram

$$\begin{array}{ccccc} & & \mathbf{P_1} & & \\ & {}^{p}\swarrow & & \searrow^{f} & \\ \mathbf{P_2} & & & & \mathbf{X} \\ & \searrow_{p \odot f} & & \swarrow & \\ & & \mathbf{P_2} \odot \mathbf{X} & & \\ \Lambda_2 : \ \mathbf{P_2} \to \mathbf{T} & & \Lambda_2 \odot \Omega \downarrow & & \Omega : \mathbf{X} \to \mathbf{T} \\ & & \mathbf{T} & & \end{array}$$

Saturation

The saturation operation was introduced in Winskel [1996] for studying weak bisimulation. Saturation (with silent steps) by a functor $\theta : \mathbf{T}^1 \to \mathbf{T}^2$ is defined by the monad on $\mathbf{T}^1_{\Uparrow}$ induced by the following adjoints $\mathbf{T}^1_{\Uparrow} \overset{\digamma}{\leftrightarrows} \mathbf{T}^1_{\uparrow} \underset{\Gamma_\theta}{\overset{\widehat{\theta}}{\leftrightarrows}} \mathbf{T}^2_{\uparrow}$.

6. HYBRID TIME SYSTEMS

6.1 The shuffle model

Let L be a set and $[L]$ be the generated (category) free monoid. The category $[L]_{\Uparrow}$ corresponds to the discrete, $L-$ labelled transition systems.

Consider $\mathbf{R}_+$ the category of positive reals. The category $(\mathbf{R}_+)_{\Uparrow}$ corresponds to the category of durations (i.e. continuous dynamical systems).

A HT system is an object of the category

$$([L] \otimes \mathbf{R}_+)_{\Uparrow}$$

i.e. an RT system $\Psi : \mathbf{S} \rightarrow [L] \otimes \mathbf{R}_+$ for which the evolutions in the state space $\mathbf{S}$ factor uniquely as shuffles of discrete and continuous evolutions.

One can observe that hybrid systems as defined in Henzinger [1996] are models of HT systems.

There exist two adjunctions

$$[L]_{\Uparrow} \leftrightarrows ([L] \otimes \mathbf{R}_+)_{\Uparrow}$$

and $(\mathbf{R}_+)_{\Uparrow} \leftrightarrows ([L] \otimes \mathbf{R}_+)_{\Uparrow}$ that relate the models of discrete time, real time and hybrid time systems. In this way, the concepts of open maps and bisimilarity for HT systems extend that of discrete and continuous systems.

6.2 Concurrency and Abstraction

The symbol τ denotes the silent action.

Relabelling

Consider a function $\mu : M \rightarrow N$ and define the functor $\_\{\mu\} : [M]_{\Uparrow} \rightarrow [N]_{\Uparrow}$ as $\Lambda\{\mu\} = \Gamma_{\mu^*}[\Lambda]$ where μ^* is the free homomorphic extension of the function which maps $m \in M \mapsto \mu(m)$.

Restriction

Consider an inclusion function $\iota : M \rightarrow N$ and define the functor $\_|\iota : [N]_{\Uparrow} \rightarrow [M]_{\Uparrow}$ as $\Lambda|\iota = \widehat{(\mu^*)}[\Lambda]$

Prefix

The prefix with an action $m \in M$ is given by the functor $m.\_ : [M]_{\Uparrow}^{\epsilon[M]} \rightarrow [M]_{\Uparrow}^{\epsilon[M]}$ defined as $m.\Lambda = (f_{m,\varepsilon} \odot \Lambda); (\Lambda | f_{m,\varepsilon})$

Choice

Consider the inclusion function $\iota_M : M \rightarrow M \cup N$, $\iota_N : N \rightarrow M \cup N$ and define the functor $\_ \vee \_ : [M]_{\Uparrow} \times [N]_{\Uparrow} \rightarrow [M \cup N]_{\Uparrow}$ as $\Lambda \vee \Omega = (\Lambda\{\iota_M\}, \Omega\{\iota_N\}); (\Lambda \oplus \Omega)$

Parallel composition of RT systems

We follow Winskel [1995] and define parallel composition using product, restriction and relabelling.

