Open Petri Nets: Non-deterministic Processes and Compositionality*

Paolo Baldan[1], Andrea Corradini[2], Hartmut Ehrig[3], and Barbara König[4]

[1] Dipartimento di Matematica Pura e Applicata, Università di Padova, Italy
[2] Dipartimento di Informatica, Università di Pisa, Italy
[3] Institut für Softwaretechnik und Theoretische Informatik,
Technische Universität Berlin, Germany
[4] Abteilung für Informatik und Angewandte Kognitionswissenschaft,
Universität Duisburg-Essen, Germany

Abstract. We introduce ranked open nets, a reactive extension of Petri nets which generalises a basic open net model introduced in a previous work by allowing for a refined notion of interface. The interface towards the external environment of a ranked open net is given by a subset of places designated as open and used for composition. Additionally, a bound on the number of connections which are allowed on an open place can be specified. We show that the non-deterministic process semantics is compositional with respect to the composition operation over ranked open nets, a result which did not hold for basic open nets.

Introduction

Petri nets are a well-known model of concurrent and distributed systems, widely used both in theoretical and applicative areas [14]. While the basic model is mainly aimed at representing closed, completely specified systems evolving autonomously through the firing of transitions, in recent years there has been an increasing attention to the development of reactive Petri net models, directly supporting certain features needed for modeling *open* systems, which can interact with the surrounding environment [11, 15, 12, 13, 9, 3].

In particular, open Petri nets, as introduced in [1], are a mild extension of basic nets with the possibility of interacting with the environment and of composing a larger net out of smaller open components. An open net is an ordinary net with a distinguished set of places, designated as open, through which the net can interact with the environment. As a consequence of such interaction, tokens can be freely generated and removed in open places. Open nets are endowed with a composition operation, characterised as a pushout in the corresponding category, suitable to model both interaction through open places and synchronisation of transitions.

It is very convenient if compositionality at the system level is reflected at the semantic level, i.e., if the behaviour of a system can be suitably expressed on the

* Supported by the MIUR Project ART, the DFG project Behaviour-GT and CRUI/DAAD Vigoni.

basis of the behaviour of its components. This allows for modular analysis of the systems and it helps in defining system reconfigurations (replacing a component by another) which keep the observable behaviour unchanged [2, 4].

In particular, as non-sequential processes of a Petri net can be fruitfully used as a representation of possible scenarios in the execution of a system (see, e.g., the work on workflows and the encodings of web-service description languages like OWL or BPEL as Petri nets [16, 17, 4, 18]), it can be interesting to relate the processes of a Petri net with those of its components. Specifically, one should understand under which conditions processes of the subcomponent nets can be combined into a consistent process of their composition and vice versa, how processes of the full system can be decomposed into processes of the components.

Results in this direction have been provided for open nets in [1], by showing compositionality for a semantics based on *deterministic* processes *à la* Goltz-Reisig. Unfortunately, as noticed in the same paper, the result does not extend to non-deterministic processes. To get a rough intuition of what fails, consider the open nets in the Fig. 3(b) (ignoring, for the moment, the labels 2 and ω attached to dangling arcs). The representation of the nets is standard; only open places have ingoing and/or outgoing dangling arcs, meaning that transitions of the environment could be attached and thus put and/or remove tokens in these places. The nets Z_i are simple enough to be considered processes themselves. For instance, Z_1 represents a process in which a token can be consumed either by t_1 or by the environment. When joining Z_1 and Z_2 along the net Z_0, in the result Z_3 place s is still open, the intuition being that each open place allows for an unbounded number of connections, hence adding one connection does not affect its openness. There is no way of specifying that, as a result of the composition, the open port of each of the two components is occupied by the other component, thus producing a net where place s is closed. This is problematic since a net identical to Z_3, but where place s is closed, is a valid process of Z_3 (specifying a computation having no interactions with the environment). However there is no way to obtain it as the composition of two processes of Z_1 and Z_2.

In order to overcome this problem, we introduce *ranked open nets*, a refined model of open nets where besides specifying the open places, which can be used for composition with other nets, we also specify the maximum number of allowed (input and output) connections. This provides a more expressive model, properly subsuming basic open nets (which can be seen as special ranked open nets, where open places always allow for an unbounded number of connections).

A mechanism for composing ranked open nets is defined which generalises the one for basic open nets. In this case the composition operation cannot be characterised as a colimit. Instead, it can be seen as an abstraction of a pushout in a more concrete category where ports are made explicit.

The composition operation is extended to non-deterministic processes and we prove the desired compositionality result: if a net Z_3 is the composition of Z_1 and Z_2, then any process of Z_3 can be obtained as the composition of processes of the component nets and vice versa, the composition of processes of Z_1 and Z_2, which agree on the common interface, always provides a process of Z_3.

The paper is organised as follows. In § 1 we introduce the categories of ranked open nets, and an operation of composition for such nets is defined in § 2. In § 3 we introduce non-deterministic processes for ranked open nets. In § 4 we prove the main result, i.e., compositionality for non-deterministic processes. Finally, in § 5 we draw some conclusions and directions of future investigation.

1 Ranked Open Nets

An *open net*, as introduced in [1], is an ordinary P/T Petri net with a distinguished set of places. These places are intended to represent the interface of the net towards the environment, which, interacting with the net, can "freely" add or remove some tokens in the open places. Rather than simply distinguishing between *input* and *output* places, here, for every place we specify the largest number of allowed incoming and outgoing new connections. A place is closed if it does not allow for any new connection.

Given a set X we will denote by $X^{\oplus}$ the free commutative monoid generated by X, with identity 0, and by $\mathbf{2}^X$ its powerset. Furthermore given a function $h : X \to Y$ we denote by $h^{\oplus} : X^{\oplus} \to Y^{\oplus}$ its monoidal extension, while the same symbol $h : \mathbf{2}^X \to \mathbf{2}^Y$ denotes the extension of h to sets.

A *P/T Petri net* is a tuple $N = (S, T, \sigma, \tau)$ where S is the set of places, T is the set of transitions ($S \cap T = \emptyset$) and $\sigma, \tau : T \to S^{\oplus}$ assign source and target to each transition. In this paper we will consider only *finite* Petri nets. We will denote by ${}^{\bullet}(\cdot)$ and $(\cdot)^{\bullet}$ the monoidal extensions of the functions σ and τ to functions from $T^{\oplus}$ to $S^{\oplus}$. Furthermore, given a place $s \in S$, the pre- and post-set of s are defined by ${}^{\bullet}s = \{t \in T \mid s \in t^{\bullet}\}$ and $s^{\bullet} = \{t \in T \mid s \in {}^{\bullet}t\}$.

