Bisimulation, Logic and Mobility for Markovian Systems*

Manuela Bujorianu and Marius Bujorianu † ‡

1 Introduction

Nowadays, anyone can easily observe an explosive development in distributed embedded systems like sensor networks, gene regulatory networks and other system biology areas. A general tendency in this development is the integration of different features, like mobility, randomness, continuity and discrete/continuous mixed behaviors. In this paper, we present two formal mechanisms for developing a formal framework, in which these various features can be investigated altogether. One mechanism represents a unifying axiomatization of deterministic and stochastic automata, in the spirit of the recently introduced paradigm called Hilbertian formal methods [6]. The second one proposes a generic technique based on the categorical domain theory for adding new features to an existing model. This mechanism constitutes a formal approach to a recent development paradigm called *multi-dimensional codesign* [5]. In the limited space of this paper, we restrict our presentation to a class of systems that mix continuous evolutions with logical mobility.

Continuous behaviors have been investigated formally mostly in the area of hybrid systems. Usually, these behaviors are associated with man made technical systems and their mathematical description consists of rather simple differential equations. In the case of embedded systems, the continuous evolutions of the environment often involve complex mathematical descriptions. For example, in a meteo system, a continuous evolution is described by a system containing up to one hundred partial differential equations. In the case of a cardiac implant, the continuous

*This work was partially funded by the NWO project AiSHA
†CICADA, University of Manchester, UK and EWI, University of Twente, NL,
‡This work was carried out before the authors affiliation with CICADA

evolutions are subject to randomized changes. The main difficulty in developing formal methods for such systems is given by their very different mathematical foundations. When probabilities are considered, fundamental system properties are lost, like the uniqueness of solution given an initial state. The idea of considering two different approaches, one for the deterministic case and one for the stochastic, is not feasible in practice. The selection of the environment characteristics that should be considered by the embedded controller is subject to frequent changes. The interaction between different characteristics is often not entirely mathematically understood and the initial deterministic model turns into a stochastic one. In the case of two different formal approaches, the addition of new functionalities would involve a complete redesign and a replacement of the old controllers. That can be very costly, especially if, for example, the sensor network has been placed in a geographical position difficult to access (think at a military application) or if a gene network must be re-created (to obtain accurate biological cultures in genetics is still a very complex process).

The first main contribution of this paper is a unifying semantic framework, in which both deterministic and stochastic environment behaviors can be modelled.

The second contribution of this paper considers the possibility to introduce logical mobility in the framework described above. We consider the categorical formalization of the π-calculus introduced and developed by Glynn Winskel and his co-authors [7]. This formalization relies on heavy categorical algebra and therefore we discuss only how Winskel's calculus can be used. In principle, Winskel's approach is constructed generically using an abstract model of computation specified as a category. The subtle point of this construction is that, in this category, a computational equivalence, described in terms of open maps must exist. When this category consists of labelled transition systems, as used in process algebra, the computational equivalence becomes the familiar concept of bisimulation. The mobile processes are then described as presheaves on this category. The computational equivalence between the mobile processes is then borrowed from this category via Yoneda embedding. We extend the behaviors of continuous systems with mobile processes by constructing suitable categories to replace this category. Obviously, there are many categories of continuous processes in the literature (especially in control theory), but these can not be used because the computational equivalence by open maps can not be defined. The main contribution of this paper is to construct a category of models of computation that unifies deterministic and stochastic evolutions and for which the open maps can be defined and generate an equivalence relation.

From a mathematical viewpoint, the paper follows two main streams. The first part uses intensively the general theory of Markov processes to introduce a unifying model of concurrent embedded systems and its concept of bisimulation. We show that this general concept of bisimulation subsumes the bisimulation of deterministic continuous and hybrid dynamical systems introduced and investigated by Tabuada e.a. [11] using open maps. In the second part, an approach based on category theory enriches the previous model with first order mobility, such that the bisimulation relation for mobile processes is compatible with the stochastic bisimulation.

2 A general axiomatisation of continuous processes

2.1 Markov Processes

The stochastic processes we consider here are randomized systems with a continuous state space, where the “noise” can be measured using transition probability measures. Markov processes form a subclass of stochastic systems for which, at any stage, future evolutions are conditioned only by the present state.

The state space is denoted by X. The state space should be a measurable space. Suppose that X is a Polish or analytic space. We consider X equipped with its Borel σ-algebra $\mathcal{B}$. Let $X_\Delta = X \cup \{\Delta\}$. Let $\mathcal{B}(X_\Delta)$ be the Borel σ-algebra of X_Δ. The set of all bounded measurable numerical functions on X is denoted by $\mathbf{B}(X)$.