Let us consider two RT systems, belonging to the categories $[M \cup N \cup \{\tau\}]_{\Uparrow}$ and $\{N \cup L \cup \{\tau\}]_{\Uparrow}$. The set of actions N plays the role synchronization. Their parallel composition is given by the functor

$$\_||\_ : \mathbf{T}^1_{\Uparrow} \times \mathbf{T}^2_{\Uparrow} \rightarrow \mathbf{T}^3_{\Uparrow}$$

defined by

$$\Lambda_1 || \Lambda_2 = \widehat{\Xi}[\Lambda_1, \Lambda_2]; \Gamma_\rho,$$

where $\Xi : \mathbf{T}^1 \times \mathbf{T}^2 \rightarrow \mathbf{S}$ is called the synchronization functor and $\rho : \mathbf{S} \rightarrow \mathbf{T}^3$ is a ("relabelled") RT system.

In the discrete case, the synchronization functor is defined as $\widehat{\Xi}_\rho[\_, \_] : [M \cup N \cup \{\tau\}]_{\Uparrow} \times \{N \cup L \cup \{\tau\}]_{\Uparrow} \rightarrow [M \cup L \cup \{\tau\}]_{\Uparrow}$ with the synchronisation map

$[M \cup N \cup L \cup \{(\tau, *)\} \cup \{(*, \tau)\}] \stackrel{\Xi}{\rightarrow} [M \cup N \cup \{\tau\}] \times [N \cup L \cup \{\tau\}]$

given as the free homomorphic extension of the function mapping:

$m \in M \mapsto (m, \varepsilon), \quad (\tau, *) \mapsto (\tau, \varepsilon), \quad n \in N \mapsto (n, n),$

$(*, \tau) \mapsto (\varepsilon, \tau), \quad l \in L \mapsto (\varepsilon, l);$

and where the relabelling function $M \cup \{(\tau, *)\} \cup N \cup \{(*, \tau)\} \cup L \stackrel{\rho}{\rightarrow} M \cup \{\tau\} \cup L$ is the following mapping:

$m \in M \mapsto m, \quad (\tau, *) \mapsto \tau, \quad n \in N \mapsto \tau,$

$(*, \tau) \mapsto \tau, \quad l \in L \mapsto l.$

Time abstraction of HT systems

Consider the monoid $\mathbf{B} = (\{id, \tau\}, \bullet, id\}$ with $\tau \bullet \tau = \tau$, $\tau \bullet id = \tau$ and the pullback square

$$\begin{array}{ccc} \mathbf{C} & \stackrel{\overline{f}}{\longrightarrow} & [M] \otimes \mathbf{R}_+ \\ \overline{\sigma} \downarrow & & \downarrow \sigma \\ [M \cup \{\tau\}] & \underset{f}{\longrightarrow} & [M] \times \mathbf{B} \end{array}$$

where $\sigma : [M] \otimes \mathbf{R}_+ \rightarrow [M] \times \mathbf{B}$ is the unique homomorphism mapping $m \in M \mapsto m$, $t \in \mathbf{R}_+ \mapsto \tau$ and $f : [M \cup \{\tau\}] \rightarrow [M] \times \mathbf{B}$ is the free homomorphic extension of the function mapping $m \in M \mapsto m$, $\tau \mapsto \tau$

The time abstraction is defined by the functor $\partial : [M] \otimes \mathbf{R}_+ \rightarrow [M \cup \{\tau\}]$ computed as $\partial = \digamma_{\overline{\sigma}}; \widehat{(\overline{f})}$

7. FINAL REMARKS

In this paper, we have defined a functorial semantics for time systems. Such a system is simply a functor from a category of "abstract states" (also generically called the plant) to a category of "abstract time" (generically called the control). We have introduced a concept of bisimulation and composition operators for these systems. We have shown that when the control category is the free monoid, generated by a set of labels L, then the time systems are precisely the category of L-labelled transition systems. When the control category is the monoid of nonnegative reals then the time systems are precisely the category of continuous dynamical systems. A semantics of the concurrent language from Winskel [1995] is defined in this category, and a bisimulation concept is introduced. Moreover, in Bujorianu [2004] we show that the standard operational semantics from Winskel [1995] is sound in this categorical semantics (that means that the two semantics are compatible, with the categorical semantics playing the role of the denotational one). This construction provides a very elegant specification of concurrent real time systems, as well a rigorous semantics for sytem verification (in the spirit of Tabuada [2004]). Many theoretical results and verification algorithms become in this way available for timed systems.

An important class of time systems constitutes the hybrid time systems. These systems have the control category given by the product of a free semigroup and the monoid of nonnegative reals. The evolutions in the system state space consist of sequences interleaving discrete and continuous transitions. Obviously, hybrid automata are examples of hybrid time systems. There are also HT systems that are not necessarily hybrid automata. That is happening when the state space is discrete and some system behaviours are real time. We have defined software viewpoints that are HT systems and we have constructed their unification. In general, viewpoint unification is very difficult to construct and to the authors' knowledge there is no such construction for time systems.