Let N_0 and N_1 be Petri nets. A *Petri net morphism* $f : N_0 \to N_1$ is a pair of total functions $f = \langle f_T, f_S \rangle$ with $f_T : T_0 \to T_1$ and $f_S : S_0 \to S_1$, such that for all $t_0 \in T_0$, ${}^{\bullet}f_T(t_0) = {f_S}^{\oplus}({}^{\bullet}t_0)$ and $f_T(t_0)^{\bullet} = {f_S}^{\oplus}(t_0{}^{\bullet})$. The category of P/T Petri nets and Petri net morphisms will be denoted by **Net**.

We use $\mathbb{N}$ for the set of natural numbers and $\mathbb{N}^{\omega}$ for the same set extended with infinity, i.e., $\mathbb{N} \cup \{\omega\}$. Operations and relations on $\mathbb{N}^{\omega}$ are defined in the expected way, i.e., $n \leq \omega$ for each $n \in \mathbb{N}$, $\omega - n = \omega + n = \omega + \omega = \omega$ for each $n \in \mathbb{N}$, while $\omega - \omega$ is undefined. The same operators will be applied, pointwise, to functions over natural numbers. E.g., given $f, g : X \to \mathbb{N}^{\omega}$ we denote by $f + g : X \to \mathbb{N}^{\omega}$ the function defined by $(f + g)(x) = f(x) + g(x)$ for any $x \in X$.

Definition 1 (ranked open net). *A* (ranked) open net *is a pair $Z = (N_Z, o_Z)$, where $N_Z = (S_Z, T_Z, \sigma_Z, \tau_Z)$ is an ordinary P/T Petri net (called the* underlying net*) and $o_Z = (o_Z^+, o_Z^-) : S_Z \to \mathbb{N}^{\omega}$. We define $O_Z^x = \{s \in S_Z : o_Z^x(s) > 0\}$, for $x \in \{+, -\}$ and call them the sets of* input *and* output open places *of the net.*

As mentioned above, the functions o_Z^+ and o_Z^- intuitively specify for each place in S_Z the maximum number of allowed new ingoing/outgoing connections, also referred to as the *ranks* of s. In [1] whenever a place was open, intuitively there was no limit to the number of new connections. Hence the open nets of [1] can be seen as special ranked open nets, where $o_Z^x(s) \in \{0, \omega\}$ for any place s.

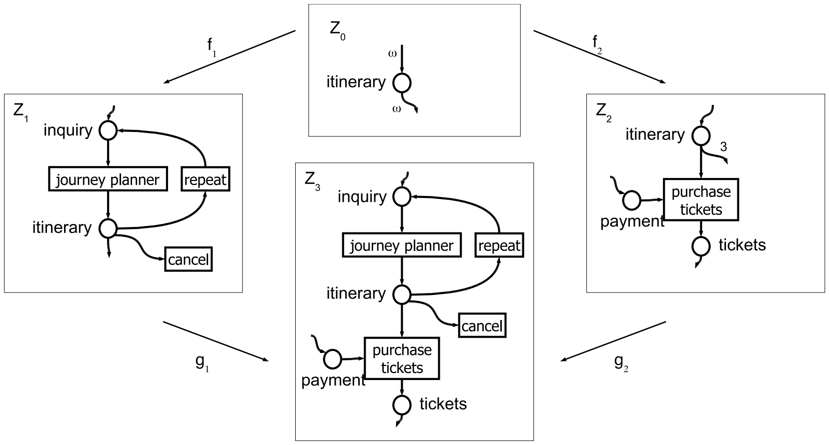

Fig. 1. Composing ranked open nets.

As an example of ranked open nets, consider net Z_3 in Fig. 1, intuitively modelling the booking of a ticket in a travel agency. In the graphical representation an input (resp. output) open place s has a dangling ingoing (resp. outgoing) arc, marked by the corresponding rank. When the rank is 1 it is omitted.

Conceptually, we can think that every place of an open net has a set of *attaching points*, which can either be used by an existing transition connected to the place, or can be free and thus usable for connecting new transitions. Sometimes, as in the definition of concrete morphisms below (Definition 6), we need to consider an explicit identity of such attaching points, that we call *ports*. A port used by a transition is identified with the transition itself, while free ports are identified by a progressive number. Most often, however, we will be interested only in their number, i.e., in the *degree* of a place. Given $n \in \mathbb{N}$, let $[n]$ denote the set $\{0 \ldots, n-1\}$. For all considered Z we assume that $T \cap \mathbb{N}^{\omega} = \emptyset$.

Definition 2 (input and ouput ports and degree). *Let Z be an open net. For any place $s \in S$ we define the sets of* input *and* output ports *of s as follows:*

$$p^{+}(s) = [o_{Z}^{+}(s)] \cup {}^{\bullet}s \quad \text{and} \quad p^{-}(s) = [o_{Z}^{-}(s)] \cup s^{\bullet}$$

The ports in $[o_{Z}^{+}(s)]$ and $[o_{Z}^{-}(s)]$ are called open ports.

Furthermore, we define the input degree *of s as $deg^{+}(s) = |p^{+}(s)|$, and, similarly, the* output degree *of s as $deg^{-}(s) = |p^{-}(s)|$.*

The token game of open nets. The notion of enabledness for transitions is the usual one, but, besides the changes produced by the firing of the transitions of the net, we consider also the interaction with the environment which is modelled by events, denoted by $+_s$ and $-_s$, which produce or consume a token in an open place s. For an open net Z, the *set of extended events*, denoted $\bar{T}_Z$, is defined as

$$\bar{T_Z} = T_Z \cup \{+_s : s \in O_Z^+\} \cup \{-_s : s \in O_Z^-\}.$$

Pre- and post-set functions are extended by defining ${}^\bullet{+_s} = 0$ and ${+_s}^\bullet = s$, and symmetrically, ${}^\bullet{-_s} = s$ and ${-_s}^\bullet = 0$.

Definition 3 (firing). *Let Z be an open net. A* firing *in Z consists of the execution of an extended event $\epsilon \in \bar{T_Z}$, i.e., $u \oplus {}^\bullet\epsilon\ [\epsilon\rangle\ u \oplus \epsilon^\bullet$.*

A firing can be (i) the execution of a transition $u \oplus {}^\bullet t\ [t\rangle\ u \oplus t^\bullet$, with $t \in T_Z$; (ii) the creation of a token by the environment $u\ [+_s\rangle\ u \oplus s$, with $s \in O_Z^+$; (iii) the deletion of a token by the environment $u \oplus s\ [-_s\rangle\ u$, with $s \in O_Z^-$.

Morphisms of open nets. Morphisms of open nets will be defined as standard net morphisms satisfying suitable conditions on the place ranks. Intuitively, a morphism $f : Z_1 \to Z_2$ "inserts" net Z_1 into a larger net Z_2, allowing a place s of Z_1 to be connected to "new" transitions, i.e., transitions in $Z_2 \setminus f(Z_1)$. The condition we impose guarantees that each new connection of s and each open port of $f(s)$ can be mapped to an open port of s.

For reasons discussed in § 2, we define two kinds of morphisms. In the more abstract ones, we impose only a cardinality constraint, while in the more concrete ones we require an explicit mapping relating, for each place s of Z_1, the ports of s to those of $f(s)$. We next formalise the idea of "new connections" of a place.