A probability space $(\Omega, \mathcal{F}, P)$ is fixed and all $X-$valued random variables are defined on this probability space. The trajectories in the state space are modelled by a family of random variables (x_t) where t denotes the time. The reasoning about state change is carried out by a family of probabilities P_x one for each state $x \in X$. The construction is similar to the coalgebraic reasoning in the semantics of specification languages: the system behavior is described by given for each state the possible evolutions. For Markov processes, for each state x, the probability $P_x(x_t \in A)$ to reach a given set of state $A \subset X$ (provided that A is measurable) starting from x describes the system evolution.
The stochastic analysis identifies concepts (like infinitesimal generator, semigroup of operators, resolvent of operators) that characterize in an abstract sense the evolutions of a Markov process.

2.2 Deterministic dynamical systems

In this subsection we present the class of semi-dynamical systems, which can be thought of as “Markov processes” that “degenerated” into determinism, or what “Markov processes” would be if its transition probabilities would be given by some Dirac distributions[1].

Markov processes are generalizations of semidynamical in continuous time. They might be thought of as restrictions of dynamical systems to the positive time interval.

A *semi-dynamical system* [2] is a function ϕ: $\mathbb{R}_+ \times X_\Delta \to X_\Delta$ such that
1. ϕ is a measurable map; 2. $\phi(0, x) = x$; 3. $\phi(t_1 + t_2, x) = \phi(t_1, \phi(t_2, x))$,
4. $\phi(t, x) = \Delta \Rightarrow \phi(s, x) = \Delta, \forall s \geq t$; 5. $\phi(t, x) = \phi(t, y), \forall t > 0 \Rightarrow x = y$.

The *life time* of the system ϕ is the map $\zeta : X_\Delta \to [0, \infty]$ defined by $\zeta(x) = \inf\{t \geq 0 | \phi(t, x) = \Delta\}$. We can suppose without loosing the generality that for all $x \in X$ the life time $\zeta(x) > 0$. For each $x \in X$ the *trajectory* starting from x is $\Gamma_x = \{\phi(t, x) | t \in [0, \zeta(x))\}$. The semi-dynamical system ϕ is called *transient* if there exists $(A_n) \subset \mathcal{B}$ such that $X = \bigcup_{n \in \mathbb{N}} A_n$ and $m\{t \in [0, \infty) | \phi(t, x) \in A_n\} < \infty$, $\forall x \in X$, where m is the Lebesgue measure.

[1] Recall that the Dirac measure $\delta_x(A)$, for $x \in X$ and $A \in \mathcal{B}(X)$ is equal to 1 iff $x \in A$, and 0 otherwise.

2.3 A unifying framework

We can abstract away a set of common properties of Markov processes and semi-dynamical systems. These properties are defined less operational but rather algebraic. This unifying method derives from the so-called weak solutions of differential equations. For equations where solutions can not be computed, the existence and important analytic properties of the solutions can be established. The key point is to consider a larger space of elements that contains the solutions. A typical example of such a space constitutes $\mathbf{B}(X)$. The differential operator becomes then a linear operator on a subset of this large space. Again this operator is too complex and it is replaced by a time-indexed family of "approximating" simpler operators. This approximating family is the so-called semigroup of operators.

A family $\{P_t : \mathbf{B}(X) \to \mathbf{B}(X), t \geq 0\}$ of linear operators on $\mathbf{B}(X)$ is called *semigroup of operators* if the following conditions are satisfied: (i) semigroup property: $P_t P_s = P_{t+s}$, $t, s \geq 0$; (ii) contraction property: $||P_t f|| \leq ||f||$, $f \in \mathbf{B}(X)$. In addition, if $\lim_{t\to 0} P_t f = f$, then (P_t) is called *strongly continuous* semigroup. This concept has enough components to allow us to define powerful analytic tools such as the operator resolvent and the infinitesimal generator.
To each operator semigroup $\mathcal{P} = (P_t)$ on the Banach space $\mathbf{B}(X)$, the following mathematical objects can be associated:
1) The *resolvent of operators* $\mathcal{V} = (V_\alpha)_{\alpha \geq 0}$ associated to $\mathcal{P}$ is the Laplace transform of $\mathcal{P}$, given by formula $V_\alpha f(x) = \int_0^\infty e^{-\alpha t} P_t f(x) dt$.
2) The *kernel operator,* denoted by V, is the initial operator V_0 of $\mathcal{V}$ (for $\alpha = 0$).
3) The *infinitesimal generator* of $\mathcal{P}$ is the possibly unbounded linear operator $\mathcal{A}$ defined by:

$$\mathcal{A}f = \lim_{t \searrow 0} \frac{P_t f - f}{t} \qquad (1)$$

The domain $D(\mathcal{A})$ is the subspace of $\mathbf{B}(X)$ for which this limit exists.

