# Bisimilarity and Behaviour-Preserving Reconfigurations of Open Petri Nets*

Paolo Baldan[1], Andrea Corradini[2], Hartmut Ehrig[3], Reiko Heckel[4], and Barbara König[5]

[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] Department of Computer Science, University of Leicester, UK
[5] Abteilung für Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen, Germany

**Abstract.** We propose a framework for the specification of behaviour-preserving reconfigurations of systems modelled as Petri nets. The framework is based on open nets, a mild generalisation of ordinary Place/Transition nets suited to model open systems which might interact with the surrounding environment and endowed with a colimit-based composition operation. We show that natural notions of (strong and weak) bisimilarity over open nets are congruences with respect to the composition operation. We also provide an up-to technique for facilitating bisimilarity proofs. The theory is used to identify suitable classes of reconfiguration rules (in the double-pushout approach to rewriting) whose application preserves the observational semantics of the net.

## Introduction

Petri nets are a well-known model of concurrent and distributed systems, widely used both in theoretical and applicative areas [19]. In classical approaches, nets are intended to represent closed, completely specified systems evolving autonomously through the firing of transitions. Therefore, ordinary Petri nets do not support directly certain features that are needed to model *open* systems, namely systems which can interact with the surrounding environment or, in a different view, systems which are only partially specified.

Firstly, a large (possibly still open) system is typically built out of smaller open components. Syntactically, an open system is equipped with suitable interfaces, over which the interaction with the external environment can take place. Semantically, openness can be represented by defining the behaviour of a component as if it were embedded in general environments, determining any possible interaction over the interfaces.

* Research partially supported by the EU IST-2004-16004 SEnSOria, the MIUR Project ART, the DFG project SANDS and CRUI/DAAD Vigoni "Models based on Graph Transformation Systems: Analysis and Verification".

Secondly, often the building components of an open system are not statically determined, but they can change during the evolution of the system, according to predefined reconfiguration rules triggered by internal or external solicitations.

In this paper we present a framework where open systems can be modelled as Petri nets. Observational semantics based on (weak) bisimulation are shown to be congruences with respect to the composition operation defined over Petri nets. Building on this, suitable reconfigurations of such systems can be specified as net rewritings, which preserve the behaviour of the system.

The framework is based on so-called *open nets*, a mild generalisation of ordinary Petri nets introduced in [2, 3] to answer the first of the requirements above, i.e., 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 surrounding environment. As a consequence of such interaction, tokens can be freely generated and removed in open places. In the mentioned papers 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.

In the first part of the paper, after having extended the existing theory for open nets to deal with *marked* nets, we introduce bisimulation-based observational equivalences for open nets. Following the intuition about reactive systems discussed in [12], such equivalences are based on the observation of the interactions between the given net and the surrounding environment. The framework treats uniformly *strong bisimilarity*, where every transition firing is observed, and *weak bisimilarity*, where a subset of unobservable transition labels is fixed and the firings of transitions carrying such labels are considered invisible. Bisimilarity is shown to be a congruence with respect to the composition operation over open nets. Interestingly enough, this holds also when the set of non-observable labels is not empty, i.e., for weak bisimilarity: some natural questions regarding the relation with weak bisimilarity in CCS are also addressed. In addition, we also define an up-to technique for facilitating bisimulation proofs.

Exploiting the results in the first part of the paper we introduce a framework for open net reconfigurations. The fact that open net components are combined by means of categorical colimits, suggests a setting for specifying net reconfigurations, based on double-pushout (DPO) rewriting [9]. Using the congruence result for bisimilarity we identify classes of transformation rules which ensure that reconfigurations of the system do not affect its observational behaviour.

A concluding section discusses some related work. A full version of the paper, with proofs and additional results, is available as [4].

## 1 Marked open nets

An *open net*, as introduced in [2, 3], is an ordinary P/T Petri net with a distinguished set of *open places*, which represent the interface through which the environment can interact with the net. An open place can be an *input place*, meaning that the environment can put tokens into it, or an *output place*, from which the environment can remove tokens, or both. In this section we introduce the basic notions for open nets as presented in [3], generalising them to nets with initial marking: this will be needed in the treatment of bisimilarity in Section 3.

Given a set $X$ we write $\mathbf{2}^X$ for the powerset of $X$ and $X^\oplus$ for the free commutative monoid over $X$. Moreover, given a function $h : X \to Y$ we denote by the same symbol $h : \mathbf{2}^X \to \mathbf{2}^Y$ its extension to sets, and by $h^\oplus : X^\oplus \to Y^\oplus$ its monoidal extension. Given a multiset $u \in X^\oplus$, with $u = \bigoplus_{x \in X} u_x \cdot x$, for $x \in X$ we will write $u(x)$ to denote the coefficient $u_x$. The symbol 0 denotes the empty multiset.

**Definition 1 (multiset projection).** *Given a function $f : X \to Y$ and a multiset $u \in Y^\oplus$ we denote by $(u \downarrow f)$ the projection of $u$ along $f$, which is the multiset over $X$ defined as $(u \downarrow f) = \bigoplus_{x \in X} u_{f(x)} \cdot x$.*

For instance, given $f : \{s_0, s_1, s_2\} \to \{s'_1, s'_2, s'_3\}$ such that $f(s_0) = f(s_1) = s'_1$ and $f(s_2) = s'_2$, we have $(2s'_1 \oplus s'_2 \oplus s'_3 \downarrow f) = 2s_0 \oplus 2s_1 \oplus s_2$. We will mainly work with injective functions, for which the projection operation satisfies some expected properties, such as $f^\oplus((u \downarrow f)) \leq u$ and $(f^\oplus((u \downarrow f)) \downarrow f) = (u \downarrow f)$.

We consider nets where transitions are labelled over a fixed set of labels $\Lambda$.

**Definition 2 (P/T Petri net).** *A* P/T Petri net *is a tuple $N = (S, T, \sigma, \tau, \lambda)$ where $S$ is the set of places, $T$ is the set of transitions, $\sigma, \tau : T \to S^\oplus$ are functions mapping each transition to its pre- and post-set and $\lambda : T \to \Lambda$ is a labelling function for transitions.*

In the sequel we will denote by ${}^\bullet(\cdot)$ and $(\cdot)^\bullet$ the monoidal extensions of the functions $\sigma$ and $\tau$ to functions from $T^\oplus$ to $S^\oplus$. Moreover, given $s \in S$, the pre- and post-set of $s$ are defined by ${}^\bullet s = \{t \in T : s \in t^\bullet\}$ and $s^\bullet = \{t \in T : s \in {}^\bullet t\}$.