Definition 4 (in-set and out-set of a place along a morphism). *Given open nets Z_1 and Z_2 and a Petri net morphism $f : N_{Z_1} \to N_{Z_2}$, for each place $s_1 \in S_1$ the* in-set of s_1 along f *is defined as* $\mathsf{in}(f)(s_1) = \{{}^\bullet f_S(s_1) - f_T({}^\bullet s_1)\}$, *and similarly the* out-set *is* $\mathsf{out}(f)(s_1) = \{f_S(s_1)^\bullet - f_T(s_1{}^\bullet)\}$*. This defines the functions* $\mathsf{in}(f), \mathsf{out}(f) : S_1 \to \mathbf{2}^{T_2}$*.*

The functions $\#\mathsf{in}(f), \#\mathsf{out}(f) : S_1 \to \mathbb{N}$ are defined, respectively, as $\#\mathsf{in}(f)(s_1) = |\mathsf{in}(f)(s_1)|$ and $\#\mathsf{out}(f)(s_1) = |\mathsf{out}(f)(s_1)|$.

Definition 5 (open net morphisms). *An* open net morphism $f : Z_1 \to Z_2$ *is a Petri net morphism $f : N_{Z_1} \to N_{Z_2}$ such that*

$$\textit{(i) } \#\mathsf{in}(f) + o_2^+ \circ f_s \le o_1^+ \textit{ and} \qquad \textit{(ii) } \#\mathsf{out}(f) + o_2^- \circ f_s \le o_1^- .$$

A morphism f is called an open net embedding *if both f_T and f_S are injective.*

Intuitively, condition (i) requires that the number of new incoming transitions added to $s \in S_1$ in the target net Z_2 plus the input connections which are still allowed for $f_S(s)$ in Z_2 must be bounded by the maximum number of allowed input connections for s. Examples of open net embeddings can be found in Fig. 1. The mappings are those suggested by the labelling of the nets.

Definition 6 (concrete morphisms). *Let Z_1 and Z_2 be open nets. A* concrete open net morphism $\mathbf{f} : Z_1 \to Z_2$ *is a pair $\mathbf{f} = \langle f, \{f_{s_1}\}_{s_1 \in S_1}\rangle$, where $f : N_{Z_1} \to N_{Z_2}$ is a Petri net morphism and for any $s_1 \in S_1$, f_{s_1} consists of a pair of partial surjections $f_{s_1}^x : p^x(s_1) \to p^x(f(s_1))$ for $x \in \{+, -\}$, consistent with f, i.e., satisfying, for any $t \in {}^\bullet s_1$, $f_{s_1}^+(t) = f(t)$ and for any $t \in s_1{}^\bullet$, $f_{s_1}^-(t) = f(t)$.*

A morphism $\mathbf{f}$ *is called an* open net embedding *if all components are injective.*

As anticipated, concrete morphisms explicitly relate, for each place $s \in S_1$, the ports of $f(s)$ and of s, using the component f_s. In the sequel, instead of f_s^+ and f_s^-, when place s is clear from the context, we will often write f^+ and f^-. Moreover, when defining f_s^x we will only specify its values on the open ports $[o^x_{Z_1}(s)]$, which, as f_s^x must be consistent with f, completely determines f_s^x.

As expected, the two notions of morphism just introduced determine two categories related by an obvious forgetful functor. In fact, given a concrete morphism $\mathbf{f} = \langle f, \{f_{s_1}\}_{s_1 \in S_1} \rangle \colon Z_1 \to Z_2$ it is straightforward to check that the Petri net morphism f satisfies conditions (i) and (ii) of Definition 5.

Definition 7 (open nets categories). *We denote by* $\mathbf{ONet}_r$ *the category of ranked open nets and open net morphisms, and by* $\mathbf{ONet}_c$ *the category having the same objects and concrete open net morphisms as arrows.*

Furthermore, we denote by $U \colon \mathbf{ONet}_c \to \mathbf{ONet}_r$ *the forgetful functor which is the identity on objects, and acts on an arrow* $\mathbf{f} = \langle f, \{f_s\} \rangle$ *as* $U(\mathbf{f}) = f$.

Sometimes, categories $\mathbf{ONet}_c$ and $\mathbf{ONet}_r$ will be referred to as the *concrete* and the *abstract* category of (ranked) open nets, respectively.

The category of basic open nets introduced in [1] is (isomorphic to) the full subcategory of $\mathbf{ONet}_r$ including all the nets Z such that for any place s we have $o^x_Z(s) \in \{0, \omega\}$, i.e., either s is closed or it allows for an unbounded number of connections. In the following this subcategory will be referred to as $\mathbf{ONet}$.

2 Composing Open Nets

Intuitively, two open nets Z_1 and Z_2 are composed by specifying a common subnet Z_0, and then by joining the two nets along Z_0. Composition will be characterised as a pushout in the concrete category of open nets $\mathbf{ONet}_c$. But since for specification purposes the abstract category $\mathbf{ONet}_r$ is often more appropriate and easier to deal with, next we will focus on the notion of composition induced on such category by the colimit based composition in $\mathbf{ONet}_c$.

Composition is possible if it respects the interface of the involved nets. This is formalised by the notion of composability of a span of embeddings in $\mathbf{ONet}_c$.

Definition 8 (composable span in $\mathbf{ONet}_c$). *A span of embeddings* $\mathbf{f}_1 : Z_0 \to Z_1$ *and* $\mathbf{f}_2 : Z_0 \to Z_2$ *in* $\mathbf{ONet}_c$ *is called* composable *if, for any* $s_0 \in S_0$

1. *for all* $i \in [o^+_{Z_0}(s_0)]$, *if* $f_1^+(i) \in \mathsf{in}(f_1)(s_0)$ *then* $f_2^+(i) \in [o^+_{Z_2}(f_2(s_0))]$
2. *for all* $i \in [o^-_{Z_0}(s_0)]$, *if* $f_1^-(i) \in \mathsf{out}(f_1)(s_0)$ *then* $f_2^-(i) \in [o^-_{Z_2}(f_2(s_0))]$

plus the analogous conditions, exchanging the roles of Z_1 *and* Z_2.

Intuitively, condition (1) says that, given a place s_0 and an open input port $i \in [o^+_{Z_1}(s_0)]$, if according to $\mathbf{f}_1$ the transition $f_1^+(i) \in \mathsf{in}(f_1)(s_0)$ is going to be attached to this port, then the corresponding port in Z_2 must be open, i.e., $f_2^+(i) \in [o^+_{Z_2}(f_2(s_0))]$. The other conditions are analogous.

$$\begin{tikzcd}
 & Z_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\
Z_1 \arrow[dr, "g_1"'] & & Z_2 \arrow[dl, "g_2"] \\
 & Z_3 &
\end{tikzcd}$$

Fig. 2. Composition of ranked open nets.

