Modeling Systems by Probabilistic Process Algebra: An Event Structures Approach

Joost-Pieter Katoen \(^a\) and Rom Langerak \(^a\) and Diego Latella \(^b\)

\(^a\)Department of Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands, e-mail: \{katoen, langerak\}@cs.utwente.nl

\(^b\)CNR, Ist. CNUCE, Via S. Maria 36, 56126 Pisa, Italy, e-mail: latella@fdt.cnuce.cnr.it

Performance and reliability analysis of distributed systems based on formal specifications is an important and widely recognized issue. This paper treats a probabilistic version of (a subset of) the process algebra LOTOS. It incorporates a probabilistic choice assigning a probability of occurrence to each of its alternatives. Opposed to the traditional interleaving semantics used for existing probabilistic process algebras the presented language is based on a true concurrency semantics. This enables us to distinguish between non-determinism and parallelism, to reduce the state explosion problem and, moreover, to analyse part of the system without considering other (irrelevant) parts. In this paper the language is presented and the formal semantics is defined by using an extension of bundle event structures. A short example illustrates the novelties of the language and links the language to stochastic analysis based on semi-Markov chains.

**Keyword Codes**: C.2.4; C.4; D.2.1; F.3.2; F.4.3

**Keywords**: Distributed Systems; Event Structures; LOTOS; Performance of Systems; Requirements/Specification; Semantics

1. INTRODUCTION

The process algebra LOTOS is an ISO standard for the specification of distributed systems [7]. In this paper we extend (a subset of) this language with a probabilistic choice operator in order to model the stochastic behaviour of systems, which is important for performance and reliability analysis.

The semantics we propose in this paper is a probabilistic extension of an event structures variant, namely *bundle event structures*, introduced in [10, 11]; this extension is called *probabilistic BES* \(^2\).

Being a “true concurrency” model, BES distinguish between parallelism and non-determinism. This provides a good basis for probabilistic reasoning since probabilistic information is usually associated to non-determinism and not to parallelism. Moreover, the direct representation of parallelism leads to state spaces of reasonable size. This reduces the state explosion problem which is typical for other concurrency models and which makes stochastic analysis prohibitive. Finally, BES allow for local analysis: i.e. if one

---

\(^*\)This work has been partially funded by the CNR-NATO Advanced Fellowships Program.

\(^2\)In the sequel, the following abbreviations will be used: bes for *bundle event structure*, BES for *bundle event structures*, πbes for *probabilistic bes*, and πBES for *probabilistic BES*. 
is interested in analyzing only part of a given system, it is relatively easy to isolate the information related to that part from the whole bundle event structure.

To our knowledge, this is the first proposal of a Probabilistic Process Algebra (PPA) with a true concurrency semantics. Most PPAs proposed in literature are based either on algebras for clock-synchronous processes, (e.g. [4]), or on interleaving semantics (e.g. [20, 5]). Synchronous algebras are suitable for modeling hardware components or synchronous systems like systolic arrays etc. However, there are many systems, like communication protocols, that cannot be adequately described by these algebras. On the other hand, the interleaving approach does not distinguish parallelism from non-determinism, suffers from the state explosion problem, and lacks locality.

The language we propose, called $L$ in the sequel, is a simple extension of a subset of Basic LOTOS [1]. It includes action-prefix ($\mu; B$), non-deterministic choice ($B_1 \parallel B_2$), and parallel composition ($B_1 || G || B_2$) without value passing. $L$ is enriched with a binary probabilistic choice operator $[\parallel p]$. Under the assumption that the choice between $B_1$ and $B_2$ cannot be influenced by the external environment, $B_1 [\parallel p] B_2$ non-deterministically chooses to behave either like $B_1$ or $B_2$, the probability to behave like $B_1$ (respectively $B_2$) being $p$ (respectively $1-p$).

In our language, the above assumption is met by means of syntactical constraints on $B_1$ and $B_2$ and allows us to think of $B_1 [\parallel p] B_2$ as a stochastic experiment (see Section 2).

Other approaches [20, 2, 4, 24, 8, 15, 16, 19, 22, 23] allow the use of probabilities in contexts in which the probabilistic choice can be influenced by the external environment. Such a possibility leads to a more complicated semantics since it must deal with stochastic experiments that are not independent and, consequently, it must allow the dynamic redefinition of the probability space of behaviours depending on the context in which a given behaviour expression is placed. On the other hand, there are many applications for which the use of probabilities in the above contexts is not necessary since the phenomena one usually wants to describe by probabilistic behaviours are typically out of control from the environment (like faults, for instance) [5, 18].