The most succesful categorical approaches in control engineering are those of Tabuada [2004], Pappas [2004], and van Schuppen [2005]. The most related to our approach is obviously that of P. Tabuada, G. Pappas e.a.: in section 3 we have shown actually that our concept of bisimulation is equivalent with their concept. Our work is based on different primitives, both in hybrid systems and category theory. We continue their line of research, which is characterised by a very clear connection between system engineering and computer science, by introducing process algebra and modular development. The functorial semantics is very suitable for simulations using functional languages (like ML and Haskell). For future work we consider developing further the multiclock interpretation towards an implementation into synchronous languages.

There is also a limited number of categorical approaches to hybrid systems from computer science, but none of them has generated a line of research relevant to the control system community. These approaches comprise:

- Sernadas [2000] have defined hybrid systems as specifications in an institution. An institution is a categorical, model theoretic formalisation of logics (in this case a temporal logic). This work treats only the logical aspects of a specification language for hybrid systems.

- Jacobs [2000] has defined a coalgebraic model for hybrid systems. In this model, discrete and continuous transitions coexist on a common state space. The model is reach in examples and the coalgebraic concept of bisimulation can be defined. This approach is based on a different branch of category theory, and it is mainly focuss on defining bisimulation of hybrid systems using coalgebraic bisimulation.

- Bunge [2000] and Fiore [2000] have defined a model for linear control systems using fibrations (i.e. categorical logic). This model is rich in universal characterisations and it is similar to our approach. The system development approach used is driven by refinement.

The mathematical "cookbook" relies heavily on the functorial algebra. The abstractness of the mathematical framework is the only price to pay for constructing a compositional semantics for a class of systems that are not, in general, easy composable.

In a forthcoming paper, we will define a modular development methodology based on viewpoints citeAMST and we will present case studies, as well HT systems abstractions of hybrid systems.

ACKNOWLEDGEMENTS

This work was partially supported by the NWO project AiSHA. The authors thank Glynn Winskel and Marcelo Fiore for useful suggestions.

REFERENCES

M. Barr and G. Wells. Category theory for computing science. Prentice Hall, 1990.

M.C. Bujorianu. Integration of specification languages using viewpoints. *IFM 2004*, Springer-Verlag LNCS 2999, pages 421–440, 2004.

M.C. Bujorianu and E.A. Boiten. Towards correspondence carrying specifications. *AMAST 2004*, Springer-Verlag LNCS, 2004.

M. Bunge and M.P. Fiore. Unique factorisation lifting functors and categories of linearly-controlled processes. *Mathematical Structures in Computer Science*, 10(2): 137–163, 2000.

M.P. Fiore. Fibred models of processes. *IFIP TCS 2000*, Springer-Verlag, pages 457–473, 1956.

Hagverdi, P. Tabuada, and G.J. Pappas Bisimulation relations for dynamical and control systems. *Category Theory and Computer Science*, ENTCS, 2004.

T.A. Henzinger. The theory of hybrid automata. *LICS'96*, IEEE Press, pages 278–292, 1996.

B. Jacobs. Object-oriented hybrid systems of coalgebras plus monoid actions. *Theoretical Computer Science*, 239(1):41–95, 2000.

A. Joyal, M. Nielsen, G. Winskel. Bisimulation from Open Maps. *Information and Computation*, 14(2):203-238, 1996.

J. Komenda, J.H. van Schuppen. Control of discrete event systems with partial observations using coalgebra and coinduction. *Discrete Event Systems: Theory and Applications*, 127(2):164–185, 2005.

F.W. Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci. USA*, 1963.

F.W. Lawvere and S.H. Schaunel. Categories in continuum physics. Springer Verlag, Lecture Notes in Mathematics, vol. 1174, pages 1–16, 1986.

H. Lourenco, A. Sernadas. An Institution of Hybrid Systems. *WADT 2000*, Springer-Verlag LNCS 1827, pages 219–236, 2000.

P. Tabuada, G.J. Pappas, P. Lima. Compositional Abstractions of Hybrid Control Systems. *Journal of Discrete Event Dynamical Systems*, 14(2):203-238, 2004.

G. Winskel and M. Nielsen. Models for concurrency. *Handbook of Logic and the Foundations of Computer Science 4*, Oxford University Press, pages 1–148, 1995.