DISTRIBUTED STOCHASTIC HYBRID SYSTEMS

Manuela L. Bujorianu [*] **Marius C. Bujorianu** [**]
Savi Maharaj [***]

[*] *Department of Engineering University of Cambridge Cambridge, CB2 1PZ, UK Email: lmb56@eng.cam.ac.uk*
[**] *Computing Laboratory University of Kent Canterbury CT2 7NF, UK Email M.C.Bujorianu@kent.ac.uk*
[***] *Department of Computing Science and Mathematics University of Stirling Stirling, FK9 7LA, UK Email: savi@cs.stir.ac.uk*

Abstract: We consider a theoretical, but very general mathematical model of control systems, namely stochastic hybrid systems. Then we study how to define concurrency for these systems.

Keywords: distributed, stochastic, hybrid systems; process algebra.

1. INTRODUCTION

Almost 90% of control systems consists of software. The complex communication inherent to control system development requires new languages and methodologies. Nowadays, distribution becomes inherent in control systems. In particular, hybrid systems are no exception (Simsek, T. *et al.*, 2003). In this setting, distribution may present additional aspects, in comparison with the case of discrete systems. Distribution may be present in control, execution (which may have stochastic features), communication or parallelism. Moreover, many concepts specific to distributed systems do not always admit a smooth translation into the continuous case. Many times the definition of this concept strongly depends on the continuous mathematics employed to describe the systems.

In distributed hybrid systems, interaction is very complex especially because of the continuous/ discrete aspects involved. Recent developments in computer science and hybrid systems have identified the need for proper treatment of interaction in continuous systems. For embedded systems CSP-based approaches define interaction grounded on a solid tradition. Hybrid extensions of process-algebra (Vereijken, J.J., 1995) are doing the same role for hybrid systems. The limitations of all these approaches consist of the necessity to define the interactions using the intrinsic language of continuous processes. This necessity is generated by the fact that continuous aspects can not be neglected in the context of concurrent hybrid systems. The distributed systems considered until now are discrete concurrent systems enriched with some continuous features.

In this work we make an attempt to define parallelism and communication for stochastic hybrid systems, in a very general setting. We use the model introduced in (Bujorianu, M.L., 2004; Bujorianu, M.L. and Lygeros, J., 2004*a*) (and further developed in (Bujorianu, M.L. and Lygeros, J., 2004*b*)), namely GSHS (general stochastic hybrid systems). It generalizes the most used models of stochastic hybrid systems used in control engineering. The stochastic features of the model make this attempt difficult. From a computer science perspective our approach uses a mixture of ACP (Algebra of Communicating Processes) developed by Bergstra and Klop (Baeten, J.C.M and Weijland, W.P., 1990) and Robin Miner's CCS techniques. Parallelism is introduced in an axiomatic manner as in the ACP tradition. Communication is based on dually labelled transitions (where duality means sent/received transitions or active/passive transitions). Communication takes place as a handshake between dually labelled transitions.

We introduce the concept of *distributed stochastic hybrid systems* (DSHS) as an automata formalism for compositional specifications of GSHS. A DSHS can be thought of as an automaton representation of a GSHS, with an extra possibility to interact with other processes via so-called *passive transitions* (which are *discrete* transitions). This concept is a generalization of the so-called communicating piecewise deterministic Markov processes developed in (Strubbe, S.N. *et al.*, 2003), because the underlying model used is more general.

Section 2 starts by motivating this work (subsection 2.1), then the DSHS model is formally defined (subsection 2.2). After that, the parallel operator and the communication between DSHS models are defined in subsection 2.3. The partial conclusions of this experiment are sketched in the last section.

2. DISTRIBUTED STOCHASTIC HYBRID SYSTEMS

2.1 Motivation

Hybrid systems are used as a paradigm for modelling embedded systems, recorded in Air Traffic Management (ATM), with safety critical performance requirements. Embedded systems of this type have to operate in an uncertain and often adversarial environment. Stochastic analysis and control of hybrid systems is therefore essential to study and improve the performance of ATM systems in the presence of uncertainty.