In the present proposal probabilistic choice is restricted to be performed between processes the first action of which are required to be silent moves, denoted by $i$. So, for instance, probabilistic choices like $a; B_1 [\parallel p] a; B_2$ and $a; B_1 [\parallel p] i; B_2$ are not taken into consideration here although their non-probabilistic counterparts express instances of internal non-determinism. The reason for this choice is to keep our model as simple as possible. It is worth reminding that $a; B_1 [\parallel p] a; B_2$ is testing equivalent [17] to $i; a; B_1 [\parallel p] i; a; B_2$ and that $a; B_1 [\parallel p] i; B_2$ is testing equivalent to $i; ((a; B_1) [\parallel p] B_2) [\parallel p] i; B_2$, so that the model we propose is still expressive enough as long as reasoning modulo testing equivalence is acceptable.

This paper is organized as follows. In Section 2 the syntax of $L$ is defined. $\pi$BES are defined in Section 3 and in Section 4 a $\pi$BES-based semantics for $L$ is given. Moreover, in Section 4 it is shown that using BES semantics the correctness relation is identity. That is, the probabilistic bundle event structure for an arbitrary expression $e (e \in L)$ is identical to the (ordinary) bundle event structure associated with the LOTOS counterpart of $e$. Finally, in Section 5 a sample specification is given, on which some stochastic analysis is performed.
2. SYNTAX OF $\mathcal{L}$

We define the set of well-formed expressions by means of an abstract syntax and an extra constraint on this syntax.

**Definition 2.1** Subset of LOTOS

$$B ::= \text{stop} \mid (\mu ; B) \mid (B \parallel B) \mid (B \parallel_p B) \mid (B \parallel[G] B) .$$

With $G \subset Gates$, with $Gates$ the set of LOTOS gate-identifiers, $\mu \in Gates \cup \{i\}$ and probability $p \in (0, 1)$ (that is, $0 < p < 1$). Parentheses are omitted when not necessary.

The idea is that probabilistic choices give rise to stochastic experiments, that is, sets of possible events where each event is assigned a certain probability. As the use of probabilistic choices together with non-deterministic choices leads to alternatives without a probability assigned to them, we want to avoid the intermixing of these two forms of choices. For similar reasons parallel compositions cannot be operands of probabilistic choices. (The effect of these restrictions is reflected in lemmas 4.5 and 4.6, see Section 4.) These ideas are formalized as follows.

**Definition 2.2** Predicate $\mathcal{P}I$

$$\mathcal{P}I : Bex \rightarrow \text{Bool}$$

is defined as follows:

- $\mathcal{P}I(\text{stop}) = \text{false}$
- $\mathcal{P}I(\mu ; B) = \text{false}$
- $\mathcal{P}I(B_1 \parallel B_2) = \text{false}$
- $\mathcal{P}I(B_1 \parallel_p B_2) = \text{true}$
- $\mathcal{P}I(B_1 \parallel[G] B_2) = \mathcal{P}I(B_1) \lor \mathcal{P}I(B_2) .$

Informally, $\mathcal{P}I$ identifies (parallel compositions of) probabilistic choices (possibly with other constructs). As it will be clear from the semantics of $\mathcal{L}$ the initial events of such expressions will be assigned probabilities (see Section 4).

**Definition 2.3** Language $\mathcal{L}$

Expression $B \in \mathcal{L}$ if and only if $B \in Bex$ and

1. $B = B_1 \parallel B_2 \Rightarrow \neg (\mathcal{P}I(B_1) \lor \mathcal{P}I(B_2))$
2. $B = B_1 \parallel_p B_2 \Rightarrow (\exists B'_1, B''_1, q : B_1 = B'_1 \parallel_q B''_1 \lor B_1 = i ; B'_1) \land (\exists B'_2, B''_2, r : B_2 = B'_2 \parallel_r B''_2 \lor B_2 = i ; B'_2) .$

□
3. THE $\pi$BES MODEL

As already mentioned in Section 1 the semantics of $\mathcal{L}$ is based on a true concurrency model and is defined by using an extension to bundle event structures. Bundle event structures [10] consist of labelled events which model the occurrence of actions, together with relations of causality, conflict and independence between events. Causality is a relation between a set of events $X$ (bundle set) and an event $e$. The interpretation is that if $e$ happens, at least one of the events in $X$ must have happened already. In order to make the causal dependencies between events in a system run unambiguous all events in $X$ must be pairwise in conflict, so if $e$ happens, exactly one event in $X$ has happened already. Conflict is a symmetric binary relation between events and the intended meaning is that when events $a$ and $b$ are in conflict, they can never both happen in a single system run. When there is neither a conflict nor causal relation between events the events are said to be independent. This means that if independent events are enabled they can occur in any order or simultaneously.

**Definition 3.1 Bundle event structure**

A bundle event structure $E$ is a 4-tuple $(E, \# , \rightarrow , l)$ with:

- $E$, a set of events
- $\# \subseteq E \times E$, the conflict relation
- $\rightarrow \subseteq 2^E \times E$, the causality relation
- $l : E \rightarrow \text{Act}$, the action-labeling function, where $\text{Act}$ is a set of action labels

such that the following properties hold:

1. $\forall X \subseteq E, e \in E : X \rightarrow e \Rightarrow (\forall e1, e2 \in X : e1 \neq e2 \Rightarrow e1 \# e2)$
2. $\#$ is irreflexive and symmetric .