**Definition 3 (Petri net category).** *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 $t_0 \in T_0$, ${}^\bullet f_T(t_0) = f_S^\oplus({}^\bullet t_0)$, $f_T(t_0)^\bullet = f_S^\oplus(t_0{}^\bullet)$ and $\lambda_1(f_T(t_0)) = \lambda_0(t_0)$. The category of P/T Petri nets and Petri net morphisms is denoted by* **Net**.

We next introduce the notion of open net. As anticipated above, differently from [2, 3], we work here with marked nets.

**Definition 4 (open net).** *An* open net *is a pair $Z = (N_Z, O_Z)$, where $N_Z = (S_Z, T_Z, \sigma_Z, \tau_Z, \lambda_Z)$ is a P/T Petri net and $O_Z = (O_Z^+, O_Z^-) \in \mathbf{2}^{S_Z} \times \mathbf{2}^{S_Z}$ are the sets of input and output open places of the net. A* marked *open net is a pair $(Z, \hat{u})$ where $Z$ is an open net and $\hat{u} \in S_Z^\oplus$ is the initial marking.*

Hereafter, unless stated otherwise, all open nets will be implicitly assumed to be marked. An open net will be denoted simply by $Z$ and the corresponding initial marking by $\hat{u}$. Subscripts carry over to the net components. The graphical representation for open nets is similar to that for standard nets. In addition, the fact that a place is input or output open is represented by an ingoing or outgoing dangling arc, respectively. For instance, in net $Z_1$ of Fig. 1, place $s$ is both input and output open, while $s'$ is only output open.

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 $s$.

**Definition 5 (set of extended events).** *Let $Z$ be an open net. The* set of extended events *of $Z$, denoted by $\bar{T}_Z$ and ranged over by $\epsilon$ is defined as*

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

*Defining ${}^\bullet{+_s} = 0$ and ${+_s}^\bullet = s$, and symmetrically, ${}^\bullet{-_s} = s$ and ${-_s}^\bullet = 0$, the notion of pre- and post-set extends to multisets of extended events.*

Given a marking $u \in {O_Z^+}^\oplus$, we denote by $+_u$ the multiset $\bigoplus_{s \in S} u(s) \cdot +_s$. Similarly, $-_u = \bigoplus_{s \in S} u(s) \cdot -_s$ for $u \in {O_Z^-}^\oplus$.

**Definition 6 (firings and steps).** *Let $Z$ be an open net. A* step *in $Z$ consists of the execution of a multiset of (extended) events $A \in \bar{T}_Z^\oplus$, i.e., $u \oplus {}^\bullet A\ [A\rangle\ u \oplus A^\bullet$. A step is called a* firing *when it consists of a single event, i.e., $A = \epsilon \in \bar{T}_Z$.*

A firing can be (i) the execution of a transition $u \oplus {}^\bullet t\ [t\rangle\ u \oplus t^\bullet$, with $u \in S_Z^\oplus$, $t \in T_Z$; (ii) the creation of a token by the environment $u\ [+_s\rangle\ u \oplus s$, with $s \in O_Z^+$, $u \in S_Z^\oplus$; (iii) the deletion of a token by the environment $u \oplus s\ [-_s\rangle\ u$, with $u \in S_Z^\oplus$, $s \in O_Z^-$. A step is the firing of a multiset of transitions and interactions with the environment, of the kind $A \oplus -_w \oplus +_v$ for $A \in T_Z^\oplus, w \in {O_Z^-}^\oplus$ and $v \in {O_Z^+}^\oplus$.

**Definition 7 (open net category).** *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, if we define $\mathsf{in}(f) = \{s \in S_1 : {}^\bullet f_S(s) - f_T({}^\bullet s) \neq \emptyset\}$ and $\mathsf{out}(f) = \{s \in S_1 : f_S(s)^\bullet - f_T(s^\bullet) \neq \emptyset\}$, then*

1. *(i) $f_S^{-1}(O_2^+) \cup \mathsf{in}(f) \subseteq O_1^+$ and (ii) $f_S^{-1}(O_2^-) \cup \mathsf{out}(f) \subseteq O_1^-$.*
2. *$\hat{u}_1 = (\hat{u}_2 \downarrow f_S)$ (*reflection of initial marking*).*

*The morphism $f$ is called an* open net embedding *if both $f_T$ and $f_S$ are injective. We will denote by* **ONet** *the category of open nets and open net morphisms.*

Intuitively, an embedding $f : Z_1 \to Z_2$ "inserts" net $Z_1$ into a larger net $Z_2$, which might constrain the behaviour of $Z_1$. Conditions 1.(i) and 1.(ii) first require that open places are reflected and hence that places which are "internal" in $Z_1$ cannot be promoted to open places in $Z_2$. Furthermore, they ensure that the context in which $Z_1$ is inserted can interact with $Z_1$ only through the open places. In fact, if $s$ is a place of $Z_1$ and its image $f_S(s)$ is in the post-set of a transition of $Z_2$ which is not in the image of $Z_1$, from the perspective of $Z_1$ the environment can generate tokens in $s$; in this case $s \in \mathsf{in}(f)$, and thus Condition 1.(i) requires $s$ to be an input place. Condition 1.(ii) is analogous for output places. Finally, condition 2 requires that the marking of $Z_1$ is the projection of the marking of $Z_2$: any place $s_1 \in S_1$ must carry the same number of tokens as its image $f(s_1) \in S_2$, i.e., $\hat{u}_1(s_1) = \hat{u}_2(f(s_1))$ for any $s_1 \in S_1$. All morphisms $f_1$, $f_2$, $\alpha_1$ and $\alpha_2$ in Fig. 1 are examples of open net embeddings (the mappings on places and transitions are those suggested by the shape and labelling of the nets).

It is worth observing that most of the constructions in the paper will be defined for open net embeddings, hence readers can limit their attention to embeddings if this helps the intuition. Still, on the formal side, working in a larger host category with more general morphisms is essential to obtain a characterisation of the composition operation in terms of pushouts. Specifically, non-injective open net morphisms are needed as mediating morphisms (recall, for example, that the category of sets with injective functions does not have all pushouts).

In the sequel, given an open net morphism $f = \langle f_S, f_T \rangle : Z_1 \to Z_2$, to lighten the notation we will omit the subscripts "$S$" and "$T$" in its place and transition components, writing $f(s)$ for $f_S(s)$ and $f(t)$ for $f_T(t)$. Moreover we will write $f^{\oplus} : \bar{T}^{\oplus}_{Z_1} \to \bar{T}^{\oplus}_{Z_2}$ to denote the monoidal function defined on the generators by $f^{\oplus}(t) = f(t)$ for $t \in T_{Z_1}$ and, for $x \in \{+, -\}$, $f^{\oplus}(x_s) = x_{f(s)}$, if $f(s) \in O^x_2$ and $f^{\oplus}(x_s)$ undefined, otherwise. Note that $f^{\oplus}$ can be partial since open places can be mapped to closed places.