The problem of safety analysis is addressed from the perspective of the current centralized ATM systems, where aircraft are prescribed to follow certain flight plans, and all flights are controlled by an Air Traffic Controller (ATC) from gate to gate. In the context of ATM, different safety relevant operation cases might occur as follows: vertical crossings; overtake manoeuvres in unmanaged airspace; ATC sector transitions; missed approaches (see (Pola, G. *et al.*, 2003) for a detailed presentation), aircraft-to-aircraft conflict and aircraft-to-airspace conflict. For example, the ATCs are responsible for maintaining a sufficiently large distance between aircraft to avoid dangerous situations and ultimately collisions, by issuing trajectory specifications to the pilots. Separation assurance forms a major part of the current ATC workload. If the level of automation in the ATM process increases, some of the separation assurance tasks can be transferred to the automated system. One approach for doing this is to rely on conflict detection and resolution (CDR) strategies to assist ATC. These strategies try to predict the trajectories of aircraft within managed airspace, analyse these trajectories in order to decide if there is a substantial possibility of loss of separation (conflict detection) and, if there is, issue advisories to the ATC and/ or pilots on how to resolve the problem (conflict resolution).

The model for predicticting the aircraft future position should incorporate the information on the aircraft flight plan, the aircraft dynamics, the flight management systems. Each aircraft has to follow a flight plan, which typically consists of airways (straight lines between given way points traveled at constant speed). The aircraft actual motion might deviate from the planned motion because of different sources of uncertainty. We assume that wind is the main source of uncertainty on the aircraft actual dynamics. The hybrid nature of the model is due to the change in the dynamics when a way-point is reached. The stochastic component is due to the wind described by a random field, which is used to model the spatial correlation between the wind perturbation to the aircraft motion.

The use of GSHS to describe the aircraft dynamics has some advantages. Explicitly, because these models allow stochastic features both in the continuous dynamics and in the discrete transitions, they can be instantiated in different ways. GSHS, as a 'unifying' model, encompasses almost all the models already proposed to deal with different safety critical situations recorded in ATM (piecewise deterministic Markov Processes, switching diffusions, etc. (See (Pola, G. *et al.*, 2003) for an overview). Therefore, techniques and tools specific to GSHS might be employed in order to deal with all the cases of interest. The conclusion is that these safety critical situations can be treated unitary in the modelling framework of GSHS.

In real life, distribution is present in many various ways. Finding its adequate definition in a specific context is often a difficult task. In ATM systems one can easily discover that centralised control coexists with many (semi) autonomic behaviours. This situation is not very common for discrete systems. Moreover, what can be logically seen as centralised control is *distributed* from a physical point of view. In example analysis, where space aspects are important, one might need a distributed model. Simultaneous executions in ATM are, obviously everywhere. From a continuous mathematics viewpoint parallelism (simultaneous executions) can be easily modelled essentially because of the lack of interaction. Basically, the Cartesian product of two (continuous) processes can be interpreted as their simultaneous execution. On the other hand, the interaction is an important feature of these systems. This feature means possible collaboration of multiple agents (both human and non-human). Pilots talk to controllers and sometimes one with each other. Embedded controllers usually control components that interact and are aware of each other existence.

Due to the distributed nature of air traffic management, even advanced engineering design approaches have proven to fall short. It is quite relevant to know that in the USA this has already led to the involvement of distributed control theorists and stochastic analysis in air traffic management studies.

Taking in account the above considerations about the features of the ATM systems, a further development which we make in this paper is to enrich the GSHS model with two capabilities: compositionality and communication. The result is what we have called *distributed stochastic hybrid systems* (DSHS). In ATM , these models can illustrate architectures for handling conflicts between multiple aircraft, while maintaining the situational awareness of human operators. The DSHS modelling, designed on the skeleton of GSHS, constitutes the appropriate framework within which the CDR problems can be easily solved.