$\blacksquare$

$\pi$BES are graphically represented in the following way. Events are denoted as dots; near the dot the action label is given. Conflicts are indicated by dotted lines. A bundle $(X, e)$ is indicated by drawing an arrow from each element of $X$ to $e$ and connecting all lines by small lines.

Essentially, a $\pi$bes is a bes where some events are labeled also by probabilities. As mentioned in Section 1 probabilities are assigned to internal events only. See for example Figure 1. In particular, probabilities are associated to those internal events which can be grouped together so as to model a discrete probability space. That is, such events must be grouped into “cluster” of mutually conflicting events. Notice that in a given $\pi$bes there can be more than one cluster as in Figure 1(c,d). Different clusters just represent different (possibly independent, as in Figure 1(c)) stochastic experiments. Notice, finally, that requiring clusters to represent stochastic experiments implies that all the events in any cluster must be “pointed to” by the same bundle sets. Actually, the probability label associated to an event $e$, is the conditional probability of $e$ to happen given that $e$ is enabled. A cluster meaningfully represents a stochastic experiment only if all events in the cluster are enabled when $e$ is enabled. The above concepts are formalized in the following:
Definition 3.2  
Cluster

For \((E, \#, \mapsto, l)\), a cluster is a set \(Q\) of events, \(Q \subseteq E\), satisfying \(\forall e \in Q:\)

1. \(l(e) = i\)
2. \(Q \setminus \{e\} = \{e' | e' \# e\} \land Q \setminus \{e\} \neq \emptyset\)
3. \(\forall e' \in Q, X \subseteq E : X \mapsto e \iff X \mapsto e'.\)

Notice that constraint 2. implies that all distinct events in \(Q\) are in conflict with one another and are not in conflict with events not in \(Q\).

Definition 3.3  
Probabilistic bundle event structure

A probabilistic bundle event structure \((\pi \text{bes})\) is a 5-tuple \((E, \#, \mapsto, l, \pi)\) with \((E, \#, \mapsto, l)\) a bes and \(\pi : E \to (0, 1)\) a partial function, the probability-labeling function, such that for all \(e \in \text{dom}(\pi)\) and \(Q = \{e' | e' \# e\} \cup \{e\}\) we have \(^3:\)

1. \(Q\) is a cluster
2. \(Q \subseteq \text{dom}(\pi)\)
3. \(\sum_{e' \in Q} \pi(e') = 1.\)

Figure 2 shows some BES which are not \(\pi BES\); in particular the structure in Figure 2(a) violates condition 2 of definition 3.2 whereas Figure 2(b) does not fulfill condition 3 of definition 3.2.

\(^3\)For any function \(f : A \to B\), \(\text{dom}(f)\) is defined as \(\{a | \exists b : b = f(a)\}\).
Figure 2. Sample BES that are not $\pi$BES.

Once assigned probability distributions to clusters, the next step is to “reason” about them. This means computing probabilities for the dynamic representations of an event structure, namely configurations [10]. We first of all recall some definitions. Let $\mathcal{E} = (E, \# , \rightarrow, l, \pi)$ be a $\pi$bes.

**Definition 3.4 Proving sequence of $\mathcal{E}$**

A proving sequence of $\mathcal{E}$ is a sequence of distinct events $e_1 \ldots e_n \in E$, satisfying:

1. $\{e_1, \ldots, e_n\}$ is conflict-free, i.e. $\forall e_i, e_j : \neg (e_i \# e_j)$, and
2. $\forall X \subseteq E : X \not\rightarrow e_i \Rightarrow \{ e_1, \ldots, e_{i-1} \} \cap X \neq \emptyset$ for $1 < i \leq n$. 

□

**Definition 3.5 Configuration**

A set $C \subseteq E$ is called a configuration if there is a proving sequence $e_1 \ldots e_n$ such that $C = \{ e_1, \ldots, e_n \}$.

□

The concept of configuration can be thought of as to correspond intuitively to a system run.

It is worth noting that in the general case, $\pi$ being a partial function, the set of all configurations of a $\pi$bes does not generate a random space. That is, there are configurations for which it does not make sense to speak about probabilities. For instance, it does not make sense to speak about the probability of configuration $\{ e_5 \}$ for the $\pi$bes of Figure 3. Also, there are sets of configurations the elements of which must be indistinguishable from the probabilistic point of view. For instance, again with reference to Figure 3, the following configurations are probabilistically indistinguishable:

$c_1 = \{ e_1 \}, c_2 = \{ e_1, e_5 \}, c_3 = \{ e_1, e_3 \}, c_4 = \{ e_1, e_3, e_5 \}$. 

\[ c_1 = \{ e_1 \}, c_2 = \{ e_1, e_5 \}, c_3 = \{ e_1, e_3 \}, c_4 = \{ e_1, e_3, e_5 \} . \]
In other words, whenever it is known that a configuration belonging to the set \( \{ c_1, c_2, c_3, c_4 \} \) has happened (i.e. its events have happened) it does not make sense to reason about the probability that a particular one has happened. On the other hand, all the above configurations share a common feature, viz. the fact that event \( e_1 \) has been chosen; moreover, the probability of choosing \( e_1 \) is \( \frac{1}{3} \) (notice that \( e_1 \) is the only event appearing in the configurations above for which \( \pi \) is defined).

