From Hierarchical BIP to Petri Calculus*

Roberto Bruni[1], Hernán Melgratti[2], and Ugo Montanari[1]

[1] Dipartimento di Informatica, Università di Pisa, Italy
[2] Departamento de Computación, FCEyN, Universidad de Buenos Aires - CONICET, Argentina

Abstract. We focus on Hierarchical BIP, an extension of Joseph Sifakis et al's BIP component framework, to provide a semantics-preserving, compositional encoding in the Petri calculus, a recently proposed algebra of stateless connectors and one-position buffers.

1 Introduction

In recent years Joseph Sifakis has successfully pursued a research strand focussed on a component framework called BIP [2], that has also been implemented in a language and a tool-set. BIP is a component framework for constructing systems by superposing three layers of modelling:

1) Behaviour, the lowest level, representing the sequential computation of individual components as automata whose arcs are labelled by sets of ports. The sets of ports of any two different components are disjoint, i.e., each port is uniquely assigned to a component.

2) Interaction, the second level, defining the allowed interactions between components. An interaction is just a set of ports typically of different components.

3) Priority, the top layer, assigning priorities to interactions to enforce scheduling policies, typically with the aim of reducing the size of the state space.

In absence of priorities, the interaction layer of BIP admits the algebraic presentation given in [3] and comparisons with other models have been shown in [7,1,11], see [8] for an overview. In particular, an equivalent version of BIP systems is presented in [7] in terms of a compositional encoding in Petri nets with boundaries [11].

Here we investigate hierarchically structured BIP systems and show that previous correspondence results on ordinay BIP can be extended to deal with Hierarchical BIP (HBIP) as defined in [10]. HBIP systems are possibly formed by the combination of other HBIP systems, each seen as an ordinary component whose ports are its interactions. We exploit the Petri calculus, a calculus of stateful connectors introduced in [11], to encode in a compositional way HBIP

* Research supported by European FET-IST-257414 Integrated Project ASCENS, Progetto MIUR PRIN CINA Prot. 2010LHT4KM, ANPCyT Project BID-PICT-2008-00319.

systems while flattening them. Notably the encoding of components and of interactions can be given separately in the Petri calculus and then assembled by ordinary Petri calculus composition.

Structure of the paper. In § 2 we recall the main background on BIP and on the Petri calculus to keep the paper self-contained. In § 3 we define Hierarchical BIP and in § 4 we present the main result of the paper, namely the compositional encoding from HBI(P) to the Petri calculus. In § 5 we give some concluding remarks and discuss alternative approaches to HBIP. A toy running example of a client-server system is used to illustrate the main notions and constructions.

2 Background

2.1 The BIP component framework

This section reports on the formal definition of BIP as presented in [4]. Since we disregard priorities, we call BI(P) the framework presented here.

Definition 1 (Interaction). *Given a set of ports P, an* interaction *over P is a non-empty subset $a \subseteq P$.*

We write an interaction $\{p_1, p_2, \ldots, p_n\}$ as $p_1 p_2 \ldots p_n$ and $a \downarrow_{P_i}$ for the projection of an interaction $a \subseteq P$ over the set of ports $P_i \subseteq P$, i.e., $a \downarrow_{P_i} = a \cap P_i$. Projection extends to sets of interactions in the following way $\gamma \downarrow_P = \{a \downarrow_P \mid a \in \gamma \ \wedge \ a \downarrow_P \neq \varnothing\}$.

Definition 2 (Component). *A* component $B = (Q, P, \to)$ *is a transition system where Q is a set of states, P is a set of ports, and $\to \subseteq Q \times 2^P \times Q$ is the set of labelled transitions.*

As usual, we write $q \xrightarrow{a} q'$ to denote the transition $(q, a, q') \in \to$. An interaction a is enabled in q, denoted $q \xrightarrow{a}$, iff there exists q' s.t. $q \xrightarrow{a} q'$. By abusing notation, we will also write $q \xrightarrow{\varnothing} q$ for any q.

Definition 3 (BI(P) system). *A* BI(P) system $B = \gamma(B_1, \ldots, B_n)$ *is the composition of a finite set $\{B_i\}_{i=1}^n$ of components $B_i = (Q_i, P_i, \to_i)$ such that their sets of ports are pairwise disjoint, i.e., $P_i \cap P_j = \varnothing$ for $i \neq j$ parameterized by a set $\gamma \subset 2^P$ of interactions over the set of ports $P = \biguplus_{i=1}^n P_i$.*

The semantics of a BIP system $\gamma(B_1, \ldots, B_n)$ is given by the transition system $(Q, P, \to_\gamma)$, with $Q = \Pi_i Q_i$, $P = \biguplus_{i=1}^n P_i$ and $\to_\gamma \subseteq Q \times 2^P \times Q$ is the least set of transitions satisfying the following inference rule

$$\frac{a \in \gamma \qquad \forall i \in 1..n : q_i \xrightarrow{a \downarrow_{P_i}} q_i'}{(q_1, \ldots, q_n) \xrightarrow{a}_\gamma (q_1', \ldots, q_n')}$$

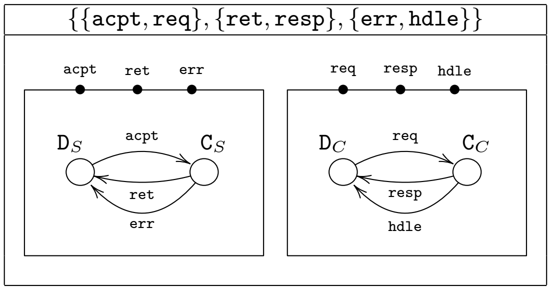


Fig. 1. A simple client/server BI(P) system

$$R ::= \ \bigcirc \ \mid \ \odot \ \mid \ \mathsf{I} \ \mid \ \mathsf{X} \ \mid \ \Delta \ \mid \ \nabla \ \mid \ \perp \ \mid \ \top \ \mid \ \wedge \ \mid \ \vee \ \mid \ \downarrow \ \mid \ \uparrow \ \mid \ R \otimes R \ \mid \ R ; R$$

Fig. 2. Petri calculus grammar

Example 1. Consider the BI(P) system shown in Fig. 1, which contains a component `Server` that sequentially interacts with a component `Client`. The `Server` accepts a request from the `Client` thanks to the interaction $\{\texttt{acpt}, \texttt{req}\}$. Then, the server can successfully answer the request by returning a value to the client (interaction $\{\texttt{ret}, \texttt{resp}\}$) or can fail the request by rising an error signal, which is handled by the client (interaction $\{\texttt{err}, \texttt{hdle}\}$). ◆

2.2 Petri Calculus

The *Petri calculus* [11] enriches the algebra of stateless connectors from [5] with one-place buffers along [1,11,6].