The following definition is inspired by a condition from the Hille-Yosida theorem (Th. 2.6, Chapter 1 in [10]). A linear operator $\mathcal{A}$ has the Hille-Yosida property if for all $\lambda > 0$, the operator $\lambda I - \mathcal{A}$ has an everywhere defined inverse $R(\lambda, \mathcal{A})$ such that $\|\lambda R(\lambda, \mathcal{A})\| \leq 1$. To say $\lambda\, I - \mathcal{A}$ has an everywhere defined inverse means that the operator $\lambda\, I - \mathcal{A}$ is injective on the domain of $\mathcal{A}$ and that its range is all of X.

We have now all ingredients to introduce an unifying concept for deterministic and stochastic continuous processes.

An *abstract continuous system* (ACS) consists of: (a) a state space X (Polish/analytic); (b) a bounded linear operator $\mathcal{A}$ on $\mathbf{B}(X)$ that is densely defined and has the Hille-Yosida property; (c) an operator semigroup $\mathcal{P} = (P_t)$ on $\mathbf{B}(X)$ such that $\mathcal{A}$ is the infinitesimal generator associated to $\mathcal{P}$.

The Hille-Yosida theorem gives necessary and sufficient conditions for a linear operator to be the generator of a strongly continuous, positive contraction semigroup. It results from the Hille-Yosida theorem that the last component of an ACS is superfluous because it can be derived from the second component. We decided to keep it in the definition motivated by practical reasons. The Hille-Yosida theorem is non-constructive and in the most practical situations the expression of the semigroup is known.

On the state space X of an ACS we can define a *preorder relation* $\prec$ as

$$x \prec y \Longleftrightarrow Vf(y) \leq Vf(x), \forall f \in \mathbf{B}(X), f \geq 0. \tag{2}$$

Now, let see how the framework looks like for a Markov process M. Let $\mathcal{P} = (P_t)_{t>0}$ denote the family of linear *operators* associated to M

$$P_t f(x) = \int f(y) p_t(x, dy) = E_x f(x_t), \forall x \in X \tag{3}$$

where E_x is the expectation w.r.t. P_x. We make the standard assumption that $f(\Delta) = 0$. The Chapman-Kolmogorov property ensures that this family of operators has indeed the semigroup property. This is a strongly continuous semigroup of operators.

To the semigroup $\mathcal{P}$ given by (3), one can associate its operator resolvent $\mathcal{V}$ and its infinitesimal generator $\mathcal{A}$. Conversely, given an operator semigroup $\mathcal{P}$, one can check if it might be associated to a Markov process (for necessary and sufficient conditions to ensure that the semigroup can be interpreted as a semigroup of conditional expectations see Th. 2.2, Chapter 4, [10]).

Assumption 1 Suppose that M is a *transient Markov process,* i.e. there exists a strict positive Borel measurable function q such that Vq is a bounded function. The transience of M means that for any Borel set E in X and for almost all trajectories there exists a finite stopping time t^* such that $x_t \notin E$ for all $t > t^*$ (for more explanations about the transience property see [8]).

Using (2), we can define a preorder relation $\prec_M$ associated to M. Intuitively, $\prec_M$ is the order on the trajectories of M. In particular, if M degenerates in a semi-dynamical system, $\prec_M$ is exactly the order relation on the trajectories.

Now we instantiate the framework with semi-dynamical systems. With every semi-dynamical system ϕ one can associate the *semigroup of operators* $\mathcal{P} = (P_t)_{t>0}$ defined by

$$P_t f(x) = f(\phi(t, x)) \tag{4}$$

for all functions $f \in \mathbf{B}(X_\Delta)$. The standard assumption $f(\Delta) = 0$ is in force.

If in the semigroup formula (4), we take $f = I_A$ with $A \in \mathcal{B}$ (the indicator function of a measurable set A) then $P_t I_A(x) = I_A(\phi(t, x))$, i.e. it takes the value one iff $\phi(t, x) \in A$, otherwise it is equal to zero (see [2] and the references therein, for more properties of the semigroup associated to a semi-dynamical system).