Given a concrete composable span $Z_1 \xleftarrow{\mathbf{f}_1} Z_0 \xrightarrow{\mathbf{f}_2} Z_2$, the *composition* of Z_1 and Z_2 along Z_0 is the open net Z_3 (see Fig. 2) obtained as the pushout of $\mathbf{f}_1$ and $\mathbf{f}_2$, which exists by the next result.

Proposition 9 (pushout in $\mathbf{ONet}_c$). *A span of embeddings $\mathbf{f}_1 : Z_0 \to Z_1$, $\mathbf{f}_2 : Z_0 \to Z_2$ in $\mathbf{ONet}_c$ is composable if and only if it has a pushout $Z_1 \xrightarrow{\mathbf{g}_1} Z_3 \xleftarrow{\mathbf{g}_2} Z_2$ in $\mathbf{ONet}_c$, whose underlying diagram is a pushout in* $\mathbf{Net}$.

The construction of the pushout of a composable span in $\mathbf{ONet}_c$ turns out to be quite complex and it is not reported for space limitations, but the intuition is simple. Firstly, the underlying net N_{Z_3} is obtained as the pushout of N_{Z_1} and N_{Z_2} along N_{Z_0} in $\mathbf{Net}$. Next, if a place is not in Z_0, then in the pushout it maintains exactly its ports. Instead, for a place s in Z_0, in the pushout the ports of the image of s are obtained by taking the pushout of the ports of the images of s in Z_1 and Z_2. Since mappings between ports can be partial, open ports can disappear. A port is open only if it is open in both nets Z_1 and Z_2.

The notions of composability of spans and of composition between nets can be transferred to the abstract category via the forgetful functor $U\colon \mathbf{ONet}_c \to \mathbf{ONet}_r$. More interestingly, these notions can be defined also directly at the abstract level, by referring only to the ranks of places of the involved nets. Thanks to this fact, in the rest of the paper we will be able to work in the abstract category only, which provides a simpler and natural framework to be used for specification purposes. Still, we stress here that we defined the composition of nets in the concrete category first, because the corresponding notion in the abstract category cannot be characterized by a universal property as a pushout.

Given a pair of embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ in $\mathbf{ONet}_r$, we say that they are *composable* if there exists a composable span of embeddings $\mathbf{f}_1 : Z_0 \to Z_1$ and $\mathbf{f}_2 : Z_0 \to Z_2$ in $\mathbf{ONet}_c$ such that $U(\mathbf{f}_1) = f_1$ and $U(\mathbf{f}_2) = f_2$.

Fact 10 (composable span in $\mathbf{ONet}_r$). A span of embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ in $\mathbf{ONet}_r$ is composable if and only if

1. $\#\mathsf{in}(f_1) \leq o^+_{Z_2} \circ f_2$ and $\#\mathsf{out}(f_1) \leq o^-_{Z_2} \circ f_2$;
2. $\#\mathsf{in}(f_2) \leq o^+_{Z_1} \circ f_1$ and $\#\mathsf{out}(f_2) \leq o^-_{Z_1} \circ f_1$.

Intuitively, the first half of condition (1) requires that the number of input connections which are added to each place s of Z_0 by f_1, namely $\#\mathsf{in}(f_1)(s)$, is bounded by the number of additional input connections allowed for $f_2(s)$ in Z_2, i.e., $o^+_{Z_2}(f_2(s))$. The remaining conditions are similar.

Now, given a composable span of embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ in $\mathbf{ONet}_r$, let $\langle \mathbf{f}_1, \mathbf{f}_2 \rangle$ be any pair of composable embeddings in $\mathbf{ONet}_c$ such that $U(\mathbf{f}_1) = f_1$ and $U(\mathbf{f}_2) = f_2$. Then the *composition* of Z_1 and Z_2 along Z_0 in $\mathbf{ONet}_r$ is defined exactly as their composition in $\mathbf{ONet}_c$, i.e., as the pushout object of $\mathbf{f}_1$ and $\mathbf{f}_2$ in $\mathbf{ONet}_c$. It can be shown that this definition is well given, and that it can be characterized as follows.

Fact 11 (composition in $\mathbf{ONet}_r$). Let $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ be a span of embeddings in $\mathbf{ONet}_r$. Compute the pushout of the corresponding diagram in the category $\mathbf{Net}$ obtaining the net N_{Z_3} and the morphisms g_1 and g_2. For $i \in \{1, 2\}$, define $res_i^+(s_3) = o^x_{Z_i}(s_i) - \#\mathsf{in}(g_i)(s_i)$ if there is some $s_i \in S_i$ such that $g_i(s_i) = s_3$ and $res_i^+(s_3) = \omega$, otherwise.[5] The function res_i^- is defined in a dual way. Then take, for $x \in \{+, -\}$

$$o^x_{Z_3} = \min\{res_1^x, res_2^x\}$$

Then Z_3 (with morphisms g_1 and g_2) is the composition along Z_0 of f_1 and f_2.

Intuitively, for a place $s_3 = g_i(s_i)$, the value $res_i^+(s_i)$ is obtained by subtracting from the number of connections allowed for s_i, i.e., $o^+_{Z_i}(s_i)$, the number of connections which have been added as an effect of the composition, i.e., $\#\mathsf{in}(g_i)(s_i)$. In other words $res_i^+(s_i)$ is the residual number of allowed connections. When joining two places, the number of allowed connections for the resulting place will be the minimum among the residuals of the two original places.

Two simple examples of composition can be found in Fig. 3. It is worth explaining why, for example, diagram (a) is not a pushout in $\mathbf{ONet}_r$. In fact, since Z_1 and Z_2 are isomorphic, we can close the span $Z_1 \overset{f_1}{\leftarrow} Z_0 \overset{f_2}{\rightarrow} Z_2$ with arrows $Z_1 \overset{id}{\rightarrow} Z_1 \overset{\cong}{\leftarrow} Z_2$ obtaining a commutative square in $\mathbf{ONet}_r$, but there is no mediating morphism $Z_3 \to Z_1$ because the counter-image of an open place cannot be closed. For a more complex example see Fig. 1, where two nets Z_1 and Z_2 representing the planning of a trip and the buying of the ticket, respectively, are composed. Note, e.g., that place itinerary in Z_2 is output open with rank 3 and input open with rank 1, as needed for adding the connections in Z_1.

3 Processes of Open Nets

A process of an open net is an open net itself, satisfying suitable acyclicity and conflict freeness requirements, together with a mapping to the original net.

The open net underlying a process is an open *occurrence* net, namely an open net K such that the underlying net N_K is an ordinary occurrence net, with some additional conditions on open places. Fig. 6 shows some examples of occurrence nets. The open places in the occurrence net are intended to represent occurrences of tokens which are produced or consumed by the environment in the considered computation. Hence, input open places must satisfy $o^+(s) = 1$ and

[5] Observe that res_i^+ is well-defined since g_i is injective.