Terms of the Petri Calculus are defined by the grammar in Fig. 2. It consists of the following constants plus parallel and sequential composition: the empty place $\bigcirc$, the full place $\odot$, the identity wire I, the twist (also swap, or symmetry) X, the duplicator (also sync) Δ and its dual ∇, the mutex (also choice) $\wedge$ and its dual $\vee$, the hiding (also bang) $\perp$ and its dual $\top$, the inaction $\downarrow$ and its dual $\uparrow$. The diagrammatical representation of terms is shown in Fig. 3. For $n \in \mathbb{N}$, we write $\underline{n}$ to denote the finite ordinal $\underline{n} \stackrel{\mathsf{def}}{=} \{0, 1, \ldots, n-1\}$.

Any term has a unique associated *sort* (also called *type*) (k, l) with $k, l \in \mathbb{N}$, that fixes the size k of the left (input) interface and the size l of the right (output) interface of P. The type of constants are as follows: $\bigcirc$, $\odot$, and I have type $(1,1)$, $\mathsf{X} : (2,2)$, Δ and $\wedge$ have type $(1,2)$ and their duals ∇ and $\vee$ have type $(2,1)$, $\perp$ and $\downarrow$ have type $(1,0)$ and their duals $\top$ and $\uparrow$ have type $(0,1)$. The sort inference rules for composed processes are in Fig. 4.

The operational semantics is defined by the rules in Fig. 5, where $x, y \in \{0, 1\}$. The labels $\alpha, \beta, \rho, \sigma$ of transitions are binary strings, all transitions are sort-

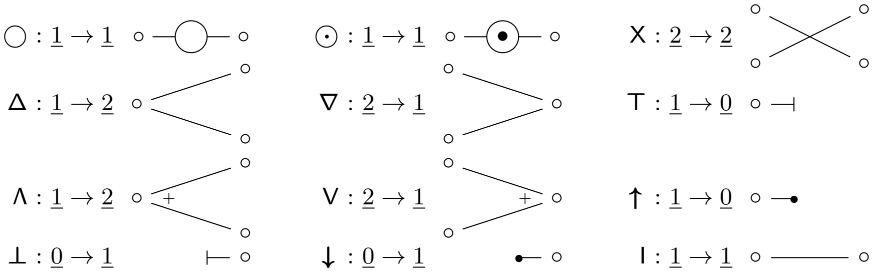


Fig. 3. Graphical representation of terms

$$\frac{R : (k, l) \quad R' : (m, n)}{R \otimes R' : (k + m, l + n)} \qquad \frac{R : (k, n) \quad R' : (n, l)}{R; R' : (k, l)}$$

Fig. 4. Sort inference rules

preserving, and if $R \xrightarrow[\beta]{\alpha} R'$ with $R, R' : (n, m)$, then $|\alpha| = n$ and $|\beta| = m$. Notably, bisimilarity induced by such a transition system is a congruence.

Due to space limitation we omit details here and refer the interested reader to [9].

Compound terms For the translation presented in § 4, we shall need additional families of compound terms, indexed by $n \in \mathbb{N}_+$ and $k \in \mathbb{N}$ (duals are omitted):

$$\mathsf{I}_n : (n,\, n) \qquad \mathsf{T}_n : (0,\, n) \qquad \uparrow_n : (0,\, n) \qquad \mathsf{X}_n : (n+1,\, n+1)$$
$$\Delta_n : (n,\, 2n) \quad \Lambda_n : (n,\, 2n) \quad \Delta_n^k : (n,\, k * n) \quad \Lambda_n^k : (n,\, k * n) \qquad d_n : (0,\, 2n)$$

Intuitively, I_n, T_n and $\uparrow_n$ correspond to n parallel copies of I, T and $\uparrow$, respectively. Connector Δ_n (and its dual ∇_n) is similar to Δ but duplicates n wires in the other interface, while Δ_n^k (and its dual ∇_n) replicates k-times the n wires of the other interface. Connector d_n (and its dual e_n) stands for the synchronisation of n pairs of wires. We now give the definitions:

$$\begin{array}{rlrlrl}
\mathsf{I}_n &= \bigotimes_n \mathsf{I} & \mathsf{X}_1 &= \mathsf{X} & \mathsf{X}_{n+1} &= (\mathsf{X}_n \otimes \mathsf{I}); (\mathsf{I}_n \otimes \mathsf{X}) \\
\uparrow_n &= \bigotimes_n \uparrow & \Delta_1 &= \Delta & \Delta_{n+1} &= (\Delta \otimes \Delta_n); (\mathsf{I} \otimes \mathsf{X}_n \otimes \mathsf{I}_n) \\
\mathsf{T}_n &= \bigotimes_n \mathsf{T} & \Lambda_1 &= \Lambda & \Lambda_{n+1} &= (\Lambda \otimes \Lambda_n); (\mathsf{I} \otimes \mathsf{X}_n \otimes \mathsf{I}_n) \\
d_n &= \mathsf{T}_n; \Delta_n & \Delta_n^0 &= \mathsf{T}_n & \Delta_n^{k+1} &= \Delta_n; (\Delta_n^k \otimes \mathsf{I}_n) \\
 & & \Lambda_n^0 &= \uparrow_n & \Lambda_n^{k+1} &= \Lambda_n; (\Lambda_n^k \otimes \mathsf{I}_n)
\end{array}$$

The behaviour of compound terms is characterised by the next proposition.

Proposition 1 (from [6]). *For $n > 0$,*

$$\frac{}{\bigcirc \xrightarrow[0]{1} \odot}\ (\textsc{TkI}) \qquad \frac{}{\odot \xrightarrow[1]{0} \bigcirc}\ (\textsc{TkO1}) \qquad \frac{}{\mathsf{I} \xrightarrow[1]{1} \mathsf{I}}\ (\textsc{Id}) \qquad \frac{}{\mathsf{X} \xrightarrow[ba]{ab} \mathsf{X}}\ (\textsc{Tw})$$

$$\frac{}{\bot \xrightarrow{1} \bot}\ (\bot) \qquad \frac{}{\top \xrightarrow[1]{} \top}\ (\top) \qquad \frac{}{\Delta \xrightarrow[11]{1} \Delta}\ (\Delta) \qquad \frac{}{\nabla \xrightarrow[1]{11} \nabla}\ (\nabla)$$

$$\frac{}{\wedge \xrightarrow[(1-a)a]{1} \wedge}\ (\wedge a) \qquad \frac{}{\vee \xrightarrow[1]{(1-a)a} \vee}\ (\vee a) \qquad \frac{\mathsf{C} : (k,\, l) \text{ a basic connector}}{\mathsf{C} \xrightarrow[0^l]{0^k} \mathsf{C}}\ (\textsc{Refl})$$

$$\frac{P \xrightarrow[\gamma]{\alpha} Q \quad R \xrightarrow[\beta]{\gamma} S}{P \,;\, R \xrightarrow[\beta]{\alpha} Q \,;\, S}\ (\textsc{Cut}) \qquad \frac{P \xrightarrow[\beta]{\alpha} Q \quad R \xrightarrow[\Delta]{\gamma} S}{P \otimes R \xrightarrow[\beta\Delta]{\alpha\gamma} Q \otimes S}\ (\textsc{Ten})$$