The semigroup formula (4) can be derived as a particular case of (3), taking the transition probabilities $p_t(x, \cdot) = \delta_{\phi(t,x)}(\cdot), t \geq 0$ where $\delta_{\phi(t,x)}$ is the Dirac distribution corresponding to $\phi(t, x)$. To the semigroup (4), one can associate its resolvent $\mathcal{V}$ and its generator $\mathcal{A}$. If $A \in \mathcal{B}$ then $VI_A(x)$ is exact the Lebesgue measure of those moments of time $t \geq 0$ for which the trajectory Γ_x has a non-empty intersection with A. We denote $x \prec_\phi y$ if there exists $t \in [0, \infty)$ such that $y = \phi(t, x)$. If the system under consideration is transient then $\prec_\phi$ is an *order relation* [2]. This order relation can be characterized using the initial resolvent kernel (Prop. 13 [2]) via (2).

3 Bisimulation in the Presence of Probability and Continuity

In this section we define a bisimulation concept for abstract continuous systems, organized in a category. We further instantiate this category for continuous time, continuous space Markov processes. Frst, we present the general view of the methodology for defining bisimulation for Markov processes. In the remainder of the section, this methodology will be generalized using operator parameterizations of stochastic processes, in order to make it applicable to a general category of Markov processes. The resulting concept of bisimulation will be compared with a concept of bisimulation via open maps (as introduced by Winskel et.a. [12]) for continuous dynamical system by P. Tabuada, G. Pappas et.a. - see [11]) and of bisimulation for different classes of Markov chains.

3.1 Algebraic concepts of bisimulation

For ACS, the open maps definition of bisimulation can not be adapted straightforward. The main problem is how to define the simulation morphisms and the open maps.

In a category, a semi-pullback means that, for any pair of morphisms $\varphi^1 : M^1 \to M$ and $\varphi^2 : M^2 \to M$ (M^1, M^2, M are objects in that category) there exists an object M^0 and morphisms $\pi^i : M^0 \to M^i$ ($i = 1, 2$) such that $\varphi^1 \circ \pi^1 = \varphi^2 \circ \pi^2$.

We develop a concept of *unifying bisimulation* for ACS defined on Polish/analytic spaces, which can be instantiated with the bisimulation defined for different particular classes of Markov processes studied in the literature. A zigzag morphism between two ACS should 'commute' with the infinitesimal operators of the processes considered. Then the bisimulation relation is naturally given via zigzag morphism spans between ACS. Moreover, the category of ACS defined on Polish/analytic spaces with these zigzag morphisms as arrows has semi-pullback. Therefore, the bisimulation relation is an equivalence relation.

We also derive from the above bisimulation for ACS, a notion of bisimulation for (deterministic) semi-dynamical systems. For dynamical systems, we prove that our concept of zigzag morphism and the open map concept, defined in [11], are equivalent.

3.2 A category of abstract continuous systems

We define the category **ACS** of abstract continuous systems, which has objects a countable set of ACS, defined on Polish/analytic spaces, denoted $S^1, S^2, \ldots$ and arrows - zigzag morphisms, which will be defined in the following.

The aim of this subsection is to give an appropriate definition of these *zigzag morphisms* (and of *simulation morphisms*) between such processes, which will allow us to define a general concept of unifying bisimulation in this category.

Let S^1 and S^2 be two objects of **ACS**. The state space of S^1 (resp. S^2) is $X^{(1)}$ (resp. $X^{(2)}$). For any mapping $\psi : X^{(2)} \to X^{(1)}$, we denote by ψ^* the *action* of ψ on the their monoids of bounded measurable functions, i.e. $\psi^* : \mathbf{B}(X^{(1)}) \to \mathbf{B}(X^{(2)})$

given by

$$\psi^* f = f \circ \psi, \forall f \in \mathbf{B}(X^{(1)}) \tag{5}$$

Let $\mathcal{A}^1$ and $\mathcal{A}^2$ the infinitesimal generators of S^1 and S^2, with the domains $D(\mathcal{A}^1)$ and $D(\mathcal{A}^2)$, respectively. The following assumption is essential for defining the arrows in the category **ACS**.

Assumption 2 If $f \in D(\mathcal{A}^1)$ then $\psi^* f \in D(\mathcal{A}^2)$, i.e. the twisted function ψ^* can action between the domains of the generators $\mathcal{A}^1$ and $\mathcal{A}^2$

A simulation morphism between the processes S^2 and S^1 (the process S^1 simulates the process S^2) is a measurable, monotone increasing, continuous application $\psi : X^{(2)} \to X^{(1)}$ such that it satisfies the Assumption 2 and $\mathcal{A}^2 \circ \psi^* \leq \psi^* \circ \mathcal{A}^1$, where $\mathcal{A}^1$ (resp. $\mathcal{A}^2$) is the infinitesimal generator associated to S^1 (resp. S^2) and ψ^* is given by (5).