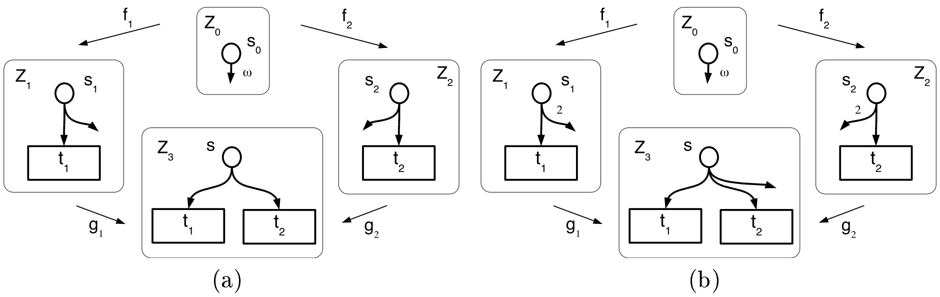

Fig. 3. Composing ranked open nets.

additionally they must be minimal. In fact, an input open place with $o^+(s) > 1$ would represent a token possibly produced by two different transitions in the environment; similarly an input open place in the post-set of some transition would represent a token which can be produced either internally or by some transition in the environments. In both cases the situation would correspond to a backward conflict and it would prevent one to interpret the place as a token occurrence. Instead, an output open place can be in the pre-set of a transition, as it happens for place itinerary in the open occurrence nets K_1 and K_2 of Fig. 6, and it might be that $o^-_Z(s) > 1$. The idea is that the token occurrence represented by place s can be consumed either by transition t or by two or more occurrences of transitions in the environment.

For a Petri net $N = (S, T, \sigma, \tau)$ the *causality relation* $<_N \subseteq (S \cup T)^2$ is the least transitive relation such that $x <_N y$ if $y \in x^\bullet$. Moreover, the *conflict relation* $\#_N \subseteq (S \cup T)^2$ is the least symmetric relation generated by the rules:

$$\frac{{}^\bullet t \cap {}^\bullet t' \neq \emptyset \qquad t \neq t' \qquad t, t' \in T}{t \,\#_N\, t'} \qquad\qquad \frac{x \,\#_N\, y \qquad y <_N y'}{x \,\#_N\, y'} \ (\mathsf{hereditarity})$$

These definitions lift to open nets by considering the underlying net. We will omit the subscripts when clear from the context.

Definition 12 (open occurrence net). *An* open occurrence net *is an open net K such that*

1. *${}^\bullet t$ and $t^\bullet$ are sets rather than proper multisets, for each transition $t \in T$;*
2. *the causality relation $<_K$ is a finitary strict partial order;*
3. *the conflict relation $\#_K$ is irreflexive;*
4. *there are no* backward conflicts, *i.e., $deg^+(s) \leq 1$ for each place $s \in S$.*

Notice that the net N_K underlying an open occurrence net is an occurrence net according to the standard definition.

We next introduce the notion of process for open nets.

Definition 13 (open net process). *A* process *of an open net Z is a mapping $\pi : K \to Z$ where K is an open occurrence net and $\pi : N_K \to N_Z$ is a Petri net morphism, such that $\pi_S(O^+_K) \subseteq O^+_Z$ and $\pi_S(O^-_K) \subseteq O^-_Z$.*

Note that the mapping from the occurrence net K to the original net Z, is *not* an open net morphism in general. In fact, the process mapping, differently from open net morphisms, must be a simulation, i.e., it must preserve the behaviour. To this aim the image of an open place in K must be an open place in Z, since tokens can be produced (consumed) by the environment only in input (output) open places of Z. Instead, there is no relation between the rank of open places in the occurrence net and in the net Z since a token in an open place can be consumed by distinct occurrences of the same transition in the environment.

We next introduce the category of processes, where objects are processes and arrows are pairs of open net morphisms.

Definition 14 (category of processes). *We denote by* **Proc** *the category where objects are processes and given two processes* $\pi_0 : K_0 \to Z_0$ *and* $\pi_1 : K_1 \to Z_1$*, an arrow* $\psi : \pi_0 \to \pi_1$ *is a pair of open net morphisms* $\psi = \langle \psi_Z : Z_0 \to Z_1, \psi_K : K_0 \to K_1 \rangle$ *such that the diagram on the right (indeed the underlying diagram in* **Net***) commutes.*

$$\begin{tikzcd}
K_0 \arrow[r, "\psi_K"] \arrow[d, "\pi_0"'] & K_1 \arrow[d, "\pi_1"] \\
Z_0 \arrow[r, "\psi_Z"'] & Z_1
\end{tikzcd}$$

3.1 Projecting Behaviours along Embeddings

Since open net morphisms are designed to capture the idea of “insertion” of a net into a larger one, they are expected to “reflect” the behaviour in the sense that given $f : Z_0 \to Z_1$, the behaviour of Z_1 can be projected along the morphism to the behaviour of Z_0. As in [1], this intuition can be formalised for open net embeddings by showing how a process of Z_1, as defined before, can be projected along f giving a process of Z_0. Intuitively, each possible computation in Z_1 can be “projected” to Z_0, by considering only the part of the computation of the larger net which is visible in the smaller net. Ranks are defined correspondingly.

Definition 15 (projection of a process). *Let* $f : Z_0 \to Z_1$ *be an open net embedding and let* $\pi_1 : K_1 \to Z_1$ *be a process of* Z_1*. A* projection of π_1 along f, *is a pair* $\langle \pi_0, \psi \rangle$ *where* $\pi_0 : K_0 \to Z_0$ *is a process of* Z_0 *and* $\psi : \pi_0 \to \pi_1$ *is an arrow in* **Proc**, *constructed as follows. Consider the pullback of* π_1 *and* f *in* **Net**, *thus obtaining the net morphisms* π_0 *and* ψ_K *(see the diagram on the right). Then* K_0 *is obtained by taking* N_{K_0} *as underlying net, and defining*

$$o^+_{K_0} = o^+_{K_1} \circ \psi_K + \#\mathsf{in}(\psi_K) \quad \text{and} \quad o^-_{K_0} = o^-_{K_1} \circ \psi_K + \#\mathsf{out}(\psi_K)$$

(i.e., by opening the places as least as possible to make $\psi_K : K_0 \to K_1$ *an open net morphism) and* $\psi = \langle \psi_K, f \rangle$*.*

$$\begin{tikzcd}
N_{K_1} \arrow[r, "\pi_1"] & N_{Z_1} \\
N_{K_0} \arrow[u, dotted, "\psi_K"] \arrow[r, dotted, "\pi_0"'] & N_{Z_0} \arrow[u, "f"']
\end{tikzcd}$$

4 Composing Non-deterministic Processes

Consider a composition diagram in $\mathbf{ONet}_r$, as in Fig. 2, where f_1 and f_2 are open net embeddings. One would like to establish a clear relationship among the

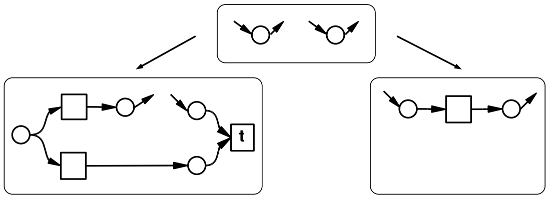

Fig. 4. Transition t would be in self-conflict in the composition.

behaviours of the involved nets. Roughly, we would like that the behaviour of Z_3 could be constructed "compositionally" out of the behaviours of Z_1 and Z_2.