Unlike most of the morphisms considered over Petri nets in the literature, open net morphisms are *not* simulations. Instead, since open net embeddings are designed to capture the idea of inserting a net into a larger one, they are expected to reflect the behaviour, in the sense that given an embedding $f : Z_0 \to Z_1$, the behaviour of $Z_1$ can be projected along $f$ to the behaviour of $Z_0$.

To formalise reflection of the behaviour along open nets embeddings, we define the projection operation also over steps.

**Definition 8 (projecting extended events).** *Given an open net embedding $f : Z \to Z'$ and an extended event $\epsilon' \in \bar{T}_{Z'}$ we define the* projection of $\epsilon'$ along $f$ *as follows:*

- *if $\epsilon' = t' \in T_{Z'}$ is a transition then*
$$(t' \Downarrow f) = \begin{cases} t & \text{if } t \in T_Z \text{ and } f(t) = t' \\ -_{(\bullet t' \downarrow f)} \oplus +_{(t'^{\bullet} \downarrow f)} & \text{if } t' \notin f(T_Z) \end{cases}$$
- *if $\epsilon' = x_{s'}$, with $x \in \{+, -\}$, then $(x_{s'} \Downarrow f) = x_{(s' \downarrow f)}$.*

*The projection operation over multisets of extended events $(\_ \Downarrow f) : \bar{T}^{\oplus}_{Z'} \to \bar{T}^{\oplus}_{Z}$, is defined as the monoidal extension of the projection of firings.*

In words, if we think of the embedding as an inclusion, given a transition $t'$, the projection $(t' \Downarrow f)$ is the transition itself if $t'$ is in $Z$. Otherwise, if $t'$ is not in $Z$ but it consumes or produces tokens in places of $Z$, the projection of $t'$ contains the corresponding extended events, expressing the interactions over open places.

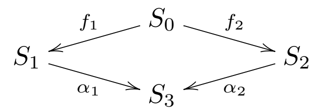
**Lemma 1 (reflection of behaviour).** *Let $f : Z \to Z'$ be an open net embedding. For every step $u'\ [A'\rangle\ v'$ in $Z'$ there is a step $(u' \downarrow f)\ [(A' \Downarrow f)\rangle\ (v' \downarrow f)$ in $Z$, called the* projection *of the step $u'\ [A'\rangle\ v'$ over $Z$.*

## 2 Composing open nets

We introduce next a basic mechanism for composing open nets which is characterised as a pushout construction in the category of open nets. The case of unmarked nets was already discussed in [3]. Here we extend the theory to deal with marked 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 $Z_0$.

Let us start with a technical definition which will be useful below.

**Proposition 1 (composition of multisets).** *Consider a pushout diagram in the category of sets as below, where all morphisms are injective. Given $u_1 \in S_1^{\oplus}$ and $u_2 \in S_2^{\oplus}$ such that $(u_1 \downarrow f_1) = (u_2 \downarrow f_2) = u_0$, there is a (unique) multiset $u_3 \in S_3^{\oplus}$ such that $(u_3 \downarrow \alpha_i) = u_i$, for $i \in \{1, 2\}$. Such a multiset $u_3$ will be denoted by $u_3 = u_1 \uplus_{u_0} u_2$.*




As in [2, 3], two embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ are called *composable* if the places which are used as interface by $f_1$, i.e., the places $\mathsf{in}(f_1)$ and $\mathsf{out}(f_1)$, are mapped by $f_2$ to input and output open places of $Z_2$, respectively, and also the symmetric condition holds.

**Definition 9 (composability).** *Let $f_1 : Z_0 \to Z_1$, $f_2 : Z_0 \to Z_2$ be embeddings in* **ONet**. *We say that $f_1$ and $f_2$ are* composable *if 1. $f_2(\mathsf{in}(f_1)) \subseteq O^+_{Z_2}$ and $f_2(\mathsf{out}(f_1)) \subseteq O^-_{Z_2}$; and 2. $f_1(\mathsf{in}(f_2)) \subseteq O^+_{Z_1}$ and $f_1(\mathsf{out}(f_2)) \subseteq O^-_{Z_1}$.*

Composability is necessary and sufficient to ensure that the pushout of $f_1$ and $f_2$ can be computed in **Net** and then lifted to **ONet**.

**Proposition 2 (pushouts in ONet).** *Let $f_1 : Z_0 \to Z_1$, $f_2 : Z_0 \to Z_2$ be embeddings in* **ONet** *(see Fig. 2(a)). Compute the pushout of the corresponding diagram in category* **Net** *(componentwise on places and transitions) obtaining net $N_{Z_3}$ and morphisms $\alpha_1$ and $\alpha_2$, and then take as open places, for $x \in \{+, -\}$,*

$$O^x_{Z_3} = \{s_3 \in S_3 : \alpha_1^{-1}(s_3) \subseteq O^x_{Z_1} \ \wedge \ \alpha_2^{-1}(s_3) \subseteq O^x_{Z_2}\}$$

*and as marking $\hat{u}_3 = \hat{u}_1 \uplus_{\hat{u}_0} \hat{u}_2$, defined according to Proposition 1. Then $(\alpha_1, Z_3, \alpha_2)$ is the pushout in* **ONet** *of $f_1$ and $f_2$ if and only if $f_1$ and $f_2$ are composable. In this case we write $Z_3 = Z_1 +_{f_1, f_2} Z_2$.*

As an example, the open net embeddings $f_1$ and $f_2$ in Fig. 1 are composable and $Z_3$ is the resulting pushout object.

We next analyse the behaviour of an open net $Z_3$ arising as the composition of two nets $Z_1$ and $Z_2$ along an interface $Z_0$. More specifically, we show that steps of the component nets $Z_1$ and $Z_2$ can be "composed" to give a step of $Z_3$ when they agree on the interface and satisfy suitable compatibility conditions.