A surjective simulation morphism ψ between the processes S^2 and S^1 is called zigzag morphism if

$$\mathcal{A}^2 \circ \psi^* = \psi^* \circ \mathcal{A}^1 \tag{6}$$

Using the relationships between generator, operator semigroup and kernel operator (see, for example, [10]), we can prove the following result.

Proposition 1. *A surjective simulation morphism ψ between the processes S^2 and S^1 is a zigzag morphism iff for almost all $t \geq 0$ (i.e. except with a zero Lebesgue measure set of times) the following equality holds*

$$P_t^2 \circ \psi^* = \psi^* \circ P_t^1 \tag{7}$$

where (P_t^1) (resp. (P_t^2)) is the semigroup of operators associated to S^1 (resp. S^2).

Corollary 2. *A surjective simulation morphism ψ between S^2 and S^1 is a zigzag morphism iff for almost all $t \geq 0$ (i.e. except with a zero Lebesgue measure set of times) and for all $E \in \mathbf{B}(X^{(1)})$ and $x^2 \in X^{(2)}$, the following equality holds*

$$p_t^2(x^2, \psi^{-1}(A)) = p_t^1(\psi(x^2), A) \tag{8}$$

where (p_t^1) (resp. (p_t^2)) is the transition probability functions associated to S^1 (resp. S^2).

This corollary illustrates that the simulating process can make all the transitions of the simulated process with the same transition probabilities as in the process being simulated.

The monotony of a zigzag morphism ψ can be derived from the condition satisfied by a zigzag morphism. Roughly speaking, this means that whilst the process S^2 evolves from u to $\psi^{-1}(A)$ ($A \in \mathcal{B}(X^{(1)})$) on a trajectory with a given probability, the process S^1 evolves from $\psi(u)$ to A with the same probability.

3.3 Bisimulation

We consider the category **ACS** defined in the previous section. We define the *bisimulation* between two processes in this category as the existence of a span of zigzag morphisms between them.

Let S^1 and S^2 be two objects in **ACS**. S^1 is bisimilar to S^2 (written $S^1 \sim S^2$) if there exists a span of zigzag morphisms between them, i.e. there exists S^{12} (object in **ACS**) and two zigzag morphisms ψ^1 and ψ^2 as follows: $X^{(1)} \overset{\psi^1}{\leftarrow} X^{12} \overset{\psi^2}{\rightarrow} X^{(2)}$.

Theorem 3. *The category* **ACS** *has semi-pullbacks.*

An immediate corollary of the existence of semi-pullbacks in the category **ACS** is the following.

Proposition 4. *The bisimulation in the category* **ACS** *is an equivalence relation.*

3.4 Particular cases

In this subsection we investigate the cases when all objects of **ACS** have the same type. In the case when all objects are Markov processes we obtain a generalization **GMP** of the category defined in [4]. In the case of Markov models, we say that a Markov process M^1 *simulates* another Markov M^2 if there exists a surjective continuous morphism $\psi : X^2 \rightarrow X^1$ between their state spaces such that each transition probability on X^2 'is matched' by a transition probability on X^1. The meaning of this 'matching' is that for each measurable set $A \subset X^1$ and for each $u \in X^2$ we have $p_t^2(u, \psi^{-1}(A)) \leq p_t^1(\psi(u), A)$, $\forall t \geq 0$ $(*)$, where (p_t^2) and (p_t^1) are the transition functions corresponding to M^2, respectively to M^1. Such a morphism ψ is called *simulation morphism*. The open maps are replaced then by the *zigzag morphisms*, which are simulation morphisms for which the above condition holds with equality.

In the case when all objects are semi-dynamical systems, we obtain a new category **SD**. In fact, **SD** is a full subcategory of **ACS**.

Proposition 5. *A surjective simulation morphism $\psi : X^{(2)} \rightarrow X^{(1)}$ between two semi-dynamical systems ϕ^2 and ϕ^1 is a zigzag morphism if and only if*

$$\int_0^\infty P_t^2(\psi^* f)dt = \int_0^\infty \psi^*[(P_t^1 f)]dt, \forall f \in \mathbf{B}(X^{(1)}),\ f \geq 0, \tag{9}$$

where (P_t^1) and (P_t^2) are the semigroups of kernels associated to ϕ^1 and ϕ^2.