In [1] we have shown that in the setting of basic open nets this can be done only for deterministic processes. Here we show how, in the setting of ranked open nets, the result extends to general, possibly non-deterministic processes. Given two processes π_1 of Z_1 and π_2 of Z_2 which "agree" on Z_0, one can construct a process π_3 of Z_3 by amalgamating π_1 and π_2. Vice versa, each process π_3 of Z_3 can be projected over two processes π_1 and π_2 of Z_1 and Z_2, which can be amalgamated to produce π_3 again. Hence, all and only the processes of Z_3 can be obtained by amalgamating the processes of the components Z_1 and Z_2.

4.1 Composition of Non-deterministic Occurrence Open Nets

A basic step towards the composition operation is the formalisation of the intuitive idea of processes of different nets which "agree" on a common part. Concretely, this amounts to identify suitable conditions which ensure that the composition of occurrence open nets exists and produces a net in the same class.

First, given a span $K_1 \overset{f_1}{\leftarrow} K_0 \overset{f_2}{\rightarrow} K_2$ we introduce the notion of causality relation induced by K_1 and K_2 over K_0. When the two nets are composed their causality relations get "fused". Hence, to ensure that the resulting net is again an occurrence net, the induced causality must be a strict partial order.

Definition 16 (induced causality). *Let $K_1 \overset{f_1}{\leftarrow} K_0 \overset{f_2}{\rightarrow} K_2$ be a span of embeddings in $\mathbf{ONet}_r$, where K_i ($i \in \{0,1,2\}$) are occurrence open nets. The relation of causality $<_{1,2}$ induced over K_0 by K_1 and K_2, through f_1 and f_2 is the least transitive relation such that for any x_0, y_0, if $f_1(x_0) <_{K_1} f_1(y_0)$ or $f_2(x_0) <_{K_2} f_2(y_0)$ then $x_0 <_{1,2} y_0$.*

When composing non-deterministic occurrence nets, which can include mutual exclusive branches of computation, we must also avoid that transitions becomes non-firable due to the creation of self-conflicts. For example, Fig. 4 shows a span where the induced causality is a strict partial order, but there would be a self-conflict on t in the composed occurrence net. Hence t would not be firable in any computation of the net.

To this aim, we introduce new relations, called *anti-causality* and *anti-conflict*. Intuitively, two items x and y in K are related by anti-causality (anti-conflict) if, to ensure the firability of each transition in the net, x and y *must* remain causally unrelated (not in conflict, resp.) when K is composed with other nets. Then the idea is to avoid compositions which can lead to situations in which two items are related both by a relation and by the corresponding anti-relation.

Definition 17 (anti-relations). *Let K be an occurrence open net. The* anti-causality $\neg{<_K}$ *and* anti-conflict $\neg\#_K$ *relations over $(S \cup T)^2$ are defined by the following rules (subscripts are omitted as clear from the context):*

$$\frac{}{x \neg\# x \quad x \neg< x}\ (\mathsf{anti1}) \qquad \frac{x \neg\# y \quad x' < x}{x' \neg\# y}\ (\mathsf{anti2})$$

$$\frac{x \neg\# y \quad x \# y'}{y' \neg< y}\ (\mathsf{anti3}) \qquad \frac{x \neg\# y}{y \neg\# x}\ (\mathsf{anti4})$$

The rules have a clear interpretation. Rule $(\mathsf{anti1})$ states that the each single item must remain concurrent, while rules $(\mathsf{anti2})$ and $(\mathsf{anti3})$ are obtained by "reverting" the rule which expresses hereditarity of conflict w.r.t. causality. Finally, $(\mathsf{anti4})$ states that $\neg\#$ is symmetric.

Given an open net morphism $f_1 : K_0 \to K_1$, where K_0 and K_1 are occurrence nets, in the following we will use the symbols $<_1$, $\#_1$, $\neg\#_1$ and $\neg<_1$ to denote the *projection over* K_0 of the corresponding relations over K_1, i.e., for any $r \in \{<, \#, \neg\#, \neg<\}$ and x_0, y_0 in K_0 we will write

$$x_0\ r_1\ y_0 \quad \text{iff} \quad f_1(x_0)\ r_{K_1}\ f_1(y_0)$$

Given a span of occurrence open nets $K_1 \overset{f_1}{\leftarrow} K_0 \overset{f_2}{\rightarrow} K_2$ we next define the conflict relation and the anti-relations induced over the net K_0 by K_1 and K_2, through f_1 and f_2. This has been already done for causality in Definition 16, where induced causality $<_{1,2}$ is defined as the transitive closure of $<_1 \cup <_2$.

Definition 18 (induced relations). *Let $K_1 \overset{f_1}{\leftarrow} K_0 \overset{f_2}{\rightarrow} K_2$ be a span in* $\mathbf{ONet}_r$, *where K_i ($i \in \{0, 1, 2\}$) are occurrence open nets. The conflict relation and the anti-relations induced over K_0 by K_1 and K_2, through f_1 and f_2 are as follows.*

For x_0, y_0 in K_0, let $x_0 \searrow_1 y_0$ be a shortcut for $x_0 <_1 y_0$ and there is no z_0 such that $x_0 <_{K_0} z_0 \leq_1 y_0$. Observe that in this case x_0 must be a place, connected to y_0 through a chain of transitions in K_1, but not in K_0. The notation $x_0 \searrow_2 y_0$ is defined in the dual way.

- ***induced conflict*** $\#_{1,2}$*: The relation $\#_{1,2}$ over K_0 is the least relation, hereditary w.r.t. $<_{1,2}$ such that, for any x_0, y_0,*
  1) if $x_0 \#_1 y_0$ or $x_0 \#_2 y_0$ then $x_0 \#_{1,2} y_0$.
  2) if $x_0 \searrow_1 y_0$ and $x_0 \searrow_2 z_0$ then $y_0 \#_{1,2} z_0$.
- ***induced anti-relations*** $\neg<_{1,2}$ ***and*** $\neg\#_{1,2}$*: The relations $\neg\#_{1,2}$ and $\neg<_{1,2}$ over K_0 are defined as the least relations such that for x_0, y_0, for $i \in \{1, 2\}$, if $x \neg\#_i y$ then $x \neg\#_{1,2} y$, and similarly, if $x \neg<_i y$ then $x \neg<_{1,2} y$, and closed under rules* $(\mathsf{anti1}) - (\mathsf{anti4})$.

Now we can identify the conditions which guarantee that the composition of two occurrence open nets is still an occurrence open net.