**Lemma 2 (composing steps).** *Let $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ be composable embeddings in* **ONet** *and let $Z_3 = Z_1 +_{f_1, f_2} Z_2$ (see Fig. 2(a)). Let $u_1\ [A_1\rangle\ v_1$ and $u_2\ [A_2\rangle\ v_2$ be steps in $Z_1$ and $Z_2$, respectively, such that $(u_1 \downarrow f_1) = (u_2 \downarrow f_2) = u_0$ and $A_2 = f_2^{\oplus}((A_1 \Downarrow f_1))$.*

*Then, $(v_1 \downarrow f_1) = v_0 = (v_2 \downarrow f_2)$ and, if we define $A_3 = \alpha_1^{\oplus}(A_1)$,*

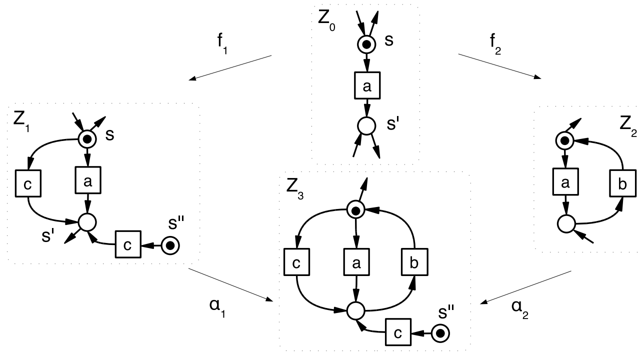



**Fig. 1.** An example of a pushout in **ONet**.

$$u_1 \uplus_{u_0} u_2 \ [A_3\rangle \ v_1 \uplus_{v_0} v_2.$$

The above result can be used to get a compositionality result for steps, showing that the steps of $Z_3$ can be obtained by "composing" steps of the components $Z_1$ and $Z_2$ satisfying suitable compatibility requirements. However, this is outside the main focus of the paper and can be found in the full version [4].

## 3 Bisimilarity of open nets

We next study (strong and weak) bisimilarity for open nets, proving that it is a congruence with respect to the colimit-based composition of open nets.

First, we define the labelled transition system associated to an open net. Net transitions carry a label which is observed when they fire. Additionally, in the labelled transition system we also observe what happens at the open places. As discussed in the conclusions, this resembles the labelled transition system arising from the view of Petri nets as reactive systems in [14, 20]. More precisely, given an open net $Z$, the corresponding labelled transition system has the markings of the net as states. Transitions are generated by the firings of $Z$ and labelled over the set $\Lambda_Z = \Lambda \cup \{+_s : s \in O^+_Z\} \cup \{-_s : s \in O^-_Z\}$.

For notational convenience we extend the labelling function $\lambda_Z$ to the set of extended events $\bar{T}_Z$, by defining $\lambda_Z(x) = x$ for $x \in \bar{T}_Z - T_Z$ (i.e., for $x = +_s$ or $x = -_s$ with $s \in S_Z$).

**Definition 10 (lts for an open net).** *The* labelled transition system *associated to an open net $Z$, denoted by $\mathsf{lts}(Z)$, is the pair $\langle S^{\oplus}_Z, \rightarrow_Z\rangle$, where states are markings $u_Z \in S^{\oplus}_Z$ and the transition relation $\rightarrow_Z \subseteq S^{\oplus}_Z \times \Lambda_Z \times S^{\oplus}_Z$ includes all transitions $u_Z \xrightarrow{\lambda_Z(x)}_Z u'_Z$ such that there is a firing $u_Z\ [x\rangle\ u'_Z$ in $Z$.*

When observing the behaviour of a system, usually only a subset of events is considered observable. Here this is formalised by selecting a subset of labels representing internal firings, playing a role similar to $\tau$-actions in process calculi, and then considering a corresponding notion of weak bisimilarity. Let $\Lambda_\tau \subseteq \Lambda$ be a subset of *unobservable* labels, fixed for the rest of the paper. Given a $\Lambda$-labelled open net $Z$, for markings $v, v' \in S_Z^\oplus$ we write $v \overset{\tau}{\rightsquigarrow}_Z v'$ if $v \xrightarrow{\ell}_Z v'$ with $\ell \in \Lambda_\tau$, and $v \overset{\ell}{\rightsquigarrow}_Z v'$ if $v \xrightarrow{\ell}_Z v'$ with $\ell \in \Lambda_Z - \Lambda_\tau$. Then we define

- $v \overset{\tau}{\Longrightarrow}_Z v'$ when $v \overset{\tau}{\rightsquigarrow}{}^*_Z v'$.
- $v \overset{\ell}{\Longrightarrow}_Z v'$ when $v \ \overset{\tau}{\rightsquigarrow}{}^*_Z \overset{\ell}{\rightsquigarrow}_Z \overset{\tau}{\rightsquigarrow}{}^*_Z \ v' \qquad \ell \neq \tau$.

Weak bisimilarity is now defined in a standard way (but note that when the set of unobservable labels is empty, this actually corresponds to strong bisimilarity). Only, we need to specify for each open place of one net which is the corresponding open place in the other net. Given two open nets $Z_1$ and $Z_2$ a *correspondence* $\eta : O_1 \leftrightarrow O_2$ between $Z_1$ and $Z_2$ is a bijection $\eta : O_1^+ \cup O_1^- \to O_2^+ \cup O_2^-$ such that for $s_1 \in O_1$, $x \in \{+, -\}$ we have $s_1 \in O_1^x$ iff $\eta(s_1) \in O_2^x$.

**Definition 11 ((weak) bisimilarity).** *Let $Z_1$, $Z_2$ be open nets and $\eta : O_1 \leftrightarrow O_2$ be a correspondence between $Z_1$ and $Z_2$. A (weak) $\eta$-bisimulation over $Z_1$ and $Z_2$ is a relation over markings $\mathcal{R} \subseteq S_1^\oplus \times S_2^\oplus$ such that if $(u_1, u_2) \in \mathcal{R}$ then*

- *if $u_1 \overset{\ell}{\rightsquigarrow}_{Z_1} u_1'$, there exists $u_2'$ such that $u_2 \overset{\eta(\ell)}{\Longrightarrow}_{Z_2} u_2'$ and $(u_1', u_2') \in \mathcal{R}$;*
- *the symmetric condition holds;*

*where $\eta(+_s) = +_{\eta(s)}$, $\eta(-_s) = -_{\eta(s)}$, and $\eta(\ell) = \ell$ for any $\ell \in \Lambda \cup \{\tau\}$.*

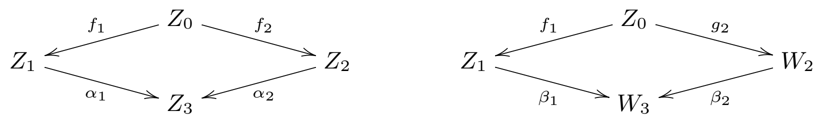
*Two open nets $Z_1$ and $Z_2$ are* (weakly) $\eta$-bisimilar*, denoted $Z_1 \approx_\eta Z_2$, if $\eta : O_1 \leftrightarrow O_2$ is a correspondence and there exists a (weak) $\eta$-bisimulation $\mathcal{R}$ over $Z_1$ and $Z_2$ such that $(\hat{u}_1, \hat{u}_2) \in \mathcal{R}$. We will say that $Z_1$ and $Z_2$ are (weakly) bisimilar, written $Z_1 \approx Z_2$, if $Z_1 \approx_\eta Z_2$ for some correspondence $\eta$.*

According to the following lemma, which is a corollary of Lemma 2, given *composable* embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$, the firing of a transition in $Z_2$, projected along $f_2$ to $Z_0$ can then be simulated in $Z_1$.