2.2 Model Description

In this section we introduce the DSHS formalism. First we formally define its structure and after we give an algorithm which describes its executions.

Definition 1. A DSHS, denoted by $\mathcal{DH}$, is a collection $((Q, d, m, \mathcal{X}), (f, \sigma), L, A, P)$ where

(i) $(Q, d, m, \mathcal{X})$ describes the state space, which is countable union of open sets from an euclidean space (modes), each one corresponding to a discrete location. Note that the dimension of embedding euclidean space might be different for different locations.

(ii) (f, σ) gives the continuous dynamics between jumps of the continuous state within the locations.

(iii) L is the set of labels.

(iv) A are the set of *active transitions.* These are the union of the boundary-hit transitions B and the spontaneous transitions S. The boundary-hit transitions B depend on the transition-choice function C.

Formally, the elements of (i) are defined as follows:

- Q is a countable set of locations.
- $d : Q \to \mathbb{N}$ is a map giving for each location the dimension of the continuous state space in that location.
- $m : Q \to \mathbb{N}$ is a map giving the dimension of the Weiner processes that govern the evolution of the continuous state.
- $\mathcal{X} : Q \to \mathbb{R}^{d(\cdot)}$ maps each $q \in Q$ into an open subset $\mathcal{X}(q) = X^q$ of $\mathbb{R}^{d(q)}$; this means that for each $q \in Q$, X^q is the *mode* (the *invariant* set) associated to the location q. Let us denote by X the whole space, $X = \cup\{(q, X^q)|q \in Q\}$. We also define the boundary set $\partial X^q := \overline{X^q} \backslash X^q$ of X^q and the boundary of the whole space $\partial X = \cup\{(q, \partial X^q)|q \in Q\}$.

The continuous motion parameters from (ii) are given as follows:

- $f : X \to \mathbb{R}^{d(\cdot)}$ is a vector field and $\sigma : X \to \mathbb{R}^{d(\cdot)\times m(\cdot)}$ is a $X^{(\cdot)}$-valued matrix. For all $q \in Q$, the functions $f^q : X^q \to \mathbb{R}^{d(q)}$ and $\sigma^q : X^q \to \mathbb{R}^{d(q)\times m(q)}$ are bounded and Lipschitz continuous and the continuous motions is governed by the following stochastic differential equation (SDE):

  $$dx(t) = f^q(x(t))dt + \sigma^q(x(t))dW_t$$

  where $(W_t, t \geq 0)$ is an $m(q)$-dimensional standard Wiener process in a complete probability space.

The active transitions of (iii) are given as follows:

- B is the set of boundary-hit transitions (forced transitions). Each element $b \in B$ is a quadruple (q, l, q', R_b) where q is the origin location, l is the label of the jump, q' is the target location, and R_b is the reset map of the jump, i.e. for each $x \in \partial X^q$ with $C(b, q, x) > 0$ (see next item) and for all Borel sets A of $X^{q'}$ the quantity $R_b(x, A)$ is the probability to jump in the set A when the transition b is taken from the boundary state x.
- The function $C : B \times Q \times \partial X \to [0, 1]$ is defined such that for all $q \in Q$, all $x \in \partial X^q$, and all $b \in B$, which are outgoing transitions of q, the quantity $C(b, q, x)$ is the probability of executing a boundary-hit transition b. In rest, C takes the zero value. Moreover, $\sum\limits_{b\in B_{q\to}} C(b, l, x) = 1$ where $B_{q\to}$ is the set of all elements of B that are outgoing transitions of q.
- S is the set of spontaneous transitions. Each element $s \in S$ is a pentuple (q, l, q', R_s, λ), where q is the origin location, l is the label of the jump, q' is the target location, R_s is the

reset map of the jump, and λ is the jump rate (it determines the rate of process jumping).

The passive transitions of (iv) are given as follows:

- P is the set of passive transitions. Each element $p \in P$ is a quadruple (q, l, q', R_p), where q is the origin location, l is the label of the jump, q' is the target location, and R_p is the reset map of the jump. The passive transitions play a role in communication with the outside world.