So, the only question which makes sense in this example is “What is the probability of having any configuration which contains \( e_1 \)?” The answer to this question can be put in the following way: \( \mathbb{P}(\{ c_1, c_2, c_3, c_4 \}) = \frac{1}{3} \). Below we will formalize this idea. Let \( \mathcal{E} = (E, \#, \rightarrow, l, \pi) \) be a \( \pi \)bes and \( \mathcal{C}_\mathcal{E} \) be the set of configurations of \( \mathcal{E} \).

**Definition 3.6** Relation \( \Rightarrow \)

For configurations \( C_1, C_2 \in \mathcal{C}_\mathcal{E} \) binary relation \( \Rightarrow \) is defined as

\[
C_1 \Rightarrow C_2 \iff (C_1 \cap \text{dom}(\pi) = C_2 \cap \text{dom}(\pi))
\]

\( \Box \)

**Lemma 3.7** \( \Rightarrow \) is an equivalence relation on configurations.

Let \( [C]_\Rightarrow \) denote the equivalence class of configuration \( C \) under \( \Rightarrow \). That is, \( [C]_\Rightarrow \) contains all configurations of \( \mathcal{E} \) that are equivalent (under \( \Rightarrow \)) to \( C \). An equivalence class \( [C]_\Rightarrow \) is represented by \( C \cap \text{dom}(\pi) \), which is called the stochastic choice of \( [C]_\Rightarrow \).

**Definition 3.8** Probability of configurations

For a set of configurations \( V \) and \( \pi \)bes \( \mathcal{E} \) such that \( V = [C]_\Rightarrow \) for some \( C \in \mathcal{C}_\mathcal{E} \) and \( C \cap \text{dom}(\pi) \neq \emptyset \), the probability of \( V \), denoted \( \mathbb{P}(V) \), is defined by

\[
\mathbb{P}(V) = \prod_{e \in C \cap \text{dom}(\pi)} \pi(e)
\]

\( \Box \)

Stated in words, the probability of a set of configurations is defined for (non-empty) equivalence classes of configurations (under \( \Rightarrow \)) and is equivalent to the probability of the stochastic choice of this equivalence class.
As an example consider the $\pi bes$ of Figure 4(a). The equivalence classes, stochastic choices and probability function of this $\pi bes$ are summarized in Table 1.

Notice that the set of all stochastic choices of $\mathcal{E}$ does not constitute a random space, since the sum of its probabilities may differ from 1. The set of maximal (under set inclusion) stochastic choices of $\mathcal{E}$ constitutes a random space 4. Other (sub-)spaces can be derived from the set of maximal stochastic choices by replacing all stochastic choices containing a given stochastic choice $H$ by $H$ itself. For the $\pi bes$ of Figure 4(a) we have the following two random spaces (see Figure 4(b)): $S_1 = \{H_1, H_2\}$ and $S_2 = \{H_1, H_3, H_4, H_5\}$. Notice that $S_1$ can be obtained from $S_2$, since $H_2 \subseteq H_3, H_2 \subseteq H_4$, and $H_2 \subseteq H_5$.

4. $\pi BES$ SEMANTICS OF $\mathcal{L}$

In this section we give a $\pi BES$ semantics to $\mathcal{L}$. We define a mapping $[\ ]$ that maps each expression $B$ of $\mathcal{L}$ to a $\pi bes$ $[B]$. This function is a straightforward extension of the

\[ [x] = \begin{cases} \{x\} & \text{if } x \in \text{Var} \\ \emptyset & \text{otherwise} \end{cases} \]

\[ [B_1 \cdot B_2] = [B_1] \cdot [B_2] \]

\[ [B_1 + B_2] = [B_1] + [B_2] \]

\[ [\neg B] = \neg [B] \]

4A set $X$ is called maximal w.r.t. set inclusion iff $\forall Y : \neg(X \subseteq Y)$. 