Definition 19 (consistent span). *A span $K_1 \xleftarrow{f_1} K_0 \xrightarrow{f_2} K_2$ of occurrence open nets is* consistent *if it is composable in $\mathbf{ONet}_r$ and for any x_0, y_0 in K_0*

1. $x_0 \neg\!<_{1,2} y_0 \;\Rightarrow \neg(x_0 <_{1,2} y_0)$ *and* $x_0 \neg\#_{1,2} y_0 \;\Rightarrow \neg(x_0 \#_{1,2} y_0)$*;*
2. *for* $i, j \in \{1, 2\}$, $i \neq j$, *we have that* $x_0 \neg\#_i y_0$ *implies* $\neg(x_0 \searrow_j y_0)$*.*

Condition (1) just requires that each anti-relation does not intersect the corresponding relation. Condition (2), instead, just imposes that two anti-conflictual places in K_1 are never connected by a chain of transitions in K_2 (and vice versa), otherwise in the composition one would get a self-conflict.

We can now show that the composition in $\mathbf{ONet}_r$ of a consistent span of occurrence nets produces an occurrence net. We first need a preliminary result.

Lemma 20. *Let $K_1 \xleftarrow{f_1} K_0 \xrightarrow{f_2} K_2$ be a composable span of embeddings in $\mathbf{ONet}_r$, where K_i ($i \in \{0, 1, 2\}$) are occurrence open nets, and let $K_1 \xrightarrow{g_1} K_3 \xleftarrow{g_2} K_2$ be the composition. Then for any x_0, y_0 in K_0, if we let $x_3 = g_1(f_1(x_0)) = g_2(f_2(x_0))$ and $y_3 = g_1(f_1(y_0)) = g_2(f_2(y_0))$, we have*

1. $x_0 <_{1,2} y_0$ *iff* $x_3 <_{K_3} y_3$*;*
2. $x_0 \#_{1,2} y_0$ *iff* $x_3 \#_{K_3} y_3$*;*
3. $x_0 \neg\#_{1,2} y_0$ *iff* $x_3 \neg\#_{K_3} y_3$*;*
4. $x_0 \neg\!<_{1,2} y_0$ *iff* $x_3 \neg\!<_{K_3} y_3$*.*

Proposition 21. *In the hypotheses of Lemma 20 above, $K_1 \xleftarrow{f_1} K_0 \xrightarrow{f_2} K_2$ is a consistent span* iff *the composition K_3 is an occurrence open net.*

4.2 Amalgamating Non-deterministic Processes

For the rest of this section we refer to a fixed composition in $\mathbf{ONet}_r$, as in Fig. 2, where f_1 and f_2 are composable open net embeddings. Two processes π_1 of Z_1 and π_2 of Z_2 can be amalgamated when they agree on the common subnet Z_0.

Definition 22 (agreement of non-deterministic processes). *We say that two non-deterministic processes $\pi_1 : K_1 \to Z_1$ and $\pi_2 : K_2 \to Z_2$* agree *on Z_0 if there are projections $\langle \pi_0, \psi_K^i \rangle$ along f_i of π_i for $i \in \{1, 2\}$ such that the span $K_1 \xleftarrow{\psi_K^1} K_0 \xrightarrow{\psi_K^2} K_2$ is consistent and, for any s_0 in K_0, if $s_3 = f_i(g_i(\pi_0(s_0)))$ is the corresponding place in Z_3, the following holds:*

$$\text{if} \quad \#\mathsf{out}(\psi_K^1)(s_0) + \#\mathsf{out}(\psi_K^2)(s_0) < o^-_{K_0}(s_0) \quad \text{then} \quad s_3 \in O^-_{Z_3}. \tag{1}$$

In this case $\langle \pi_0, \psi_K^1 \rangle$, $\langle \pi_0, \psi_K^2 \rangle$ are called agreement projections *for π_1 and π_2.*

Intuitively, the two processes agree if they have the same projection over Z_0. Additionally, as required by condition (1), if, for a place s_0 in K_0, the number of external events that can consume the token in s_0 exceeds the events provided by Z_1 and Z_2 then the corresponding place in Z_3 must be open.

$$\begin{tikzcd}
 & K_0 \arrow[dl, "\psi^1_K"'] \arrow[dr, "\psi^2_K"] \arrow[d, "\pi_0"] & \\
K_1 \arrow[d, "\pi_1"] \arrow[dr, dotted, "\phi^1_K"'] & Z_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & K_2 \arrow[d, "\pi_2"] \arrow[dl, dotted, "\phi^2_K"] \\
Z_1 \arrow[dr, "g_1"'] & K_3 \arrow[d, dotted, "\pi_3"] & Z_2 \arrow[dl, "g_2"] \\
 & Z_3 &
\end{tikzcd}$$

Fig. 5. Amalgamation of open net processes.

Definition 23 (process amalgamation). *Let $\pi_i : K_i \to Z_i$ ($i \in \{0,1,2,3\}$) be non-deterministic processes and let $\langle \pi_0, \psi^1_K \rangle$ and $\langle \pi_0, \psi^2_K \rangle$ be agreement projections of π_1 and π_2 along f_1 and f_2 (see Fig. 5). We say that π_3 is an* amalgamation *of π_1 and π_2, written $\pi_3 = \pi_1 +_{\psi^1_K, \psi^2_K} \pi_2$, if there are projections $\langle \pi_1, \phi^1 \rangle$ and $\langle \pi_2, \phi^2 \rangle$ of π_3 over Z_1 and Z_2, respectively, such that the upper square is a composition in $\mathbf{ONet}_r$.*

We next give a more constructive characterisation of process amalgamation, which also proves that the result is unique up to isomorphism.

Lemma 24 (amalgamation construction). *Let $\pi_1 : K_1 \to Z_1$ and $\pi_2 : K_2 \to Z_2$ be non-deterministic processes that agree on Z_0, and let $\langle \pi_0, \psi^1_K \rangle$ and $\langle \pi_0, \psi^2_K \rangle$ be corresponding agreement projections. Then the* amalgamation *$\pi_1 +_{\psi^1_K, \psi^2_K} \pi_2$ is a process $\pi_3 : K_3 \to Z_3$, where net K_3 is obtained as the composition in $\mathbf{ONet}_r$ of $\psi^1_K : K_0 \to K_1$ and $\psi^2_K : K_0 \to K_2$ and the process mapping $\pi_3 : K_3 \to Z_3$ is uniquely determined by the universal property of the underlying pushout diagram in $\mathbf{Net}$ (see Fig. 5). Hence $\pi_1 +_{\psi^1_K, \psi^2_K} \pi_2$ is unique up to isomorphism.*

As an example, in Fig. 6 a process for the net Z_3 of Fig. 1 is obtained as the amalgamation of processes of the component nets. The process for Z_1 represents a reservation activity, which can succeed after two attempts or can be finally cancelled. In the process for Z_2 two possible itineraries are visible: the first one can only be discarded (used by the environment) while the second one can also trigger a payment, thus resulting in a ticket. Composing the two processes one gets a full booking process for net Z_3.