**Lemma 3.** *Let $Z_0$, $Z_1$, $Z_2$ be open nets and let $f_i : Z_0 \to Z_i$ ($i \in \{1, 2\}$) be composable embeddings, as in Fig. 2(a). Furthermore, let $Z_3 = Z_1 +_{f_1, f_2} Z_2$. Assume that $u_2 \xrightarrow{\ell}_{Z_2} u_2'$ where $\ell \in \Lambda$, let $t \in T_2$ such that $\lambda_2(t) = \ell$ and $u_2\ [t\rangle\ u_2'$, let $u_0\ [A_0\rangle\ u_0'$ be its projection over $Z_0$ (hence $A_0 = (t \Downarrow f_2)$), and let $u_0 \xrightarrow{\ell_1}_{Z_0} \dots \xrightarrow{\ell_n}_{Z_0} u_0'$ be any sequence of transitions in $\mathsf{lts}(Z_0)$ arising as a linearisation of such step in $Z_0$. Then for any $u_1 \in S_1^\oplus$ such that $(u_1 \downarrow f_1) = u_0$ we have that $u_1 \xrightarrow{\ell_1}_{Z_1} \dots \xrightarrow{\ell_n}_{Z_1} u_1'$ and $u_1 \uplus_{u_0} u_2 \xrightarrow{\ell}_{Z_3} u_1' \uplus_{u_0'} u_2'$.*

Note that above, if transition $t$ is in the image of $Z_0$, then the sequences of transitions in $\mathsf{lts}(Z_0)$ and $\mathsf{lts}(Z_1)$ are actually single firings. Otherwise, they are sequences of interactions over open places, possibly of length greater than one.

By exploiting this lemma we can prove that bisimilarity is a congruence with respect to the composition operation on open nets.




(a) (b)

**Fig. 2.** Pushouts in **ONet**.

**Theorem 1 (bisimilarity is a congruence).** *Let $Z_0$, $Z_1$, $Z_2$, $W_2$ be open nets. Let $Z_2 \approx_\eta W_2$, for some $\eta$. Consider the nets $Z_3 = Z_1 +_{f_1,f_2} Z_2$ and $W_3 = Z_1 +_{f_1,g_2} W_2$, as in Fig. 2 where $f_1$, $f_2$ and $g_2$ are embeddings, $f_1$ and $f_2$ are composable, and $f_1$ and $g_2$ are composable as well.*

*If $g_2|_{O_0} = \eta \circ (f_2|_{O_0})$ (i.e., $f_2$ and $g_2$ are consistent with $\eta$ on open places) then $Z_3 \approx_{\eta'} W_3$, where $\eta'$ is defined as follows: for all $s \in O_{Z_3}$, $\eta'(s) = \beta_1(s')$ if $s = \alpha_1(s')$, and $\eta'(s) = \beta_2(\eta(s'))$ if $s = \alpha_2(s')$.*

We next provide a kind of *up-to technique* for open net bisimilarity. Given an open net $Z$, let us define the *out-degree* of a place $s \in S$ as the maximum number of tokens that the firing of an extended event can remove from $s$, formally:

$$deg(s) = \max\left(\{({}^\bullet t)(s) : t \in T_Z\} \cup \{1 : s \in O_Z^-\}\right)$$

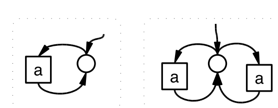
The idea, formalised in the notion of up-to bisimulation, is to allow tokens to be removed from open input places, when they exceed the out-degree of the place. More precisely, given a net $Z$ and a marking $u \in S^\oplus$, let us say that a marking $v \in {O_Z^+}^\oplus$ is *subtractable* from $u$ if $\forall s \in O_Z^+.\ deg(s) \leq u(s) - v(s)$. Note that this implies that all transitions enabled in $u$ are also enabled in $u \ominus v$.

**Definition 12 (up-to bisimulation).** *Let $Z_1$ and $Z_2$ be open nets, and let $\eta : O_1 \leftrightarrow O_2$ be a correspondence between $Z_1$ and $Z_2$. A relation $\mathcal{R} \subseteq S_1^\oplus \times S_2^\oplus$ between markings is called an* up-to $\eta$-bisimulation *if whenever $(u_1, u_2) \in \mathcal{R}$ then*

- *if $u_1 \overset{\ell}{\rightsquigarrow}_{Z_1} u_1'$, then there exist markings $u_2'$ such that $u_2 \overset{\eta(\ell)}{\Longrightarrow}_{Z_2} u_2'$, and $v_1 \in {O_1^+}^\oplus$ subtractable from $u_1$, with $(u_1' \ominus v_1, u_2' \ominus \eta^\oplus(v_1)) \in \mathcal{R}$;*
- *the symmetric condition holds.*

**Proposition 3.** *Let $Z_1$ and $Z_2$ be open nets, and let $\eta : O_1 \leftrightarrow O_2$ be a correspondence between $Z_1$ and $Z_2$. Let $\mathcal{R}$ be an up-to $\eta$-bisimulation. Then for any $(u_1, u_2) \in \mathcal{R}$ we have that $(Z_1, u_1) \approx_\eta (Z_2, u_2)$.*

As it often happens with up-to techniques, the above result might allow to show that two nets are bisimilar by exhibiting finite relations (while bisimulations are typically infinite). E.g., consider the open nets on the right, where label $a$ is observable. Then a bisimulation would include at least the pairs $\{(k \cdot s, k \cdot s) : k \in \mathbb{N}\}$, where $s$ is the only place. Instead, according to the definition above $\{(0,0), (s,s)\}$ is an up-to bisimulation.



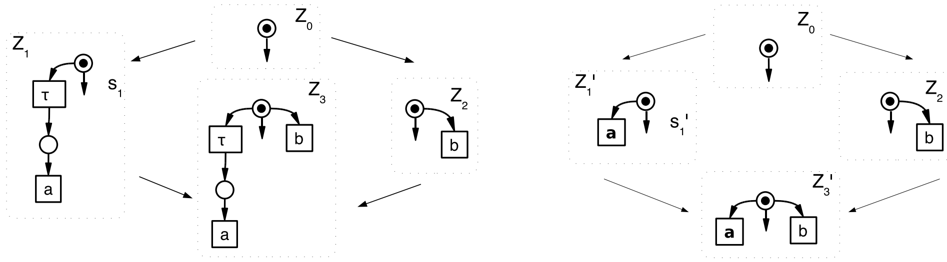



**Fig. 3.** Two pushouts of open nets for the comparison to CCS.

*Comparison to CCS.* We now give some hints as to why weak bisimilarity is a congruence in the case of open nets, but not in CCS [16]. Remember that a classical counterexample for CCS is as follows: $p_1 = \tau.a.0 \approx a.0 = p_2$, but $q_1 = \tau.a.0 + b.0 \not\approx a.0 + b.0 = q_2$. The reason for the latter inequality is that $q_1$ can do a $\tau$ and become $a.0$, while $q_2$ cannot mimic this step.