Table 1

<table>
<thead>
<tr>
<th>Equivalence Class</th>
<th>Stochastic Choice</th>
<th>$\mathbb{P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset, {x_7}$</td>
<td>$H_0 = \emptyset$</td>
<td>undefined</td>
</tr>
<tr>
<td>${x_1}, {x_1, x_7}, {x_1, x_3}, {x_1, x_3, x_7}$</td>
<td>$H_1 = {x_1}$</td>
<td>$2/3$</td>
</tr>
<tr>
<td>${x_2}, {x_2, x_7}$</td>
<td>$H_2 = {x_2}$</td>
<td>$1/3$</td>
</tr>
<tr>
<td>${x_2, x_4}, {x_2, x_4, x_7}$</td>
<td>$H_3 = {x_2, x_4}$</td>
<td>$1/6$</td>
</tr>
<tr>
<td>${x_2, x_5}, {x_2, x_5, x_7}$</td>
<td>$H_4 = {x_2, x_5}$</td>
<td>$1/12$</td>
</tr>
<tr>
<td>${x_2, x_6}, {x_2, x_6, x_7}$</td>
<td>$H_5 = {x_2, x_6}$</td>
<td>$1/12$</td>
</tr>
</tbody>
</table>

Figure 4. An example $\pi bes$ (a) and its stochastic choices (b).
BES-semantics for LOTOS given in [10], the only difference from which is the addition of the component dealing with probabilities. First, we extend the definition of $init$ to $\pi BES$.

**Definition 4.1 Set of initial events**

For $\pi bes \mathcal{E} = (E, \#, \rightarrow, l, \pi)$, $init(\mathcal{E})$ is the set $\{ e \in E \mid \neg(\exists X \subseteq E : X \rightarrow e) \}$. □

The definition of the semantics of $stop$, action-prefix, and non-deterministic choice is quite straightforward and is given below. In the sequel let $[B_1] = \mathcal{E}_1 = (E_1, \#, \rightarrow_1, l_1, \pi_1)$ and $[B_2] = \mathcal{E}_2 = (E_2, \#, \rightarrow_2, l_2, \pi_2)$ with $E_1 \cap E_2 = \emptyset$. We suppose there is an infinite universe $E_U$ of events.

**Definition 4.2 stop, action-prefix and non-deterministic choice**

$[stop] = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

$[\mu ; B_1] = (E, \#, \rightarrow, l, \pi_1)$ where

- $E = E_1 \cup \{ e \}$ for some $e \in E_U \setminus E_1$
- $\rightarrow = \rightarrow_1 \cup (\{ e \} \times init(\mathcal{E}_1))$
- $l = l_1 \cup \{ (e, \mu) \}$

$[B_1 \parallel B_2] = (E_1 \cup E_2, \#, \rightarrow_1 \cup \rightarrow_2, l_1 \cup l_2, \pi_1 \cup \pi_2)$ where

- $\# = \#_1 \cup \#_2 \cup (init(\mathcal{E}_1) \times init(\mathcal{E}_2))$. □

Apart from the probability part $\pi$ the semantics of the probabilistic expression $B=B_1 \parallel_p B_2$ is equivalent to the non-deterministic choice. For non-initial events of $B$, $\pi$ is defined as the union of $\pi_1$ and $\pi_2$. For initial events the situation is slightly more complicated. All probabilities of initial events of $B_1$ must be multiplied with $p$ and those of $B_2$ with $1-p$. In order to do so we have to distinguish between events that are assigned a probability within $B_1$ and $B_2$ and those that are not. We thus obtain:

**Definition 4.3 Probabilistic choice**

$[B_1 \parallel_p B_2] = (E_1 \cup E_2, \#, \rightarrow_1 \cup \rightarrow_2, l_1 \cup l_2, \pi)$ where

- $\# = \#_1 \cup \#_2 \cup (init(\mathcal{E}_1) \times init(\mathcal{E}_2))$
- $\pi = (\pi_1 \cup \pi_2) \setminus \{ (e, r) \mid e \in init(\mathcal{E}_1) \cup init(\mathcal{E}_2) \land r \in (0, 1) \}$
  $\cup \{ (e, p) \mid e \in init(\mathcal{E}_1) \land e \notin \text{dom}(\pi_1) \}$
  $\cup \{ (e, pq) \mid e \in init(\mathcal{E}_1) \land (e, q) \in \pi_1 \}$
  $\cup \{ (e, (1-p)) \mid e \in init(\mathcal{E}_2) \land e \notin \text{dom}(\pi_2) \}$
  $\cup \{ (e, (1-p)q) \mid e \in init(\mathcal{E}_2) \land (e, q) \in \pi_2 \}$. □
Finally we define the semantics of the parallel composition operator. We want to define \([B_1 \parallel [G] \parallel B_2] = \mathcal{E}\). The events of \(\mathcal{E}\) are constructed in the following way: event \(e\) that does not need to synchronize (i.e. \(l(e) \notin G\)) is paired with the auxiliary symbol \(*\), and an event that occurs in both processes is paired with all events in the other process that are equally labeled. Thus events are pairs of events of \(B_1\) and \(B_2\), or with one component equal to \(*\). Two events are now put in conflict if any of their components are in conflict, or if different events have a common component different from \(*\). A bundle is introduced such that if we take the projection on the \(i\)-th component \((i=1,2)\) of all events in the bundle we obtain a bundle in \([B_i]\). Finally, events are assigned a probability when one of their components is equal to \(*\) and the other component is assigned a probability in \([B_i]\). Note that no redefinition of the probabilities takes place, as usually in interleaving approaches.