A *general stochastic hybrid system* (GSHS) can be defined in a similar way as DSHS. The only difference is that the discrete transitions do not have labels and there is no choice function defined on the boundary.

A realization of a DSHS given by the definition 1, generates a stochastic process. The above remark and the structure of a DSHS assure that this process is a GSHS. We will refer to this process as the associated GSHS to the given DSHS. In this context, a DSHS execution can be defined as a sample path of this stochastic process. For the generation of the DSHS executions we assume that no communication takes place, therefore the passive transitions do not play any role in the generation of executions.

To eliminate pathological solutions that take an infinite number of discrete transitions in a finite amount of time (known as Zeno solutions) we assume that for each location $q \in Q$ the number of outgoing transitions is finite.

Execution of a DSHS

We assume that an initial hybrid state (q_0, x_0) is given. The continuous dynamics in the mode X^{q_0} is determined by the stochastic differential equation (SDE) with q replaced by q_0. Let ω_{q_0} be an arbitrary diffusion sample path starting in x_0. Suppose that ω_{q_0} reaches the boundary ∂X^{q_0} at time τ_b and suppose that location q_0 has n_0 outgoing spontaneous transitions. During the continuous motion, for every outgoing spontaneous transition, a Poisson type process is activated. This can generate a jump to another hybrid state. The probability density functions of these processes is equal to $\lambda_i(x_t(\omega_{q_0}))\exp(-\int_0^t \lambda_i(x_t(\omega_{q_0})))$, where λ_i is the jump rate of the i-th outgoing spontaneous transition s_i. Let define $\tau_{q_0}(\omega_{q_0}) := \min\limits_{i=1..n_0}$ $\tau_i(\omega_{q_0})$ where τ_i is the jump time corresponding to the i-th outgoing spontaneous transition s_i (given by the Poisson probability distribution).

There are two possibilities: 1. If $\tau_b < \tau_{q_0}$, the boundary is reached before any spontaneous transition is about to be executed, which means that a forced transition is executed at time τ_b; 2. If $\tau_{q_0} < \tau_b$, the spontaneous transition corresponding to τ_{q_0} causes a jump at time τ_{q_0} before the boundary is reached.

Remark 1. The above discussions show that the first jump time (corresponding to the diffusion path ω_{q_0}) is a minimum of $n_0 + 1$ stopping times (the first boundary hitting time and the stopping times given by the Poisson probability distributions corresponding to the n_0 outgoing spontaneous transitions from X^{q_0}).

A *boundary-hit transition* at time τ_b from the boundary state $x_{\tau_b}(\omega_{q_0}) \in \partial X^{q_0}$ is executed as follows. It could be the case that multiple boundary-hit transitions are active in state $x_{\tau_b}(\omega_{q_0})$, therefore we use the choice function C to choose one of the active transitions. A transition $b \in B_{q_0 \to}$ is taken according to the probability measure determined by $C(\cdot, q_0, x_{\tau_b})$. Then $b = (q_{0,}l_b, q'_b, R_b)$ is the transition which takes place. The post-jump location is q'_b and the continuous state after the jump is $x' \in X^{q'_b}$ which is drawn according to the reset measure $R_b(\cdot, x_{\tau_b})$. From the new hybrid state (q'_b, x') at time τ_b, the above recipe can be repeated to continue the execution.

A *spontaneous transitions* $s = (q_{0,}l_s, q'_s, R_s)$ at time τ_{q_0} from the continuous state $x_{\tau_{q_0}}(\omega_{q_0}) \in X^{q_0}$ is taken as follows. The target location is q'_s is governed by the probability measure $R_s(x_{\tau_{q_0}}(\omega_{q_0}), \cdot)$. Starting with the new location (q'_s, x') we can repeat the recipe given above to continue the execution.

The executions of the DSHS can be thought of as being generated by the following algorithm.