Fig. 3 shows a similar situation of nondeterministic choice for open nets, where $\tau$ is the only unobservable label. However, note that here the two nets $Z_1$ (corresponding to $\tau.a.0$) and $Z_1'$ (corresponding to $a.0$) are *not* weakly bisimilar. Whenever the $\tau$-transition is fired in $Z_1$, resulting in the marking $m_1$, this can not be mimicked in $Z_1'$ by staying idle, since then in $Z_1'$ a transition with label $-_{s_1'}$ is possible, while a transition labelled $-_{s_1}$ is not possible for the net $Z_1$ with marking $m_1$. Also note that the places $s_1$ respectively $s_1'$ must be output open in order to allow composition with the net $Z_2$.

Roughly, this means that for open nets we are always able to observe the first invisible action in an open component, which is reminiscent of the definition of observation congruence (denoted by $\approx^c$) in CCS: two processes $p, q$ are called observation congruent if they are weakly bisimilar, with the additional constraint that whenever the first step of $p$ is a $\tau$-action, then it has to be answered by at least one $\tau$-action of $q$ (and vice versa). In both settings it is only the first $\tau$-action that can be observed but not the subsequent ones.

## 4 Reconfigurations of open nets

The results in the previous sections are used here to design a framework where a system specified as a (possibly open) Petri net can be reconfigured dynamically by transformation rules, triggered by the state/shape of the system. The congruence result allows to characterise classes of reconfigurations which preserve the observational behaviour of the system.

The fact that the composition operation over open nets is defined in terms of a pushout construction suggests naturally a way of reconfiguring open nets by using the double-pushout approach to rewriting [9].

A *rewriting rule* over open nets consists of a pair of morphisms in **ONet**:

$$p = L_p \overset{l_p}{\leftarrow} K_p \overset{r_p}{\rightarrow} R_p$$

where $L_p$, $K_p$, $R_p$ are open nets, called *left-hand side*, *interface* and *right-hand side* of the rule $p$, and $l_p$, $r_p$ are open net embeddings. Furthermore, it is required that $(r_p \circ l_p^{-1})|_{O_{L_p}}$ is a correspondence between $L_p$ and $R_p$, which we denote by $\eta_p : L_p \leftrightarrow R_p$. Intuitively, the rule specifies that, given a net $Z$, if the left-hand side $L_p$ matches a subnet of $Z$ then this can be reconfigured into $Z'$ by replacing the occurrence of $L_p$ with the right-hand side $R_p$, preserving the subnet $K_p$. Note that by requiring the existence of the correspondence $\eta_p$, we guarantee that the interface of the transformed net, consisting of the open places, is left untouched by the reconfiguration (a more general treatment can be found in [4]). A rewriting rule $p$ is called *behaviour preserving* if its left- and right-hand sides are bisimilar: more precisely, if $L_p \approx_{\eta_p} R_p$.

**Definition 13 (open net transformation).** *Let $p$ be a rewriting rule over open nets, let $Z$ be an open net and let $m : L_p \to Z$ be a match, i.e., an open net embedding. We say that $Z$ rewrites to $Z'$ using $p$ at match $m$, writing $Z \Rightarrow^{p,m} Z'$ or simply $Z \Rightarrow^{p} Z'$, if the diagram of Fig. 4(a) can be constructed in* **ONet***, where both squares are pushouts, and morphism $n$ is composable with both $l_p$ and $r_p$.*

We stress that we are interested in transformations where the two pushout squares are built from composable arrows (technically, this ensures that the transformation can be performed in **Net** and then "lifted" to **ONet**).

The next result is now an easy consequence of Theorem 1.

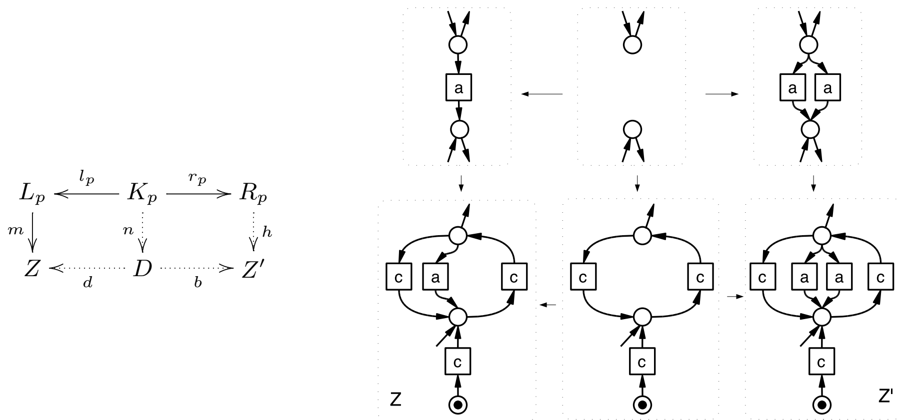
**Theorem 2 (behaviour-preserving reconfigurations).** *Let $p$ be a behaviour-preserving open net rule. Given an open net $Z$ and a match $m : L_p \to Z$, if $Z \Rightarrow^{p,m} Z'$ then $Z \approx Z'$.*

For instance, consider the double-pushout diagram in Fig. 4(b). It can be easily seen that the left- and right-hand sides of the applied rule are strongly bisimilar. Hence we can conclude that also $Z$ and $Z'$ are strongly bisimilar.

### 4.1 Applying rules to open nets

As it is common in the categorical approaches to (graph) rewriting, the notion of open net transformation proposed in Definition 13 is rather "declarative" in style, because it requires the existence of two pushouts in category **ONet**, without stating how they can be constructed, and under which conditions. A more explicit description of the conditions under which a rule can be applied to an open net and of the way the resulting net can be constructed, is clearly necessary for practical purposes. Looking at Fig. 4(a), given a rule $p$ and a match $m : L_p \to Z$, in order to build the open net transformation:

- The *pushout complement* of $l_p$ and $m$ must exist. The resulting arrows $n$ and $d$ must be such that $l_p$ and $n$ are composable. Additionally, there can be several pushout complements and in this case a canonical choice should be considered.




(a) (b)

**Fig. 4.** Transforming open nets through DPO rewriting.

– The resulting arrow $n$ must be composable with $r_p$: then we know how to build $Z'$ by Proposition 2.

Unfortunately, although a general theory of DPO rewriting has been developed recently in the framework of adhesive categories [11], we cannot exploit it here since the category of open nets falls outside the scope of the theory. Sufficient hypotheses under which the above conditions are satisfied are made explicit in the following lemma (more general conditions are considered in [4]).