Proposition 6. *The condition (9) is equivalent with*

$$\psi(\phi^2(t,u)) = \phi^1(t,x)\ (m - a.e.\ w.r.t.\ t \geq 0) \tag{10}$$

for all $u \in X^{(2)}$ such that $x = \psi(u)$.

Corollary 7. *If ψ is a zigzag morphism between two semi-dynamical systems ϕ^2 and ϕ^1 then $\psi(\Gamma_u^2) = \Gamma_{\psi u}^1$, except a set of times with Lebesgue measure zero.*

Proposition 8. *For dynamical systems a zigzag morphism is exactly an open map in the sense of [11].*

Therefore, the bisimulation for dynamical systems given in terms of zigzag morphisms is exactly the bisimulation given in terms of open maps [11]. In the light of the Corollary 7, a zigzag morphism ψ between two semi-dynamical systems ϕ^2 and ϕ^1 induces an equivalence relation (bisimulation) on the state of trajectories of ϕ^2 as follows:

Two trajectories Γ_u^2 and Γ_v^2 are equivalent if and only if their initial points are bisimilar, i.e. $\psi u = \psi v$.

3.5 A probabilistic logic

We extend the continuous stochastic logic [9] from Markov chains to ACSs.
Syntax Let $a \in AP$, $p \in \mathbb{Q}$ and $\bowtie \in \{<, \leq, \geq, >\}$. State formulas ϕ are defined by $\phi := \top | a | \neg\phi | \phi \wedge \phi' | P_{\bowtie p}(\psi) | Expr_{\bowtie p}$, where ψ is a path formula constructed by $\psi := \mathbf{X}\phi | \mathbf{X}^{[t,u]}\phi | \phi \cup \phi' | \phi \cup^{[t,u]} \phi'$, for $t, u \in \mathbb{Q}$.
The notion that a state x (or a path ω) satisfies a formula ϕ (or ψ) is denoted by $x \vDash \phi$ (or $\omega \vDash \psi$).
Semantics Given a Markov process $M = (\Omega, \mathcal{F}, \mathcal{F}_t, x_t, \theta_t, P_x)$ and $a \in AP$, the definition of the satisfaction relation $\vDash$ over state-formulas is defined inductively as follows.
$x \vDash \top$ for all $x \in X$, $x \vDash a$ iff $a \in L(x)$, $x \vDash \neg\phi$, iff $x \nvDash \phi$
$x \vDash \phi \wedge \phi'$, iff $x \vDash \phi$ and $x \vDash \phi'$, $x \vDash Expr \bowtie r$ iff $Expr(x) \bowtie r$
$x \vDash P_{\bowtie p}(\psi)$, iff $P_x\{\omega \in \Omega | \omega \vDash \psi\} \bowtie p$

For example, for any $f \in \mathcal{B}^b(X)$ the potential $Vf(x)$ can be used to construct an expression $Expr(x)$, which can be evaluated in x, and then the inequality $Vf(x) \bowtie p$ can be satisfied or not.

A state formula ϕ can be extended to a path formula if that formula is satisfied in each point of the path, i.e. $\omega \vDash \phi$ if and only if $x_t(\omega) \vDash \phi$, for all t for which ω is defined. Then the semantics of the path-formulas can be defined as follows
$\omega \vDash \mathbf{X}\phi$ iff ($\exists t_0 > 0$ s.t. $\omega^{\geq t_0} \vDash \phi$) , $\omega \vDash \mathbf{X}^{[t,u]}\phi$ iff ($\exists t_0 \in [t, u]$ s.t.$\omega_{t_0} \vDash \phi$)
$\omega \vDash \phi \cup \phi'$ iff ($\exists t_0 > 0$ s.t. $\omega_{t_0} \vDash \phi'$ and $\omega^{[0,t_0)} \vDash \phi$)
$\omega \vDash \phi \cup^{[t,u]} \phi'$ iff ($\exists t_0 \in [t, u]$ s.t. $\omega_{t_0} \vDash \phi'$ and $\omega^{[t,t_0)} \vDash \phi$).

Proposition 9. *Two ACSs are bisimilar iff they satisfy the same logic formulae.*

4 Mobile Markovian Systems

In a series of papers (see [7] and the references therein) G. Winskel and coworkers defined a generic model of mobile processes, where each process is a presheaf. We use the category theory notations from [1]. In particular, arrow composition, denoted by ;, is the sequential composition.