**Definition 4.4** Parallel composition

\([B_1 \parallel [G] \parallel B_2] = (E, \#, \mapsto, l, \pi)\) where

- \(E = (E'_1 \times \{\ast\}) \cup \{\ast\} \times E'_2) \cup \{(e_1, e_2) \in E'_1 \times E'_2 | l(e_1) = l(e_2)\}\), where
  - \(E'_j = \{e \in E_j | l(e) \in G\}, j = 1,2\) (synchronization events)
  - \(E'_j = E_j \setminus E'_j\) (non-synchronizing events)
- \((e_1, e_2)\#, (e'_1, e'_2) \iff (e_1 \# e'_1) \lor (e_2 \# e'_2) \lor (e_1 = e'_1 \neq \ast \land e_2 = e'_2) \lor (e_2 = e'_2 \neq \ast \land e_1 \neq e'_1)\)
- \(X \mapsto (e_1, e_2) \iff \exists X_1 \subseteq E_1 : (X_1 \mapsto e_1 \land X = \{e_j, e_k \in E | e_j \in X_1\}) \lor \exists X_2 \subseteq E_2 : (X_2 \mapsto e_2 \land X = \{e_j, e_k \in E | e_k \in X_2\})\)
- \(l((e_1, e_2)) = \text{if } e_1 = \ast \text{ then } l_2(e_2) \text{ else } l_1(e_1)\)
- \(\pi = \{(e, \ast, p) | (e, p) \in \pi_1\} \cup \{((\ast, e), p) | (e, p) \in \pi_2\}\).

The following behaviour expressions correspond to the \(\pi BES\) of Figure 1 (trailing \textbf{stops} are omitted, and \([\_\_\_]\) denotes \([\emptyset]\)):

(a) \(i; a [/i/3 i] i; b\)

(b) \((i; a [/i/3 i] i) | | c\)

(c) \((i; a [/i/3 i] i; b) | | (i; c [/i/3 i] i; d)\)

(d) \(((i; (a | | c)) [/i/3 i] (i; (a | | d))) [/i/3 i] ((i; (c | | b)) [/i/3 i] (d | | b))) [a, b, c, d]) (a | | b | | c | | d)\)

(e) \((i; i [/i/0 i] i) [/i/3 i] i; a\).

The following lemmas follow from the definition of \([\_\_\_]\) and the syntactical constraints of Section 2. The proofs of the lemmas are omitted in this paper and can be found in [9].

**Lemma 4.5** For all \(B \in \mathcal{L} : \neg \mathcal{P}T(B) \Rightarrow \text{init}(\[B\]) \cap \text{dom}(\pi) = \emptyset\).

**Lemma 4.6** For all \(B_1, B_2 \in \mathcal{L} : B_1 [\_\_\_] B_2 \in \mathcal{L} \Rightarrow \text{init}(\[B_1 [\_\_\_], B_2\])\) is a cluster.
The following lemma states the relationship between expressions of $\mathcal{L}$ and the $\pi BES$ model.

**Lemma 4.7** For all $B \in \mathcal{L}$ : $[B]$ is a $\pi bes$.

For all $B \in \mathcal{L}$ let $B_L$ denote the LOTOS expression obtained from $B$ by systematically replacing all probabilistic choices by non-deterministic choices. Furthermore, let $\mathcal{E}(B_L)$ be the (ordinary) bundle event structure corresponding to $B_L$ as defined in [10]. We now have the following theorem.

**Theorem 4.8** Correctness Theorem

For all $B \in \mathcal{L}$ : $[B] = (E, \#, \rightarrow, l, \pi) \Rightarrow \mathcal{E}(B_L) = (E, \#, \rightarrow, l)$.

**Proof:** Trivial, since removal of the parts concerning $\pi$ in the definition of $[ ]$ leads to the bundle event semantics of LOTOS in [10], and the semantics of $[ ]_p$ is equal to the semantics of $[ ]$ when the part concerning $\pi$ is removed. □

5. AN EXAMPLE

In this section we provide a simple example to illustrate how systems can be described in $\mathcal{L}$ and, more importantly, how the $\pi BES$ semantics can be used as a starting-point for performing a stochastic analysis. The example is rather intuitive in the sense that no formal mapping is given here between the $\pi BES$ and the analysis framework, i.e. Markov chains. Nevertheless, we think it is useful to show, even in a rather informal way, how the model can be used for stochastic analysis.