Algorithm. Generation of DSHS Executions.

```
set $T = 0$
select $X$-valued random variable $\widehat{x}$
repeat
set $q = \mathcal{X}^{-1}(\widehat{x})$, $n_q$ the number of spontaneous transitions from $X^q$
set $x_t$ as solution of (SDE) with initial condition equal to $\widehat{x}$
select $\omega_q$ a sample path for the process $(x_t)$ with the start point $\widehat{x}$
select $\mathbb{R}^+$-valued random variable $\widehat{S}$ such that
    $\widehat{S}(\omega_q) = \min(\tau_1(\omega_q), ...\tau_{n_q}(\omega_q), \tau_b(\omega_q))$ where
    $\tau_i$ is jump time of the $i$-th outgoing spont. tr.
    $\tau_b$ is first hitting time of boundary $\partial X^q$
select the transition $t = q \rightsquigarrow q'$ associated to $\widehat{S}$
select $X^{q'}$-valued r.v. $\widehat{x}$ according to $R_t(., x_{\widehat{S}})$
```

```
**set** $T = T + \widehat{S}$
**until** true
```

2.3 Parallelism and Communication of DSHS

In this section we introduce a composition operator || on the set of DSHS. In the DSHS framework, communication takes places by means of the passive transitions. The execution of an active transition is always independent on the environment. In a context with two composed DSHS, one of DSHS can execute a passive transition with label l if and only if at the same time the other DSHS executes an active event with label l.

Notations. We write $q \overset{l,R}{\mapsto} q'$, $q \overset{l,R}{\rightarrow} q'$, $q \overset{l,R,\lambda}{\rightsquigarrow} q'$ to denote the existence of respectively boundary-hit, passive and spontaneous transitions from q to q' with label l, reset map R and, in the case of spontaneous transition, jump rate λ.

<u>Parallelism</u>

The parallel composition of two DSHS is defined as follows:

Definition 2. Given two given DSHS

$\mathcal{DH}_i = ((Q_i, d_i, m_i, \mathcal{X}_i), (f_i, \sigma_i), L_i, A_i, P_i)$, $i = 1, 2$; the parallel composition $\mathcal{DH} = \mathcal{DH}_1 || \mathcal{DH}_2$ is the collection $((Q, d, m, \mathcal{X}, f, \sigma), L, B, C, S, P)$ whose components are defined as following:

(1) $Q = Q_1 \times Q_2$;
(2) $d : Q \to \mathbb{N}$ such that $d(q_1, q_2) = d_1(q_1) + d_2(q_2)$;
(3) $m = m_1 + m_2$;
(4) $\mathcal{X} : Q \to \mathbb{R}^{d(\cdot)}$ such that $\mathcal{X}(q_1, q_2) = \mathcal{X}_1(q_1) \times \mathcal{X}_2(q_2)$;
(5) $f^{(q_1,q_2)} = \begin{pmatrix} f^{q_1} \\ f^{q_2} \end{pmatrix}$ and $\sigma^{(q_1,q_2)} = \begin{pmatrix} \sigma^{q_1} \\ \sigma^{q_2} \end{pmatrix}$;
(6) $b \in B$, $s \in S$ and $p \in P$ if b, m and p can be derived from the rules r1 till r6 defined below, and r1' till r6' which are the mirrored versions of r1 till r6.

$$\text{r1.}\ \frac{q_1 \overset{l,R_1}{\mapsto} q_1', q_2 \overset{l}{\nrightarrow}}{(q_1, q_2) \overset{l,R}{\mapsto} (q_1', q_2)} \qquad \text{r2.}\ \frac{q_1 \overset{l,R_1}{\mapsto} q_1', q_2 \overset{l,R_2}{\longrightarrow} q_2'}{(q_1, q_2) \overset{l,R}{\mapsto} (q_1', q_2')}$$