In the following we define the concept of bisimulation for presheaves. A *preashef* over a category **P** is a functor from **P** to **Set**, the category of sets and functions. The preasheaves over the same category, together with the natural transformations between them, form a category, denoted $\widehat{\mathbf{P}}$. This construction comes accompanied by the Yoneda lemma [1], which provides a functor $y_{\mathbf{P}} : \mathbf{P} \to \widehat{\mathbf{P}}, y_{\mathbf{P}}(A) = \mathbf{P}(\_, A)$ which fully and faithfully embeds $\mathbf{P}$ into $\widehat{\mathbf{P}}$. Basically the Yoneda lemma ensures a preashef representation for every category $\mathbf{P}$: it can be regarded as a full subcategory of $\widehat{\mathbf{P}}$. The bisimulation of mobile processes is the standard bisimulation from open maps, as introduced in [12].

Now we give the definition of functors preserving open maps. Given two categories, $\mathbf{P}$ and $\mathbf{Q}$, and a functor F between them, an arrow $f : X \to Y$ is called $F-$open if, for every commuting square $\begin{array}{ccc} F(A) & \xrightarrow{\alpha} & X \\ F(g)\downarrow & & \downarrow f \\ F(B) & \xrightarrow[\beta]{} & Y \end{array}$ there is an arrow $\gamma : F(B) \to X$ such that $F(g); \gamma = \alpha$ and $\gamma; f = \beta$. The isomorphisms are $F-$open and the all $F-$open maps form a subcategory. In it is proved [7] that an arrow between presheaves in $\widehat{\mathbf{P}}$ is $\mathbf{P}$-open iff it $y_{\mathbf{P}}$-open. Two presheaves in $\widehat{\mathbf{P}}$ are called $\mathbf{P}$-bisimilar iff there is a span of surjective open maps between them.

An $\mathbf{P}-$indexed category, denoted $\mathbf{Q}^{\mathbf{P}}$, is formed by all functors of the shape $\mathbf{P} \to \mathbf{Q}$. A *profunctor* is a functor of the shape $F : \mathbf{P} \to \widehat{\mathbf{Q}}$. Profunctors compose and form a bicategory (i.e. there is an additional category on arrows), denoted $\mathbf{PR}$.

As a mobile process evolves, the ambient set of channel names may change. These channel names are modelled by the category $\mathbf{I}$ of finite sets (of names) and injective maps between them. To take account of this variability, we have to consider the semantic categories involved as indexed by $\mathbf{I}$. The *object of names* $\mathbf{N}$ is the functor $\mathbf{N} : \mathbf{I} \to \mathbf{PR}$, that sends a set $S \in \mathbf{I}$ to the corresponding discrete category. The category of *abstract continuous systems with names* is $\mathbf{ACS}^{\mathbf{I}.}$.
Now the integrated model is obtained by including the category $\mathbf{ACS}$ in the domain equations that define the basic processes of π-calculus.

$$\mathbf{P} = \mathbf{ACS}^{\mathbf{I}.} \otimes \mathbf{Q}\ , \qquad \mathbf{Q} \cong \mathbf{Q}_{\perp} + \mathbf{Out} + \mathbf{In}$$
$$\mathbf{In} = \mathbf{N} \otimes (\mathbf{N} \twoheadrightarrow \mathbf{Q})_{\perp}\ , \qquad \mathbf{Out} = (\mathbf{N} \otimes \mathbf{N} \otimes \mathbf{Q}_{\perp}) + (\mathbf{N} \otimes (\delta\mathbf{Q})_{\perp})$$

where $\otimes$ and $+$ denote the product, respectively the coproduct.

A method to solve the domain equations is presented in detail in [7]. We briefly describe the meaning of the solutions. The mobile processes are products of π-calculus processes and abstract continuous systems with names. The ACS can communicate values and the names of other channels. Therefore, the communication is first order and deterministic. These types could be combined, for example, with the type subsystem corresponding to the name passing CCS.

Examples of systems that mix continuous behaviours (deterministic or randomised) with software mobility abound. A trivial example is that of people travelling by car or by plane and use a mobile phone. Less trivial, imagine a mobile software that proceeds a secret security check in the pilot cabin.