A few remarks considering the example are in order. First, we do not consider value passing as this is not included in $\mathcal{L}$. As we will see, this simplification will not hamper us. Moreover, we remark that for the sake of brevity trailing stops are omitted from expressions. Furthermore, recursion is used so as to describe the iterative behaviour of processes. We stress that recursion is not (yet) included in our language. However, the extension of $\mathcal{L}$ with tail recursion —the kind of recursion used in the sequel— is rather straightforward. Finally, we want to emphasize that although the example below suggests that our semantical model restricts us to the use of Markovian models for performance analysis this is definitely not the case. We could equally well provide examples for which another type of stochastic analysis is appropriate.

### 5.1. Semi-Markov chains

In this section we briefly consider the basics of discrete semi-Markov processes. A more elaborated presentation can be found in [6, 21].

In contrast to the traditional discrete Markov processes with geometrically distributed residence times, a semi-Markov chain (SMC) allows an arbitrary distribution of residence times. Needless to say, a Markov chain is thus also a semi-Markov chain. An SMC changes states in accordance with a Markov chain but takes a random amount of time between changes. Thus an SMC does not possess the Markovian property that given the present state the future is independent of the past: when predicting the future not only the current state is of importance, but also the amount of time already spent in that state.

---

5 A proposal for dealing with recursion can be found in [14].
plays a role. At the moments of transition the SMC behaves identical to an ordinary Markov chain, in fact, when only considering transition instants (i.e., abstracting from the residence times), we have an “embedded” Markov chain.

Let $P_{ij}$ be the transition probability going from state $S_i$ to $S_j$. Given that the next state is $S_j$, the number of time-units until the transition from $S_i$ to $S_j$ has distribution $F_{ij}$. Then, the probability $R_i(k)$ of being in state $S_i$ for $k$ time-units

$$R_i(k) = \sum_j P_{ij} \cdot F_{ij}(k) . \tag{1}$$

Let $r_i$ denotes the mean of $R_i$. $r_i$ is usually called the (average) residence time in state $S_i$. Now define $T_i$ to be the average number of time-units between successive transitions into state $S_i$, and let $\phi_i$ denote the fraction of time the system is in $S_i$ (on the long run), or, equivalently, the stationary probability of the SMC being in state $S_i$. Then we have

$$\phi_i = \frac{r_i}{T_i} . \tag{2}$$

First observe the SMC abstracting from the residence times, and define $\psi_i$ as the stationary probability of this system being in state $S_i$. Thus $\psi_i$ denotes the (stationary) probability of the system being in state $S_i$ at a transition instant. Stated otherwise, $\psi_i$ is the fraction of instants at which the system is in state $S_i$, considering an infinite amount of transition instants. In order to obtain the fraction of time $\phi_i$ the system is in state $S_i$, the residence times must be taken into account. We have

$$\phi_i = \frac{\psi_i \cdot r_i}{\sum_j \psi_j \cdot r_j} . \tag{3}$$

$\psi_i$ corresponds to the stationary probabilities of the embedded Markov chain and can be calculated in the following way, provided the embedded Markov chain is ergodic:

$$\psi_i = \sum_j P_{ji} \cdot \psi_j \quad \Sigma_i \psi_i = 1 . \tag{4}$$

The SMCs of our examples all have ergodic embedded Markov chains. We use the above equations to calculate $\psi_i$. Using (1) the average recurrence times are obtained and, subsequently they are used in order to obtain expressions for $\phi_i$ (using (3)).

### 5.2. A sample system

One of the main novelties of our model is the locality aspect—if one is interested in analysing only a part of the system it is relatively easy to do so without considering other (irrelevant) parts. To illustrate this we consider the following example.

$Q = (b ; d \mid \{ d \} \mid c ; d)$ ,

$R = (i ; d \mid \rho i ; a ; d)$ , and

$P = s ; (R \mid \{ d \} \mid Q)$ .

It describes the parallel composition of two processes, $Q$ and $R$. $R$ can autonomously choose not to perform action $a$ and this choice has probability $p$; in any case it will then
Figure 5. $\pi$bes of process $P$.

synchronize on $d$ with $Q$, which first executes $b$ and $c$ in parallel, and then $d$. Figure 5 shows the corresponding $\pi$bes of their composite behaviour defined by $P$.

Now consider $X = P | [d] | d; X$. The $\pi$bes corresponding to $X$ is given in Figure 6(a). In fact, Figure 6(a) explicitly shows only the finite part of the $\pi$bes corresponding to “the body” of process $X$. Recursive calls of $X$ must be considered as instantiations of the $\pi$bes shown in the figure replacing the dotted arrows in the obvious way. (In the sequel we shall often speak of “events” whereas —strictly speaking— we should rather speak of “instances” of those events since they belong to different instantiations of the finite part of the $\pi$bes shown in Figure 6(a).) Suppose we are interested in the generation of a

Figure 6. (a) $\pi$bes of $X$, (b) relevant “view” of the system, (c) Markov chain.

events, e.g. the average delay\(^6\) between two $a$'s. We can group relevant events into two

\(^6\)Note that our model does not incorporate any timing information (yet). Therefore, we assume that each bundle lasts for one time-unit and that events are executed instantaneously. It is a subject of further
states (indicated by the two grey “islands” $S_1$ and $S_2$ in Figure 6(b)). Notice that we do not consider events labeled $b$ and $c$. The way in which events are grouped imposes a particular “view” on the system which is characterized by abstracting from details that are irrelevant for the performance analysis one performs. The grouping we have chosen gives rise to the simple Markov chain depicted in Figure 6(c).