Fig. 5. Operational semantics for the Petri Calculus

1. $\mathsf{X}_n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \mathsf{X}_n$, $\alpha = h_0 \ldots h_n$ *and* $\beta = h_1 \ldots h_n h_0$.
2. $\Delta_n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \Delta_n$, $\#\alpha = n$, $\#\beta = 2n$ *and* $\alpha_i = \beta_i = \beta_{n+i}$ *for all* $i < n$.
3. $\nabla_n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \nabla_n$, $\#\alpha = 2n$, $\#\beta = n$ *and* $\alpha_i = \alpha_{n+i} = \beta_i$ *for all* $i < n$.
4. $\wedge_n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \wedge_n$, $\#\alpha = n$, $\#\beta = 2n$, $\alpha_i = \beta_i + \beta_{n+i}$ *for all* $i < n$.
5. $\vee_n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \vee_n$, $\#\alpha = 2n$, $\#\beta = n$, $\beta_i = \alpha_i + \alpha_{n+i}$ *for all* $i < n$.
6. $\wedge_l^n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \wedge_l^n$, $\#\alpha = l$, $\#\beta = nl$ *and* $\alpha_i = \Sigma_{j<n} \beta_{jl+i}$ *for all* $i < l$.
7. $\vee_l^n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \vee_l^n$, $\#\alpha = nl$, $\#\beta = l$ *and* $\beta_i = \Sigma_{j<n} \alpha_{jl+i}$ *for all* $i < l$.
8. $\Delta_l^n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \Delta_l^n$, $\#\alpha = l$, $\#\beta = nl$ *and* $\beta_{lj+i} = \alpha_i$ *for all* $i < l$ *,*$j < n$.
9. $\nabla_l^n \xrightarrow[\beta]{\alpha} t$ *iff* $t = \nabla_l^n$, $\#\alpha = nl$, $\beta = l$ *and* $\beta_i = \alpha_{lj+i}$ *for all* $i < l$ *and* $j < n$.
10. $d_n \xrightarrow[\beta]{\alpha} t$ *iff* $t = d_n$, $\#\alpha = 0$, $\#\beta = 2n$ *and* $\beta_i = \beta_{n+i}$ *for all* $i < n$ *and* $j < n$.

Relational forms The encoding proposed in § 4 uses two classes of terms of the Petri calculus, called the left and right *relational forms*, that represent functions as Petri calculus terms [11].

For any $h \in \mathbb{N}$, there is a bijection $\ulcorner\_\urcorner : 2^{\underline{h}} \to \{0,1\}^h$ with

$$\ulcorner U \urcorner_i \overset{\mathsf{def}}{=} \begin{cases} 1 \text{ if } i \in U \\ 0 \text{ otherwise} \end{cases}$$

For Θ a set of Petri calculus terms, let T_Θ denote the set of terms generated by the following grammar:

$$T_\Theta \ ::= \ \theta \in \Theta \ \mid \ \mathsf{I} \ \mid \ T_\Theta \otimes T_\Theta \ \mid \ T_\Theta \,;\, T_\Theta.$$

We shall use t_Θ to range over terms of T_Θ.

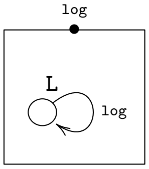


Fig. 6. Component `Logging`

Definition 4. *A term* $t : (k,\, l)$ *is in* right relational form *when it is in*

$$T_{\{\perp\}}\; ;\, T_{\{\Delta\}}\; ;\, T_{\{\mathsf{X}\}}\; ;\, T_{\{\vee\}}\; ;\, T_{\{\uparrow\}}.$$

Dually, t is said to be in left relational form *when it is in*

$$T_{\{\downarrow\}}\; ;\, T_{\{\wedge\}}\; ;\, T_{\{\mathsf{X}\}}\; ;\, T_{\{\nabla\}}\; ;\, T_{\{\top\}}.$$

The following result spells out the significance of the relational forms.

Lemma 1 (From [11]). *For each function* $f\colon \underline{k} \to 2^{\underline{l}}$ *there exists a term* $\rho_f : (k,\, l)$ *in right relational form, the dynamics of which are characterised by the following:*

$$\rho_f \xrightarrow[\beta]{\alpha} \rho_f \;\Leftrightarrow\; \exists U \subseteq \underline{k} \textit{ s.t. } \forall u, v \in U.\ u \neq v \Rightarrow f(u) \cap f(v) = \varnothing,\ \alpha = \ulcorner U \urcorner \\ \textit{and } \beta = \ulcorner f(U) \urcorner$$

The symmetric result holds for functions $f\colon \underline{k} \to 2^{\underline{l}}$ *and terms* $t : (l,\, k)$ *in left relational form. That is, there exists* $\lambda_f : (l,\, k)$ *in left relational form with semantics*

$$\lambda_f \xrightarrow[\beta]{\alpha} \lambda_f \;\Leftrightarrow\; \exists U \subseteq \underline{k} \textit{ s.t. } \forall u, v \in U.\ u \neq v \Rightarrow f(u) \cap f(v) = \varnothing,\ \beta = \ulcorner U \urcorner \\ \textit{and } \alpha = \ulcorner f(U) \urcorner$$

3 Hierarchical BIP systems

In this section we address the hierarchical composition of BI(P) systems, i.e., BI(P) systems can be taken as components of larger systems.

Example 2. Consider the scenario introduced in Example 1, which should be extended with a logging functionality in order to record all error responses sent by the component `Server`. Assume we already have the simple component for logging depicted in Fig. 6. In this case, we would like to consider the system in Fig. 1 as a single component to define a new BI(P) system, as the one shown in Fig. 7. We remark that the interface (i.e., the set of ports) exposed by the client/server subsystem is just its set of interactions. This ensures that the composed system does not change the behaviour of the underlying subsystems. ⧫

Next definition formally introduces the notion of hierarchical composition of systems

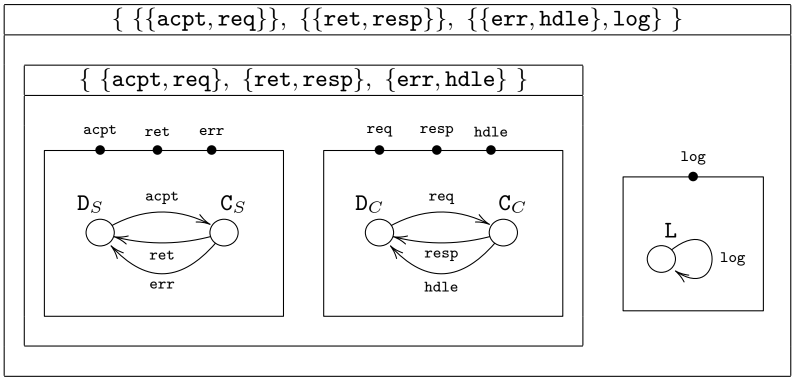

{ {{acpt, req}}, {{ret, resp}}, {{err, hdle}, log} }
{ {acpt, req}, {ret, resp}, {err, hdle} }
acpt
ret
err
D_S
acpt
C_S
ret
err
req
resp
hdle
D_C
req
C_C
resp
hdle
log
L
log