$$\text{r3.}\ \frac{q_1 \overset{l,R_1,\lambda_1}{\rightsquigarrow} q_1', q_2 \overset{l}{\nrightarrow}}{(q_1, q_2) \overset{l,R,\lambda}{\rightsquigarrow} (q_1', q_2)} \qquad \text{r4.}\ \frac{q_1 \overset{l,R_1,\lambda_1}{\rightsquigarrow} q_1', q_2 \overset{l,R_2}{\longrightarrow} q_2'}{(q_1, q_2) \overset{l,R,\lambda}{\rightsquigarrow} (q_1', q_2')}$$

$$\text{r5.}\ \frac{q_1 \overset{l,R_1}{\longrightarrow} q_1', q_2 \overset{l}{\nrightarrow}}{(q_1, q_2) \overset{l,R}{\longrightarrow} (q_1', q_2)} \qquad \text{r6.}\ \frac{q_1 \overset{l,R_1}{\longrightarrow} q_1', q_2 \overset{l,R_2}{\longrightarrow} q_2'}{(q_1, q_2) \overset{l,R}{\longrightarrow} (q_1', q_2')}$$

(7) R, the reset map is given as the product measure, namely in case r1, r3, r5

$$R((x_1, x_2), \cdot) = R_1(x_1, \cdot) \otimes 1_{x_2}$$

and in the case r2, r4, r6

$$R((x_1, x_2), \cdot) = R_1(x_1, \cdot) \otimes R_2(x_2, \cdot)$$

where $x_1 \in \partial X^{q_1}$ in case r1, r2, and $x_1 \in X^{q_1}$ (for the cases left) and $x_2 \in X^{q_1'}$;
(8) the transition map λ for cases r3 and r4 is given by

$$\lambda(x_1, x_2) = \lambda_1(x_1)$$

for all $x_1 \in X^{q_1}$ and $x_2 \in X^{q_1'}$;
(9) the choice function C should be specified for any $q = (q_1, q_2) \in Q$ and any $b : (q_1, q_2) \mapsto (q_1', q_2') \in B$. It is clear that b has been derived from one of the following cases:

$$\begin{aligned} &\text{c1}: \ q_1 \overset{b_{11}}{\mapsto} q_1', q_2 \nrightarrow, q_2 = q_2', \\ &\text{c2}: \ q_1 \overset{b_{12}}{\mapsto} q_1', q_2 \rightarrow q_2', \\ &\text{c3}: \ q_2 \overset{b_{23}}{\mapsto} q_2', q_1 \nrightarrow, q_1 = q_1', \\ &\text{c4}: \ q_2 \overset{b_{24}}{\mapsto} q_2', q_1 \rightarrow q_1' \end{aligned}$$

Then for all $(x_1, x_2) \in \partial\mathcal{X}(q)$ the function is defined as

$$C(b, a, (x_1, x_2)) = \begin{cases} C_1(b_{1i}, q_1, x_1); \ \text{if } \alpha, (ci)_{i=1,2} \\ C_2(b_{2i}, q_2, x_2); \ \text{if } \beta, (ci)_{i=3,4} \\ undefined; \ \text{if } \gamma \end{cases}$$

$\alpha \Leftrightarrow (x_1, x_2) \in \partial X \times X^{q_2}$; $\beta \Leftrightarrow (x_1, x_2) \in X^{q_1} \times \partial X^{q_2}$; $\gamma \Leftrightarrow (x_1, x_2) \in \partial X^{q_1} \times \partial X^{q_2}$.

and C takes the zero value in rest.

<u>Communication</u>

The meaning of the parallel composition of two DSHS can be express as follows.

- If one agent is able to execute an active event and the other agent does not have a matching passive event, then the active agent executes the transition and while the second agent stays in the same location (rules r1, r1', r3, r3').
- If contrary, the first agent can execute an active transition and the second agent has a matching passive transition, then both agents execute respectively the active and passive transition at the same time (rules r2, r2', r4, r4').
- If the first agent has a passive transition with label l and the second agent has no passive transition with label l, then the composed system has a passive transition with label l outgoing from the joint location, which gives the possibility to interact with other systems, in an other composition context (rules r5, r5').
- If both agents have a passive transition with the same label, then the composed system also has a passive transition with this label. The implication of this fact is that both agents can execute the passive transitions at the same time in another composition context where a third agent executes an

active transition with the same label (rules r6, r6').