**Lemma 4 (existence of transformations in ONet).** *Let $p$ be an open net rewriting rule, let $Z$ be an open net and let $m : L_p \to Z$ be a match such that:*

1. *for all $s \in L_p - l_p(K_p)$ we have ${}^\bullet m(s) \cup m(s)^\bullet \subseteq m(L_p - K_p)$;*
2. *for all $s \in K_p$, if $s \in \mathsf{in}(r_p) - \mathsf{in}(l_p)$ then $m(l_p(s)) \in O_Z^+$;*
3. *for all $s \in K_p$, if $s \in \mathsf{in}(l_p)$ then $l_p(s) \in O_L^+$ implies $m(l_p(s)) \in O_Z^+$;*

*and the dual of the last two conditions, obtained by replacing $\mathsf{in}()$ by $\mathsf{out}()$ and $+$ by $-$, hold. Then, there exists a transformation $Z \Rightarrow^{p,m} Z'$.*

The intuition underlying the conditions above is the following. Condition 1 is a typical *dangling condition*: it asserts that a place can be deleted only if all the transitions connected to this place are removed as well, otherwise the flow arcs of this transition would remain dangling. Technically, this condition ensures that the pushout complement exists and is unique in the underlying category **Net**. By condition 2, if $s \in \mathsf{in}(r_p) - \mathsf{in}(l_p)$, i.e., the rule $p$ creates a new (ingoing) transition connected to place $s$, without replacing any old one, then the image of $s$ in $Z$ must be (input) open. Finally, condition 3 says that if $s \in \mathsf{in}(l_p)$, i.e., if some (ingoing) transitions are deleted from $s$ then the image of $s$ in $Z$ must be (input) open if so is its image in $L_p$.

Technically, conditions 2 and 3 (and their dual) ensure the existence of a *minimal* pushout complement $D$, i.e., a pushout complement which embeds into any other, which is the one that we choose to define the transformation; the conditions also guarantee the composability of $n$ with both $l_p$ and $r_p$. The net underlying the minimal pushout complement is $D = Z - m(L_p - l_p(K_p))$ (with set difference componentwise on places and transitions), and the open places of $D$ are given by $O_D^x = d^{-1}(O_Z^x)$ for $x \in \{+, -\}$. The initial marking $\hat{u}_D$ is defined as $\hat{u}_D(s) = \hat{u}_Z(d(s))$ for any place $s \in S_D$.

As an example, consider again the DPO diagram in Fig. 4(b). It is not difficult to see that the rule and the match satisfy the conditions of Lemma 4. Hence we can complete the double-pushout construction transforming $Z$ into $Z'$, as depicted in the same figure.

### 4.2 Modeling dynamic reconfigurations of services

Open nets allow us to specify a system as built out of smaller components. Then, its behaviour is captured by the firing behaviour of the open net. However, for highly dynamic systems, as mentioned in the introduction, it can be useful to have the possibility of specifying that, under suitable conditions, some structural changes or reconfigurations of the system can take place. For instance the invocation of a service could trigger a rule which provides an implementation of the required service.

The theory of open net reconfigurations can do the job. As an example, consider net $N_0$ in Fig. 6 which models the view of a traveller on the journey planning and ticket purchase services offered through a travel agency portal.

We distinguish *abstract transitions* representing services that should be provided elsewhere and *concrete transitions* representing local services and control flow actions. The invocation of an external service can be seen at different levels of abstraction. From the point of view of the client process it is just the firing an abstract transition. At a lower level of abstraction, it is captured by a rule such as the one at the top of Fig. 5. An application of this rule, replacing the abstract transition by a new open net, models the discovery and binding of the concrete services required. The left- and right-hand sides of the rule are weakly bisimilar if we observe only the interactions at the open (interface) places, i.e., if we take $\Lambda_\tau = \Lambda$. This can be seen as a proof of the fact that the bound service meets the requirements: both in the abstract transition and in its concrete counterpart any inquiry will produce a corresponding itinerary.

The rule in the bottom of Fig. 5 represents a case where a simple pattern is replaced by a richer one. On the left we say that, given an itinerary, we can either purchase the required tickets or cancel the processes. On the right the transaction is refined, adding a prior reservation phase, while keeping the option to cancel. As above, the rule has weakly bisimilar left- and right-hand sides, ensuring that the visible effect of the abstract and concrete transitions at the interfaces is the same.

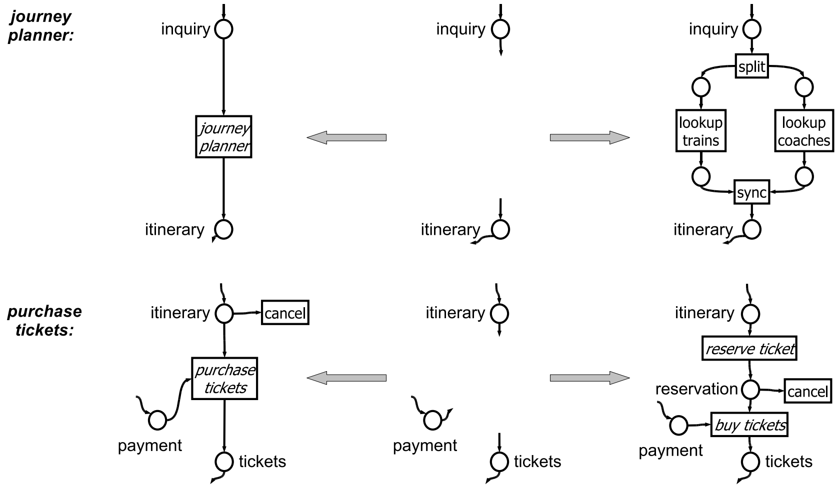



**Fig. 5.** Rules

A possible sequence of transformations is shown in Fig. 6. By the above considerations, we are sure that the transformations do not change the observable behaviour of the system, a fact that can be interpreted as a proof of conformance of the provided service with respect to the abstract specification.

## 5 Conclusion and Related Work

Open nets, introduced in [2, 3], are a reactive extension of standard Petri nets which allows to model systems interacting with an unspecified environment. Several other approaches to Petri net composition and reactivity have been proposed in the literature (see, e.g., [6, 17, 10], to mention a few) and a detailed comparison with the open net model can be found in [3].