Fig. 7. A simple BI(P) system

Definition 5 (HBI(P) system). *A Hierarchical BI(P) system (HBI(P)) is either*

- *a BI(P) component $B = (Q, P, \to)$ with interface $\iota(B) = P$; or*
- *a composite system $B = \gamma(B_1, \dots, B_n)$ with interface $\iota(B) = \gamma$ where $\{B_1, \dots, B_n\}$ is a set of hierarchical BI(P) systems with pairwise disjoint interfaces, i.e., $\iota(B_i) \cap \iota(B_j) = \varnothing$ for $i \neq j$, and γ is a set of interactions over $\biguplus_{i=1}^{n} \iota(B_i)$.*

The semantics of HBI(P) systems is defined analogously to that of BI(P) systems as the synchronous execution of the transitions of its constituent components matching one defined interaction. We start by defining the state space $\mathbb{Q}_B$ of a HBI(P) system B as follows:

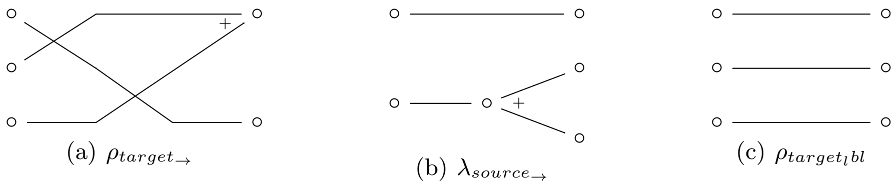
- $\mathbb{Q}_B = Q$ if $B = (Q, P, \to)$
- $\mathbb{Q}_B = \mathbb{Q}_{B_1} \times \dots \times \mathbb{Q}_{B_n}$ if $B = \gamma(B_1, \dots, B_n)$.

Then, the semantics of a composite HBI(P) system B is given by the transition system $(\mathbb{Q}_B, \iota(B), \to)$ where $\to \subseteq \mathbb{Q}_B \times \iota(B) \times \mathbb{Q}_B$ is the least set of transitions satisfying the following inference rules

$$\frac{a \in \iota(B) \qquad \forall i \in 1..n : q_i \xrightarrow{a\downarrow_{\iota(B_i)}} q'_i}{(q_1, \dots, q_n) \xrightarrow{a} (q'_1, \dots, q'_n)}$$

4 HBI(P) systems as Petri calculus terms

This section gives an encoding of HBI(P) systems into the Petri calculus. Note that, differently from BI(P), the Petri calculus uses consecutive natural numbers to designate ports over interfaces. In order to establish a correspondence between HBI(P) systems and Petri calculus terms, we will map names into natural numbers, i.e., given a finite set S with $k = \#S$, we use w_S to denote an injective



(a) $\rho_{target_{\rightarrow}}$

(b) $\lambda_{source_{\rightarrow}}$

(c) ρ_{target_lbl}

Fig. 8. Petri calculus terms for relational forms

function $w_S : S \rightarrow \underline{k}$ that orders the elements of S. By abusing notation, we write w_S to also denote its expected extension $w_S : 2^S \rightarrow 2^{\underline{k}}$.

4.1 Encoding of basic components

We first address the encoding of basic components (Definition 2) as Petri calculus terms. The encoding of components follows along the lines of the encoding of Petri nets proposed in [9], although it is simpler due to the fact that components are just sequential systems.

Given a component $B = (Q, P, \rightarrow)$ with s transitions (i.e., $\# \rightarrow = s$), we rely on the functions w_Q, w_P and $w_{\rightarrow}$ that respectively order the elements in Q, P and $\rightarrow$. In addition, we will use the following three functions $source_{\rightarrow}$, $target_{\rightarrow}$, $lbl_{\rightarrow}$, which map a labelled transition belonging to $\rightarrow$ into its source, target and labels, when considering the sets of names just as ordinals.

$$\begin{array}{llll} source_{\rightarrow} & : \underline{s} \rightarrow \underline{\#Q} & s.t. & source_{\rightarrow}(w_{\rightarrow}(q \xrightarrow{a} q')) = w_Q(q) \\ target_{\rightarrow} & : \underline{s} \rightarrow \underline{\#Q} & s.t. & target_{\rightarrow}(w_{\rightarrow}(q \xrightarrow{a} q')) = w_Q(q') \\ lbl_{\rightarrow} & : \underline{s} \rightarrow \underline{2^{\#P}} & s.t. & lbl_{\rightarrow}(w_{\rightarrow}(q \xrightarrow{a} q')) = w_P(a) \end{array}$$

Thanks to Lemma 1, we know that the relational forms $\lambda_{source_{\rightarrow}}$, $\rho_{target_{\rightarrow}}$ and $\rho_{lbl_{\rightarrow}}$ exist and their behaviours are in tight correspondence with the associated functions.

Example 3. Consider the component `Server` in Fig. 1 and assume the following ordering functions

$$\begin{array}{lll} w_Q(\mathtt{D}_S) = 0 & w_P(\mathtt{acpt}) = 0 & w_{\rightarrow}(\mathtt{D}_S \xrightarrow{\mathtt{acpt}} \mathtt{C}_S) = 0 \\ w_Q(\mathtt{C}_S) = 1 & w_P(\mathtt{ret}) \ = 1 & w_{\rightarrow}(\mathtt{C}_S \xrightarrow{\mathtt{ret}} \mathtt{D}_S) \ = 1 \\ & w_P(\mathtt{err}) \ = 2 & w_{\rightarrow}(\mathtt{C}_S \xrightarrow{\mathtt{err}} \mathtt{D}_S) \ = 2 \end{array}$$

Then, the relational forms associated to the above functions are defined as follows and depicted in Fig. 8.

$$\rho_{target_{\rightarrow}} = (\mathsf{X} \otimes \mathsf{I}) \ ; (\mathsf{X} \otimes \mathsf{I}) \ ; (\mathsf{V} \otimes \mathsf{I}) \qquad \lambda_{source_{\rightarrow}} = (\mathsf{I} \otimes \mathsf{\Lambda}) \qquad \rho_{target_{\rightarrow}} = \mathsf{I}_3$$

♦

Definition 6. *Let $B = (Q, P, \rightarrow)$ be a component. The Petri Calculus term corresponding to the behaviour of B in state $q \in Q$ is $[\![B]\!] : \underline{0} \rightarrow \underline{\#P}$, which is defined as follows*