Let’s further focus our attention on analysing the Markov chain. For simplicity reasons in this example the residence time in a certain state is taken to be the number of bundles in a state plus one. That is, the number of bundles spent in a state plus the one so as to leave that state. (This is equivalent to the number of events contained in that state.) Note that events themselves are considered to occur instantaneously, and moreover, that state transitions are also considered to be instantaneous.

The system stays in state $S_1$ for 3 time-units when by the next transition, it remains in that state, whereas it stays for a single time-unit in $S_1$ when moving to state $S_2$. Using (1) we obtain

$$r_1 = 1 + 2p \quad \text{and} \quad r_2 = 3 .$$

For the (ergodic) Markov chain we have the following set of equations:

$$\psi_1 = p\psi_1 + \psi_2$$
$$\psi_2 = (1 - p)\psi_1 .$$

Using $\sum \psi_i = 1$ we obtain

$$\psi_1 = \frac{1}{2 - p} , \psi_2 = \frac{1 - p}{2 - p} ,$$

as stationary probabilities of the embedded Markov chain, and for the semi-Markov chain we have, using (3),

$$\phi_1 = \frac{1 + 2p}{4 - p} , \phi_2 = \frac{3(1 - p)}{4 - p} .$$

The average delay between two subsequent $a$-events is equal to the average number of time-units between successive transitions to state $S_2$. According to (2) this is equal to $\frac{\phi_2}{\phi_2}$, or $\frac{1 + p}{1 - p}$. For $p \to 0$ the average delay reaches 4 time-units, which is the optimal case (it lasts at least 4 bundles each time in between two subsequent $a$’s). For $p \to 1$ $a$’s are never generated—and the average delay reaches $\infty$.

In conclusion, we like to emphasize that the average delay between two subsequent $a$’s is analysed without considering the —for our purposes— irrelevant part of process $Q$ (more precisely, events $b$ and $c$). This seems reasonable as only $R$ is involved in generating $a$ events. Here we claim that this ‘locality’ novelty is a direct consequence of the distinction between parallel composition and non-determinism in the $\pi$BES-model.

research to extend our model with timing information, leading to a timed $\pi$BES model.
This paper proposed a model for probabilistic bundle event structures, \( \pi BES \). A language for the specification of probabilistic processes has been introduced and its \( \pi BES \) semantics has been given. It has been shown that the addition of probabilities to BES is “orthogonal” to the BES model itself in the sense that the original semantics of the non-probabilistic version of the language is easily recovered from the \( \pi BES \) semantics by simply removing the probabilities associated to some events. It is worth pointing out here that this is \textit{not} the case when the interleaving semantics approach is followed [13, 12]. In that case, in fact, for any expression \( e \) of \( \mathcal{L} \), the labelled transition system obtained by removing the probability values from the probabilistic transition system [4] of \( e \) and the labelled transition system of the LOTOS counterpart of \( e \) can only proved to be testing equivalent.

It is also argued that, \( \pi BES \) being a \textit{true concurrency} model, it both reduces the \textit{state explosion} problem typical for other models and allows for \textit{local} analysis, thus facilitating stochastic analysis. This conjecture has been shown by means of a simple example. (A stop-and-wait protocol is described in [9].) Several extensions can be foreseen for \( \pi BES \). First of all it is necessary to incorporate recursion. This amounts to a simple extension to the definition for the BES semantics of recursive processes, which associates the latter to infinite BES. In [14] a model for finitely representing a rather interesting class of such infinite BES is proposed. Finite representation of (infinite) \( \pi BES \) is essential for applying Markov theory to probabilistic processes. The formal mapping from such representations to Markov processes is to be defined. Other extensions to the language as well as to the model will allow probabilities to be associated to visible events too. A possibility would be to extend the \( \pi BES \) model towards a \textit{reactive} model of probabilistic processes [24].

Finally, but equally or even more important, comes the extension of \( \pi BES \) with time parameters, both deterministic and probabilistic. In this context it may be interesting to compare (suitable extensions of) \( \pi BES \) with stochastic Petri nets [3]. Moreover, algebraic relations between processes could be defined based on probabilities (and time).

\textbf{Acknowledgements:} The third author wants to thank the University of Twente and in particular the TIOS Group for the collaboration and the nice environment they provided to him.

\textbf{REFERENCES}