We next show that each non-deterministic process of a composed net arises as the amalgamation of non-deterministic processes of the components.

Lemma 25 (process decomposition). *Let $\pi_3 : K_3 \to Z_3$ be a process of Z_3 and, for $i \in \{1,2\}$, let $\langle \pi_i, \phi^i \rangle$ be projections of π_3 along g_i. Then there are agreement projections $\langle \pi_0, \psi^1_K \rangle$, $\langle \pi_0, \psi^2_K \rangle$ of π_1, π_2 such that $\pi_3 \cong \pi_1 +_{\psi^1_K, \psi^2_K} \pi_2$.*

As a consequence we finally have our main result.

Theorem 26 (compositionality for non-deterministic processes). *All and only the non-deterministic processes of Z_3 can be obtained as amalgamations of processes of Z_1 and Z_2 which agree on Z_0.*

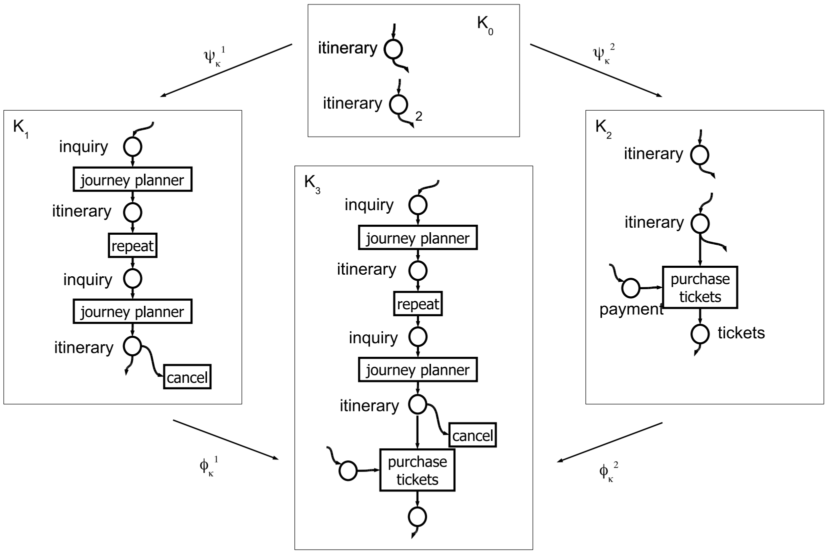

Fig. 6. An example of process amalgamation.

5 Conclusions and Future Work

We have introduced a compositional semantics based on non-deterministic processes for ranked open nets, an extension of the basic open net model of [1] where it is possible to specify, for open places, the maximum number of allowed connections. The composition operation is characterised as a pushout in a category of ranked open nets with concrete morphisms. The notion of agreement between processes of different sub-components, which is a requirement for process composition, builds upon a theory of anti-relations (i.e., anti-causality and anti-conflict) which could have an interest for Petri nets in general.

We believe that a theory of non-deterministic processes for open nets can represent a starting point for a modular verification of open nets based on finite prefixes of the unfolding [10]. There are obvious difficulties, e.g., the fact that open nets are always infinite state (whenever they have at least one input open place). However the "regularity" of the state space suggests the possibility of undertaking a symbolic approach, for which analogous work for standard Petri nets, like [6], could provide an inspiration.

We foresee also potential outcomes in the setting of graph transformation systems. In fact graph transformation systems can be seen as generalisation of Petri nets, and it has been often productive to focus first in the latter simpler setting. The notion of openness [8, 7] as well as the notion of processes [5] have already been studied in the setting of graph transformation, however until now there have been no attempts to combine them. The present work can be a first step in this direction.

References

1. P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional semantics for open Petri nets based on deterministic processes. *Mathematical Structures in Computer Science*, 15(1):1–35, 2005.
2. P. Baldan, A. Corradini, H. Ehrig, R. Heckel, and B. König. Bisimilarity and behaviour-preserving reconfigurations of open Petri nets. In *Proc. of CALCO'07*, volume 4624 of *LNCS*, pages 126–142. Springer, 2007.
3. T. Basten. *In terms of nets: System design with Petri nets and process algebra.* PhD thesis, Eindhoven University of Technology, 1998.
4. F. Bonchi, A. Brogi, S. Corfini, and F. Gadducci. A behavioural congruence for web services. In *Proc. of FSEN '07*, volume 4727 of *LNCS*, pages 240–256. Springer, 2007.
5. A. Corradini, U. Montanari, and F. Rossi. Graph processes. *Fundamenta Informaticae*, 26:241–265, 1996.
6. J. Desel, G. Juhás, and C. Neumair. Finite unfoldings of unbounded Petri nets. In *Proc. of ATPN'04*, volume 3099 of *LNCS*, pages 157–176. Springer, 2004.
7. H. Ehrig and B. König. Deriving bisimulation congruences in the DPO approach to graph rewriting. In *Proc. of FoSSaCS'04*, volume 2987 of *LNCS*, pages 151–166. Springer, 2004.
8. R. Heckel. *Open Graph Transformation Systems: A New Approach to the Compositional Modelling of Concurrent and Reactive Systems.* PhD thesis, Technische Universität Berlins, 1998.
9. E. Kindler. A compositional partial order semantics for Petri net components. In *Application and Theory of Petri Nets*, volume 1248 of *LNCS*, pages 235–252. Springer, 1997.
10. K.L. McMillan. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In *Proc. of CAV'92*, volume 663 of *LNCS*, pages 164–174. Springer, 1992.
11. R. Milner. Bigraphical reactive systems. In *Proc. of CONCUR'01*, volume 2154 of *LNCS*, pages 16–35. Springer, 2001.
12. M. Nielsen, L. Priese, and V. Sassone. Characterizing Behavioural Congruences for Petri Nets. In *Proc. of CONCUR'95*, volume 962 of *LNCS*, pages 175–189. Springer, 1995.
13. L. Priese and H. Wimmel. A uniform approach to true-concurrency and interleaving semantics for Petri nets. *Theoretical Computer Science*, 206(1–2):219–256, 1998.
14. W. Reisig. *Petri Nets: An Introduction.* EATCS Monographs on Theoretical Computer Science. Springer, 1985.
15. V. Sassone and P. Sobociński. A congruence for Petri nets. In *Proc. of PNGT'04*, number 127(2) in Electronic Notes in Theoretical Computer Science, pages 107–120. Elsevier Science, 2005.
16. W. van der Aalst. The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
17. W. van der Aalst. Interorganizational workflows: An approach based on message sequence charts and Petri nets. *System Analysis and Modeling*, 34(3):335–367, 1999.
18. G. Zhang, F. Meng, C. Jiang, and J. Pang. Using Petri Net to Reason with Rule and OWL. In *Proc. of CIT '06*, page 42. IEEE Computer Society, 2006.