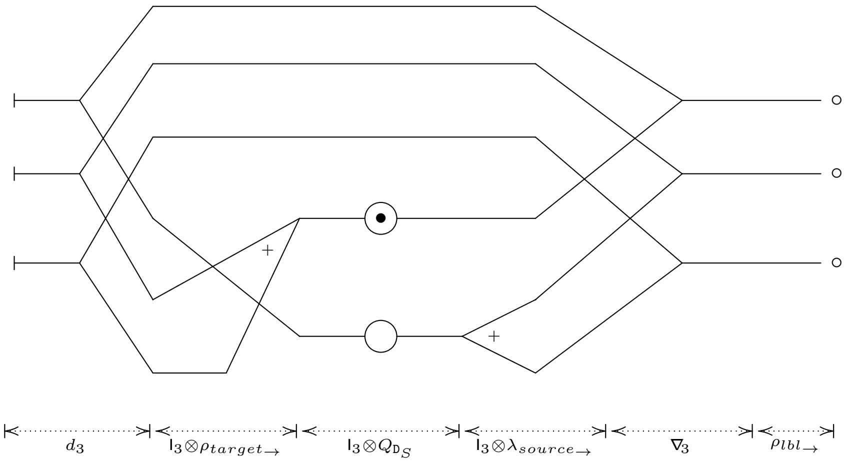

+
+
d_3
$\mathsf{I}_3\otimes\rho_{target\rightarrow}$
$\mathsf{I}_3\otimes Q_{\mathtt{D}_S}$
$\mathsf{I}_3\otimes\lambda_{source\rightarrow}$
∇_3
$\rho_{lbl\rightarrow}$

Fig. 9. Encoding of the BI(P) component `Server`

$$T_{B_q} \overset{\text{def}}{=} d_s \ ; (\mathsf{I}_s \otimes (\rho_{target\rightarrow} \ ; Q_{\{q\}} \ ; \lambda_{source\rightarrow})) \ ; \nabla_s ; \rho_{lbl\rightarrow}$$

where

$$Q_{Q'} \overset{\text{def}}{=} \bigotimes_{i<\#Q} q_i \quad \textit{where} \quad q_i \overset{\text{def}}{=} \begin{cases} \odot & \textit{if } i \in w_Q(Q') \\ \bigcirc & \textit{otherwise} \end{cases}$$

Example 4. Consider the component `Server` introduced in Example 1. Figure 9 shows the term $T_{B_{\mathtt{D}_S}}$ corresponding to the encoding of the component `Server` for the initial state $\mathtt{D}_S$ and the ordering functions given in Example 3. ♦

The following results formalise the relation between the behaviour of components and their encodings. The first lemma is auxiliary and characterises the behavior of terms of the form $Q_{Q'}$

Lemma 2. $Q_{Q'} \xrightarrow[\ulcorner W \urcorner]{\ulcorner Z \urcorner} R$ *iff* $R = Q_{Q''}$, $W \subseteq Q'$, $Z \cap Q' = \varnothing$ *and* $Q'' = (Q' \setminus W) \cup Z$.

Proof. Examination of either rules $(\bot_1)$ and $(\top_1)$, together with the rule (Cut) (when $p = 0$) or rules (TkI) and (TkO), together with the rule (Ten) (when $p > 0$).□

Next result ensures that the transitions of a component are in one-to-one correspondence with the moves of the corresponding Petri calculus term.

Theorem 1. *Let* $B = (Q, P, \rightarrow)$ *a basic component. Then,*

(i) if $q \xrightarrow{a} q'$ *then* $T_{B_q} \xrightarrow[\ulcorner w_P(a) \urcorner]{} T_{B_{q'}}$.

(ii) if $T_{B_q} \xrightarrow[\beta]{} R$ *then there exists* q' *s.t.* $q \xrightarrow{a} q'$, $R = T_{B_{q'}}$ *and* $\beta = \ulcorner w_P(a) \urcorner$.

Proof. 1) By Lemma 2, $Q_q \xrightarrow[\ulcorner w_Q(q) \urcorner]{\ulcorner w_Q(q') \urcorner} Q_{q'}$. Then, by Lemma 1 and rule (Cut) after noting that $\mathit{source}_{\rightarrow}(w_{\rightarrow}(q \xrightarrow{a} q')) = w_Q(q)$ and $\mathit{target}_{\rightarrow}(w_{\rightarrow}(q \xrightarrow{a} q')) = w_Q(q')$, we have

$$(\rho_{\mathit{target}_{\rightarrow}} \ ; Q_{\{q\}} \ ; \lambda_{\mathit{source}_{\rightarrow}})) \xrightarrow[\ulcorner w_{\rightarrow}(q \xrightarrow{a} q') \urcorner]{\ulcorner w_{\rightarrow}(q \xrightarrow{a} q') \urcorner} (\rho_{\mathit{target}_{\rightarrow}} \ ; Q_{\{q\}} \ ; \lambda_{\mathit{source}_{\rightarrow}}))$$

and subsequently by Proposition 1 and rules (Cut) and (Ten)

$$d_s \ ; (\mathsf{I}_s \otimes (\rho_{\mathit{target}_{\rightarrow}} \ ; Q_{\{q\}} \ ; \lambda_{\mathit{source}_{\rightarrow}})) \ ; \nabla_s \xrightarrow{\ulcorner w_{\rightarrow}(q \xrightarrow{a} q') \urcorner}$$
$$d_s \ ; (\mathsf{I}_s \otimes (\rho_{\mathit{target}_{\rightarrow}} \ ; Q_{\{q\}} \ ; \lambda_{\mathit{source}_{\rightarrow}})) \ ; \nabla_s$$

The proof is completed by noting that $\mathit{lbl}_{\rightarrow}(w_{\rightarrow}(q \xrightarrow{a} q')) = w_P(a)$ and using Lemma 1 and rule (Cut) to conclude

$$T_{B_q} \xrightarrow[\ulcorner w_P(a) \urcorner]{} T_{B_{q'}}$$

2) If $T_{B_q} \xrightarrow[\beta]{} R$, then by rule (Cut) $\rho_{\mathit{lbl}_{\rightarrow}} \xrightarrow[\beta]{\alpha} \rho_{\mathit{lbl}_{\rightarrow}}$, $\nabla_s ; \rho_{\mathit{lbl}_{\rightarrow}} \xrightarrow[\beta]{\alpha\alpha} \nabla_s ; \rho_{\mathit{lbl}_{\rightarrow}}$ and

$$(\mathsf{I}_s \otimes (\rho_{\mathit{target}_{\rightarrow}} \ ; Q_{\{q\}} \ ; \lambda_{\mathit{source}_{\rightarrow}})) \xrightarrow[\alpha\alpha]{\alpha\alpha} (\mathsf{I}_s \otimes R)$$

Hence,

$$(\rho_{\mathit{target}_{\rightarrow}} \ ; Q_{\{q\}} \ ; \lambda_{\mathit{source}_{\rightarrow}}) \xrightarrow[\alpha]{\alpha} R$$

The only non trivial transition implies $Q_{\{q\}} \xrightarrow[\ulcorner q \urcorner]{\ulcorner Q' \urcorner} Q'_Q$. By Lemma 1 and $w_Q(q) = \mathit{source}_{\rightarrow}(w_{\rightarrow}(q \xrightarrow{a} q'))$, we conclude

$$\lambda_{\mathit{source}_{\rightarrow}} \xrightarrow[\alpha]{\ulcorner w_Q(q) \urcorner} \lambda_{\mathit{source}_{\rightarrow}}$$

with $\alpha = \ulcorner w_{\rightarrow}(q \xrightarrow{a} q') \urcorner$. By reasoning analogously, on $\rho_{\mathit{target}_{\rightarrow}}$, we have that

$$\rho_{\mathit{target}_{\rightarrow}} \xrightarrow[\ulcorner w_Q(q') \urcorner]{\alpha} \rho_{\mathit{target}_{\rightarrow}}$$