A state (x_1, x_2) is called a *double boundary state* if both x_1 and x_2 are boundary points.

The choice function C is undefined for the double boundary states. In the composed system execution, these points play a role only when the two components reach their boundary at exactly the same time τ_b. It is not clear what to do in such a situation. Two boundary-hit transitions should be executed at the same time. It might exist the possibility that the two transitions have different labels and then a simultaneous execution of these transitions gives problems from compositionality point of view: If a third agent has both labels available in passive transitions, which passive transition should be chosen? However, for many composed systems these problems would not be present because the probability that two separate agents reach their boundary at exactly the same time is zero. We leave the choice of what to do in double boundary states open and we say that the composed DSHS is undefined on the double boundary states.

Because the reset maps are not defined for the double boundary states, the boundary-hit transitions are not defined for these states.

The choice function C from the definition 2, being defined by using the choice functions C_1, C_2 of the components, meets the conditions imposed to the choice function from the definition 1 of DSHS.

From the definition 2 and the remarks 2.3, 2.3, it results that the parallel composition $\mathcal{DH}$ of the two DSHS $\mathcal{DH}_1$ and $\mathcal{DH}_2$ is a DSHS which is undefined on double boundary states. This underspecification can not be seen as a shortcoming of the parallel composition. It follows naturally from the fact that a parallel execution of two DSHS brings forth cases where a choice has to be made. This choice can be made by the parallel operator (when the composition would result in a DSHS), but it might be better to leave this choice open because it is possible to depend on the situation or the kind of application which choice is more appropriate.

3. CONCLUSIONS

In this paper, we present a distributed version of one of the most general unifying models of hybrid systems.

Distribution in this paper means parallelism and communication. A parallel composition operator for DSHS is defined à la ACP. Communication is defined in CCS style and it takes place via some discrete transitions called passive transitions. The communication takes place via some *discrete* transitions.

From the communication point of view, the next step will be to introduce the communication for the continuous part of the DSHS. However, this further development is not straightforward. Introducing the possibility for two agents to have communication between their continuous and discrete parts, the resulted composed agent might not be an DSHS. This is why we have to find out at least sufficient conditions to ensure that the parallel composition of two agents (when they have 'continuous and discrete communication') is well defined.

REFERENCES

Bujorianu, M.L. and Lygeros, J. (2004*a*). General stochastic hybrid systems. In: *IEEE Mediterranean Conference on Control and Automation, MED'04*.

Bujorianu, M.L. and Lygeros, J. (2004*b*). General stochastic hybrid systems: Modelling and optimal control. In: *43thIEEE Conference in Decision and Control, CDC'04*.

Baeten, J.C.M and Weijland, W.P. (1990). *Process Algebra*. Cambridge University Press.

Bujorianu, M.L. (2004). Extended stochastic hybrid systems. In: *Hybrid Systems: Computation and Control* (R. Alur and G. Pappas, Eds.). pp. 234–249. Number 2993 In: *LNCS*. Springer Verlag.

Pola, G., Bujorianu, M.L., Lygeros, J. and Di Benedetto, M. D. (2003). Stochastic hybrid models: An overview with applications to air traffic management. In: *Conference on Analysis and Design of Hybrid System*.

Simsek, T., Sousa, G. and Varaiya, P. (2003). Communication and control of distributed hybrid systems. In: *American Control Conference*.

Strubbe, S.N., Julius, A.A. and van der Schaft, A. J. (2003). Communicating piecewise deterministic markov processes. In: *Conference on Analysis and Design of Hybrid System*.

Vereijken, J.J. (1995). A process algebra for hybrid systems. In: *The Second European Workshop on Real-Time and Hybrid Systems*.