In [4], the authors have introduced a concept of bisimulation for stochastic hybrid systems (SHS). In [3], it is proved that the executions of an SHS form a Markov process on a Borel space, whose trajectories are right continuous with left limits. This paper proposes a different approach where the system properties are derived from the infinitesimal generator of a continuous process. The mathematical model of an embedded system is in general constructed starting with the differential equation characterizing the evolution of the environment. This differential equation gives rise to the expression of the generator. When probabilities are introduced, the resulted stochastic process is also called in the literature a random dynamical system (RDS). In the context of this paper, an RDS is simply a Markov process, alternatively defined using the associated generator. The expression of the generator is known for large classes of processes [10] including diffusions, Poisson processes, piecewise deterministic Markov processes and so on. In consequence, the concept of bisimulation from this paper is more adequate for these classes of processes.

Summarizing this bisimulation is for controllers that are embedded in complex physical environments and that exhibit mobile communication.

5 Final Remarks

Due to physical environment of embedded systems, it is natural to consider continuous feature representations in the system model. Randomization knows recent intensive applications in modelling and verification of embedded systems. The combination of these two paradigms gives rise to models with new and sophisticated mathematical characteristics that can obscure the understanding of computational concepts. In the current work, we have addressed this issue, by introducing a unifying framework, of abstract continuous systems, for systems with (partially) continuous behaviours, deterministic or stochastic.

Bisimulation is now well understood for discrete probabilistic automata or deterministic hybrid systems , but it is far more complicated for the stochastic embedded systems. In this paper we have developed a unifying notion of bisimulation for different classes of embedded systems including semi-dynamical systems [2] and strong Markov processes defined on Polish/analytic spaces with continuous time, which are non-stationary. We define a category for each class of systems. For the former category, the morphisms are the so-called zigzag morphisms, which are surjective continuous measurable functions between their state spaces which 'commutes' with the infinitesimal generators of the processes considered. Two Markov processes are bisimilar if there exists a span of zigzag morphisms between them.

The category of abstract continuous systems is used in conjunction with a categorical semantics of π-calculus [7] to define systems mixing physical and logical mobility. The cornerstone of this construction is the concept of bisimulation, which must be equivalent with the one derived from open map [12].

The mobile stochastic hybrid systems provide a very general semantic frameworks in which embedded systems can be studied. Examples include sensor networks and air traffic control. Mobility allows system reconfiguration, which, combined with probabilities, provide the basic ingredients for randomized learning. This work puts the grounds for semantics of the most actual issues in ubiquitous computing: the self-* systems (abbreviation for features like reconfigurable, adaptive, learning, self-managed, etc. systems).

The omitted proofs can be found in a larger version of this paper [5].

Bibliography

[1] M. BARR and M., WELLS, *Category Theory for Computing Science*, Prentice Hall, 1990.

[2] G. BUCUR and M. BEZZARGA, *Theorie du Potentiel pour les Systemes Semi-Dynamiques*, Rev. Roumaine Math. Pures Appl., 39(5), (1994) pp. 439-456.

[3] M. L. BUJORIANU and J. LYGEROS, *Toward a general theory of stochastic hybrid systems. Stochastic hybrid systems*, Lecture Notes in Control and Inform. Sci., **337**(2006): 3–30.

[4] M. L. BUJORIANU, J. LYGEROS and M. C. BUJORIANU, *Bisimulation for General Stochastic Hybrid Systems*, In Proc. of *Hybrid Systems: Computation and Control*, LNCS **3414,** 2005, pp. 198-214.

[5] M. L. BUJORIANU and M. C. BUJORIANU, *Towards a Formal Framework for Multidimensional Codesign*, Technical Report TR-CTIT-08-21 Centre for Telematics and Information Technology, University of Twente, 2008, http://eprints.eemcs.utwente.nl/12108/

[6] M. C. BUJORIANU and M. L. BUJORIANU *Towards Hilbertian Formal Methods,* Proc. of ACSD, IEEE Press, 2007.

[7] G.L. CATTANI, I. STARK, and G. WINSKEL, *Presheaf Models for the pi-Calculus*, Proc. of CTCS, Springer LNCS 1290, 1997, pp. 106-126.

[8] K.L. CHUNG, *Green, Brown and Probability*, World Scientific, 1995.

[9] E. E. DOBERKAT, *Stochastic Relations. Foundations for Markov Transition Systems*, Chapman&Hall, 2007.

[10] S. N. ETHIER and T.G. KURTZ, *Markov Processes: Characterization and Convergence*, John Wiley, 1986.

[11] E. HAGHVERDI, P. TABUADA and G.J. PAPPAS, *Open maps, alternating simulations and control synthesis*, Proc. of CONCUR, Springer LNCS 3170, 2004, pp. 466-480.

[12] A. JOYAL, M., NIELSEN and G. WINSKEL, *Bisimulation from Open Maps*, Information and Computation, **127(2),** (1996), pp. 164-185.