Hence, $R = (\rho_{\mathit{target}_{\rightarrow}} \ ; Q_{\{q'\}} \ ; \lambda_{\mathit{source}_{\rightarrow}})$. The proof is completed by using Lemma 1 to conclude that $\rho_{\mathit{lbl}_{\rightarrow}} \xrightarrow[\ulcorner w_P(a) \urcorner]{\alpha} \rho_{\mathit{lbl}_{\rightarrow}}$. □

Example 5. It can be easily checked that the term $T_{B_{\mathtt{D}_S}}$ introduced in Example 4 has the following transitions:

$T_{B_{\mathtt{D}_S}} \xrightarrow[000]{} T_{B_{\mathtt{D}_S}}$ $\quad T_{B_{\mathtt{D}_S}} \xrightarrow[100]{} T_{B_{\mathtt{C}_S}}$ $\quad T_{B_{\mathtt{C}_S}} \xrightarrow[010]{} T_{B_{\mathtt{D}_S}}$ $\quad T_{B_{\mathtt{C}_S}} \xrightarrow[001]{} T_{B_{\mathtt{C}_S}}$

that correspond to the transitions

$\mathtt{D}_S \xrightarrow{\varnothing} \mathtt{D}_S$ $\quad \mathtt{D}_S \xrightarrow{\mathtt{acpt}} \mathtt{C}_S$ $\quad \mathtt{C}_S \xrightarrow{\mathtt{ret}} \mathtt{D}_S$ $\quad \mathtt{C}_S \xrightarrow{\mathtt{err}} \mathtt{D}_S$ ♦

4.2 Encoding of interactions

We now focus on the encoding of a interaction as a stateless connector. For $\alpha \in \{0,1\}^h$ a binary string of length $h > 0$, we let $R^\alpha : (h, 1)$ denote the process inductively defined by:

$$R^0 = \downarrow; \top \qquad R^1 = \mathsf{I} \qquad R^{x\alpha} = (R^x \otimes R^\alpha); \nabla$$

Intuitively, the term R^α synchronizes the ports associated to the positions of α that are set to 1.

Lemma 3. *The process R^α is stateless for any α, i.e., whenever $R^\alpha \xrightarrow[\beta']{\beta} R'$ then $R' = R^\alpha$.*

Proof. The thesis follows simply by noting that $(\!|\gamma|\!)_I$ is composed out of stateless connectors, i.e., the encoding does not exploit the constants $\bigcirc, \odot$. □

Lemma 4. *$R^\alpha \xrightarrow[\beta']{\beta} R'$ iff*

1. *$\beta = 0^{\#\alpha}$ and $\beta' = 0$; or*
2. *$\beta = \alpha$ and $\beta' = 1$.*

Proof. The thesis follows by induction on the length of α. □

Definition 7. *Let γ be a finite set of interactions over a finite set of ports I, $j = \#I$ and $k = \#\gamma$. After fixing w_I and w_γ, the encoding for the set γ is*

$$[\![\gamma]\!]_I = \wedge_j^k; (\!|\gamma|\!)_I; \Delta_k; (\mathsf{I}_k \otimes (\vee_1^k; \bot))$$

where

$$(\!|\{a\}|\!)_I = R^{\ulcorner w_I(a) \urcorner}$$
$$(\!|\{a\} \cup \gamma'|\!)_I = R^{\ulcorner w_I(a) \urcorner} \otimes (\!|\gamma'|\!)_I \quad \textit{when } a \textit{ is the minimum in } \{a\} \cup \gamma' \textit{ w.r.t. } w_\gamma$$

Example 6. Consider the set $\gamma = \{\{\mathtt{acpt}, \mathtt{req}\}, \{\mathtt{ret}, \mathtt{resp}\}, \{\mathtt{err}, \mathtt{hdle}\}\}$ defined over $I = \{\mathtt{acpt}, \mathtt{ret}, \mathtt{err}, \mathtt{req}, \mathtt{resp}, \mathtt{hdle}\}$ with the ordering functions defined as follows:

$$w_\gamma(\{\mathtt{acpt}, \mathtt{req}\}) = 0 \quad w_\gamma(\{\mathtt{ret}, \mathtt{resp}\}) = 1 \quad w_\gamma(\{\mathtt{err}, \mathtt{hdle}\}) = 2$$

$$w_I(\mathtt{acpt}) = 0 \quad w_I(\mathtt{ret}) = 1 \quad w_I(\mathtt{err}) = 2$$

$$w_I(\mathtt{req}) = 3 \quad w_I(\mathtt{resp}) = 4 \quad w_I(\mathtt{hdle}) = 5$$

Figure 6 shows the subterm $\wedge_6^3; (\!|\gamma|\!)_I$ (we use a compact representation in which a chain of several identical connectors like $\wedge$ or ∇ are collapsed in a unique node). Note that $(\!|\gamma|\!)_I = R^{\ulcorner w_I(\{\mathtt{acpt},\mathtt{req}\}) \urcorner} \otimes R^{\ulcorner w_I(\{\mathtt{ret},\mathtt{resp}\}) \urcorner} \otimes R^{\ulcorner w_I(\{\mathtt{err},\mathtt{hdle}\}) \urcorner}$ stands for the parallel evaluation of the three interactions in γ. The term $\wedge_6^3$ ensures that conflicting interactions (i.e., the ones sharing a common action) are performed in mutual exclusion. We remark that the term $\wedge_6^3; (\!|\gamma|\!)_I$ still would allow for the concurrent execution of non-conflicting interactions (e.g., involving different sets of components). Since the semantics of BI(P) is purely sequential, we need to forbid the concurrent execution of disjoint interactions. This is ensured in the complete encoding by the subterm $\Delta_3; (\mathsf{I}_3 \otimes (\vee_1^3; \bot))$. ♦

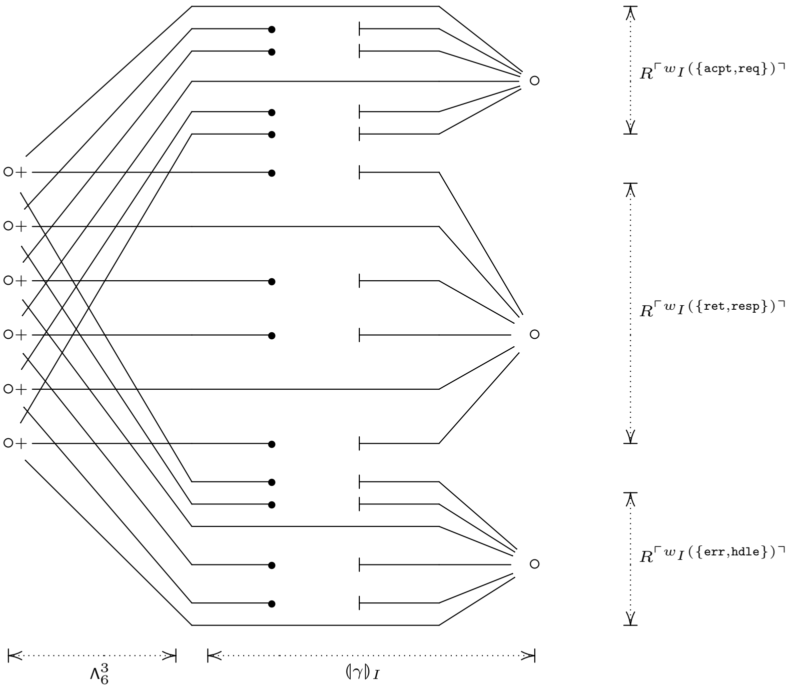

$R^{\ulcorner w_I(\{\mathtt{acpt},\mathtt{req}\})\urcorner}$
$R^{\ulcorner w_I(\{\mathtt{ret},\mathtt{resp}\})\urcorner}$
$R^{\ulcorner w_I(\{\mathtt{err},\mathtt{hdle}\})\urcorner}$
$\wedge^3_6$
$(\!|\gamma|\!)_I$

Fig. 10. Graphical representation of $\wedge^3_6; (\!|\gamma|\!)_I$

The following results characterise the behaviour of $[\![\gamma]\!]_I$ and are instrumental to the proof of our main result (Theorem 2).

Lemma 5. *Let γ be a synchronization set over I with $\#I = j$ and $\#\gamma = k$, and w_I and w_γ the functions sorting the elements of I and γ.Then*

1. $(\!|\gamma|\!)_I : (j * k, k)$.
2. $(\!|\gamma|\!)_I$ *is stateless for any γ, i.e., whenever* $(\!|\gamma|\!)_I \xrightarrow[\beta]{\alpha} R$ *then* $R = (\!|\gamma|\!)_I$.
3. $(\!|\gamma|\!)_I \xrightarrow[\beta]{\alpha} R$ *iff* $\forall 0 \leq h \leq k-1$ *either*
   - $\beta_h = 0$ *and* $\forall j * h \leq i \leq j * (h+1) - 1 : \alpha_i = 0$, *or*
   - $\beta_h = 1$, *and* $\exists a \in \gamma$ *s.t.* $w_\gamma(a) = h$ *and* $\alpha_{j*h..j*(h+1)-1} = \ulcorner w_I(a) \urcorner$.

Proof. (1) Follows by induction on $\#\gamma$. (2) Follows by noting that $(\!|\gamma|\!)_I$ is composed out of stateless connectors, i.e., the encoding does not exploit the constants $\bigcirc, \bigodot$. (3) By induction on h and using Lemma 4. □

Lemma 6. *Let γ be a synchronization set over I with $\#I = j$ and $\#\gamma = k$, and w_I and w_γ the functions sorting the elements of I and γ.Then*

1. $[\![\gamma]\!]_I : (j, k)$.

2. $[\![\gamma]\!]_I$ *is stateless for any* γ*, i.e., whenever* $[\![\gamma]\!]_I \xrightarrow[\beta]{\alpha} R$ *then* $R = [\![\gamma]\!]_I$*.*

3. $[\![\gamma]\!]_I \xrightarrow[\beta]{\alpha} R$ *iff* $\exists a \subseteq \gamma$ *s.t.* $\#a \leq 1$*,* $\alpha = \ulcorner w_I(a) \urcorner$ *and* $\beta = \ulcorner w_\gamma(a) \urcorner$*.*

Proof. (1) Follows from the fact that: $\mathsf{\Lambda}^k_{\#I} : (j, j*k)$; $(\!|\gamma|\!)_I : (j*k, k)$ by Lemma 5(1); $\mathsf{\Delta}_k : (k, 2k)$; and $(\mathsf{I}_k \otimes (\mathsf{V}^k_1; \bot)) : (2k, k)$. (2) Follows by noting that $(\!|\gamma|\!)_I$ is composed out of stateless connectors, i.e., the encoding does not exploit the constants $\bigcirc, \odot$. (3) By Proposition 1(5),

$$\mathsf{V}^k_1 \xrightarrow[\beta]{\alpha'} \mathsf{V}^k_1$$

with $\#\alpha' = k$, $\#\beta = 1$ and $\beta = \sum_{j<k} \alpha'_j$. Therefore, $\sum_{j<k} \alpha'_j \leq 1$. By the semantics of $\bot$ and ;, we have

$$\mathsf{V}^k_1; \bot \xrightarrow{\alpha'} \mathsf{V}^k_1; \bot$$

By using Proposition 1(2) and the inference rules for ; and $\otimes$

$$\mathsf{\Delta}_k; (\mathsf{I}_k \otimes (\mathsf{V}^k_1; \bot)) \xrightarrow[\alpha']{\alpha'} \mathsf{\Delta}_k; (\mathsf{I}_k \otimes (\mathsf{V}^k_1; \bot))$$

The proof is completed by using Lemma 6(3), Proposition 1(6) and the semantics of ;. □

4.3 Encoding of HBI(P) systems

The encoding of HBI(P) systems is defined by a suitable combination of the encoding of basic components and interactions.

Definition 8. *Let* B *be a HBI(P) system with initial* $q \in \mathbb{Q}_B$*. The corresponding Petri Calculus term is inductively defined as follows.*

$$[\![B]\!]_q = \begin{cases} T_{B_q} & if B = (Q, P, \rightarrow) \\ ([\![B_1]\!]_{q_1} \otimes \ldots \otimes [\![B_n]\!]_{q_n}); [\![\gamma]\!]_I & if B = \gamma(B_1, \ldots, B_n), q = (q_1, \ldots, q_n), \\ \quad I = \cup_{i=1}^n \iota(B_i), \text{ and } w_I \text{ s.t. } \forall a : w_I(a) = w_{\iota(B_i)}(a) + \sum_{j=1}^{i-1} \#\iota(B_j) & \end{cases}$$

Theorem 2 (Correspondence). *Let* B *be a HBI(P) system with initial state* q*. Then* $q \xrightarrow{a} q'$ *if and only if* $[\![B]\!]_q \xrightarrow[\ulcorner w_{\iota(B)(a)} \urcorner]{} [\![B]\!]_{q'}$*.*

Proof. By induction on the structure of the system B. Base case ($B = (Q, P, \rightarrow)$) follows by Theorem 1. Inductive step follows by applying inductive hypothesis on $[\![B_1]\!]_{q_1}, \ldots, [\![B_n]\!]_{q_n}$. Then, the proof is completed by using Lemma 6. □

5 Conclusion

This paper studies the hierarchical composition of BI(P) systems and its relation with the Petri calculus. For convenience of presentation we have chosen the particular variant of BI(P) consisting of the basic interaction model and purely sequential execution. Nevertheless, the results presented in this paper can be extended or adapted to handle several variants proposed in the literature. The remaining of this section is devoted to the discussion of some alternative presentations for HBI(P) and their relation with the encodings proposed in this paper.

Concurrent executions The work in [10] proposes a notion of hierarchical composition of BI(P) systems that allows for the concurrent execution of interactions, i.e., a set of conflict-free interactions (i.e., interactions that are pairwise disjoint) can be fired concurrently if enabled. We could encode such behaviour simply by defining $[\![\gamma]\!]_I$ as follows.

$$[\![\gamma]\!]_I = \Lambda^k_{\#I} ; (\!|\gamma|\!)_I$$

This definition simplifies Definition 7 by removing the subterm $\Delta_k ; (\mathsf{I}_k \otimes (\mathsf{V}^k_1 ; \perp))$. As already mentioned in § 4.2, the subterm $(\mathsf{V}^k_1 ; \perp)$ ensures the execution of a unique interaction at a time. The results presented in the previous sections could also be formulated for this variant with minor adjustments.

For the sake of uniformity, the proposal in [10] also considers concurrent basic components instead of just sequential components as originally proposed in BIP. Concurrent basic components could be modelled as C/E or P/T nets with boundaries, which can be encoded as Petri calculus terms as shown in [9].

Triggers In order to represent different modes of synchronisation, the BIP model has been extended with a sorting discipline for ports in [3]. Typing associates synchronization types (`trigger` or `synchron`) to ports or connectors. The main difference is that an interaction $\{p_1, \ldots, p_n\}$ actually represents a set of interactions, i.e., all nonempty subset of $\{p_1, \ldots, p_n\}$ that contains some `trigger`; otherwise (if all of the ports are `synchrons`), the only possible interaction is the maximal one. Then, an interaction set γ contains either standard interactions (i.e., without triggers), denoted as before by a, and connectors (i.e., interactions containing at least a trigger), denoted by c. Let c be a connector, we write γ_{c} for the set of all standard denoted interactions (i.e., all subsets of c that contains a trigger). By assuming a set of interactions, we extend Definition 7 with the rules for encoding connectors

$$(\!|\{\mathsf{c}\}|\!)_I = \Lambda^k_j \ ; (\!|\gamma_{\mathsf{c}}|\!)_I \ ; \mathsf{V}^k_1$$

with $j = \#I$ and $k = \#\gamma_{\mathsf{C}}$ (rule for $(\!|\{\mathsf{c}\} \cup \gamma'|\!)_I$ is analogous).

By using Lemma 5 and the semantics of Λ^k_j and V^k_j, it is easy to conclude that the only non trivial transitions of $(\!|\{\mathsf{c}\}|\!)_I$ are $(\!|\{\mathsf{c}\}|\!)_I \xrightarrow[1]{\alpha} (\!|\{\mathsf{c}\}|\!)_I$ with $\alpha = \ulcorner w_I(a) \urcorner$ and $a \in \gamma_{\mathsf{c}}$. This characterization is analogous to the one for standard transitions in Lemma 4. This suffices to show that the correspondence results smoothly extend to the semantics of triggers.

Hiding Hiding is an usual operator when composing systems hierarchically, because it enables components to compute internally. We can incorporate hiding to the definition of a HBI(P) system by adding the following item to Def. 5:

– a composed system $B = \nu a B_1$ with interface $\iota(B) = \iota(B_1) \smallsetminus \{a\}$ where B_1 is a hierarchical BI(P) system and $a \in \iota(B_1)$.

The semantics of $B = \nu a B_1$ is given by extending the definition of the space state of a HBI(P) system with the equation

– $\mathbb{Q}_B = \{\nu a q | q \in \mathbb{Q}_{B_1}\}$ if $B = \nu a B_1$

and the following two inference rules

$$\frac{q \xrightarrow{a} q'}{\nu a q \xrightarrow{\varnothing} \nu a q'} \qquad \frac{a \neq b \quad q \xrightarrow{b} q'}{\nu a q \xrightarrow{b} \nu a q'}$$

The encoding of HBI(P) system with hiding can be simply handled as follows

$$[\![\nu a B]\!]_{\nu a q} = [\![B]\!]_q \ ; H^{\ulcorner w_{\iota(B)}(\{a\})\urcorner}$$

where H^α for $\alpha \in \{0,1\}^h$ is the the process inductively defined by:

$$H^0 = \mathsf{I} \qquad H^1 = \bot \qquad H^{x\alpha} = (H^x \otimes H^\alpha)$$

We remark that the term H^α replicates on the right interface only the ports of the left interface that are in the positions of α and that are set to 0, while the others are kept hidden. Consequently, $H^{\ulcorner w_{\iota(B)}(\{a\})\urcorner}$ hides the port associated to a. The extension of the correspondence results to HBI(P) systems with hiding is straightforward.

References

1. F. Arbab, R. Bruni, D. Clarke, I. Lanese, and U. Montanari. Tiles for Reo. In A. Corradini and U. Montanari, editors, *WADT*, volume 5486 of *LNCS*, pages 37–55. Springer, 2009.
2. A. Basu, M. Bozga, and J. Sifakis. Modeling heterogeneous real-time components in BIP. In *Fourth IEEE International Conference on Software Engineering and Formal Methods (SEFM 2006)*, pages 3–12. IEEE Computer Society, 2006.
3. S. Bliudze and J. Sifakis. The algebra of connectors - structuring interaction in BIP. *IEEE Trans. Computers*, 57(10):1315–1330, 2008.
4. S. Bliudze and J. Sifakis. Causal semantics for the algebra of connectors. *Formal Methods in System Design*, 36(2):167–194, 2010.
5. R. Bruni, I. Lanese, and U. Montanari. A basic algebra of stateless connectors. *Theor. Comput. Sci.*, 366(1-2):98–120, 2006.
6. R. Bruni, H. Melgratti, and U. Montanari. A connector algebra for P/T nets interactions. In J.-P. Katoen and B. Koenig, editors, *CONCUR*, volume 6901 of *LNCS*, pages 312–326. Springer, 2011.

7. R. Bruni, H. C. Melgratti, and U. Montanari. Connector algebras, Petri nets, and BIP. In E. M. Clarke, I. Virbitskaite, and A. Voronkov, editors, *Ershov Memorial Conference*, volume 7162 of *Lecture Notes in Computer Science*, pages 19–38. Springer, 2011.
8. R. Bruni, H. C. Melgratti, and U. Montanari. A survey on basic connectors and buffers. In B. Beckert, F. Damiani, F. S. de Boer, and M. M. Bonsangue, editors, *FMCO*, volume 7542 of *Lecture Notes in Computer Science*, pages 49–68. Springer, 2011.
9. R. Bruni, H. C. Melgratti, U. Montanari, and P. Sobocinski. Connector algebras for C/E and P/T nets' interactions. *Logical Methods in Computer Science*, 9(3), 2013.
10. S. Graf and S. Quinton. Contracts for BIP: Hierarchical interaction models for compositional verification. In J. Derrick and J. Vain, editors, *FORTE*, volume 4574 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2007.
11. P. Sobocinski. Representations of petri net interactions. In P. Gastin and F. Laroussinie, editors, *CONCUR*, volume 6269 of *LNCS*, pages 554–568. Springer, 2010.