In this paper, firstly we have generalised the theory of open nets, including the characterisation of net composition using pushouts, to the case of marked nets. Next we have introduced the notions of strong and weak bisimilarity over open nets. Weak bisimilarity (and, as a particular case, also strong bisimilarity) is shown to be a congruence with respect to the colimit-based composition operation over open nets. To the best of our knowledge, this is the first time that a compositionality result is given for weak bisimilarity over Petri nets. Weak bisimilarity for Petri nets with a composition operation is studied for instance in [17], but it is not congruence, though a context closure allows one to get a congruence which is then characterised by means of a universal context. Our result about strong bisimilarity can be seen as a generalisation of those in [15, 20], which essentially are developed for a special kind of open nets, arising by viewing them as bigraphical reactive systems or as reactive systems over a cospan category. In the resulting reactive Petri net model there is no distinction between

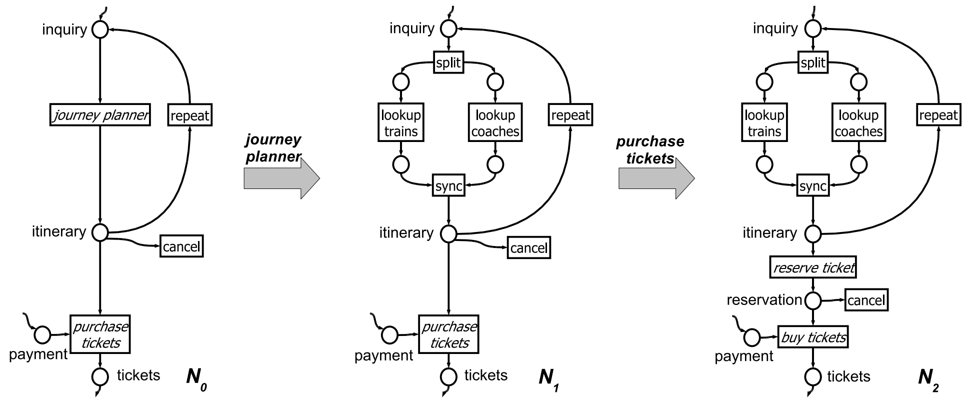



**Fig. 6.** Transformation of open nets representing a travel agent's portal.

open input and output places. Furthermore the composition operation used in these papers does not allow synchronisation of transitions. Similarities exist also with the problem studied in [7], where a reactive Petri net model which admits a compositional behavioural equivalence is exploited, in the framework of web-services, to provide a theoretical basis to service composition and discovery.

In the second part of the paper we have proposed a rewriting-based framework for Petri nets with reconfigurations. We have shown how our congruence results for the observational semantics can be used to identify classes of reconfigurations which do not alter the observational behaviour of the system. This is applied to a small case study of a workflow-like model of a travel agency.

The idea of using rewriting techniques for providing a reconfiguration mechanism for Petri nets has been already explored in the literature (see, e.g., reconfigurable nets of [1, 13] and high-level replacement systems applied to Petri nets in [18]). In future work, besides analysing the relationships between these approaches and ours, we will continue to study the notion of reconfigurable open nets and describe in more detail how reconfigurations can be triggered by the net itself, for example by reaching certain markings or by firing certain transitions, following an intuition similar to that of dynamic nets [8].

Finally, it would be worth studying whether a formal duality can be established between our morphisms and standard simulation morphisms for Petri nets. Viewing our morphisms as inverses of (partial) simulation morphisms would allow to get a precise correspondence between our pushout-based composition and pullback-based synchronisation of Petri nets. Surely by simply taking Winskel's morphisms [22] this does not work (technically because when they are undefined on a transition they must be undefined on the corresponding pre- and post-set). However a duality result could be possibly obtained by considering suitable extensions of Winkel's morphisms, like those in [21, 5].

## References


1. E. Badouel, M. Llorens, and J. Oliver. Modeling concurrent systems: Reconfigurable nets. *Proc. of PDPTA'03*, vol. 4, pp. 1568–1574. CSREA Press, 2003.
2. P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional modeling of reactive systems using open nets. *Proc. of CONCUR'01*, *LNCS* 2154, pp. 502–518. Springer, 2001.
3. 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, 2004.
4. P. Baldan, A. Corradini, H. Ehrig, R. Heckel, and B. König. Bisimilarity and behaviour-preserving reconfigurations of open Petri nets. Technical Report CS-2006-9, Computer Science Department, University Ca' Foscari of Venice, 2006.
5. A.M. Bednarczyk and M.A. Borzyszkowski. General morphisms of Petri nets. *Proc. of ICALP'99*, *LNCS* 1644, pp. 190–199. Springer, 1999.
6. E. Best, R. Devillers, and J. G. Hall. The Petri box calculus: a new causal algebra with multi-label communication. *Advances in Petri Nets*, *LNCS* 609, pp. 21–69. Springer, 1992.
7. F. Bonchi, A. Brogi, S. Corfini, and F. Gadducci. A behavioural congruence for web services. *Proc. of FSEN'07*, to appear in *LNCS*, Springer, 2007
8. M.G. Buscemi and V. Sassone. High-level Petri nets as type theories in the join calculus. In *Proc. of FoSSaCS'01*, *LNCS* 2030. Springer, 2001.
9. H. Ehrig. Tutorial introduction to the algebraic approach of graph-grammars. *Proc. of the 3rd International Workshop on Graph-Grammars and Their Application to Computer Science*, *LNCS* 291, pp. 3–14. Springer, 1987.
10. E. Kindler. A compositional partial order semantics for Petri net components. *Application and Theory of Petri Nets*, *LNCS* 1248, pp. 235–252. Springer, 1997.
11. S. Lack and P. Sobociński. Adhesive and quasiadhesive categories. *RAIRO – Theoretical Informatics and Applications*, 39(3), 2005.
12. J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. *Proc. of CONCUR'00*, *LNCS* 1877, pp. 243–258. Springer, 2000.
13. M. Llorens and J. Oliver. Introducing structural dynamic changes in Petri nets: Marked-controlled reconfigurable nets. *Proc. of ATVA'04*, *LNCS* 3299, pp. 310–323. Springer, 2004.
14. R. Milner. Bigraphical reactive systems. *Proc. of CONCUR'01*, *LNCS* 2154, pp. 16–35. Springer, 2001.
15. R. Milner. Bigraphs for Petri nets. *Lectures on Concurrency and Petri Nets*, *LNCS* 3098, pp. 686–701. Springer, 2003.
16. R. Milner. *A Calculus of Communicating Systems*. *LNCS* 92, Springer, 1980.
17. M. Nielsen, L. Priese, and V. Sassone. Characterizing Behavioural Congruences for Petri Nets. In *Proc. of CONCUR'95*, *LNCS* 962, pp. 175–189. Springer, 1995.
18. J. Padberg, H. Ehrig, and L. Ribeiro. High level replacement systems applied to algebraic high level net transformation systems. *Mathematical Structures in Computer Science*, 5(2):217–256, 1995.
19. W. Reisig. *Petri Nets: An Introduction*. EATCS Monographs. Springer, 1985.
20. V. Sassone and P. Sobociński. A congruence for Petri nets. *Proc. of PNGT'04*, Electronic Notes in Computer Science 127(2), pp. 107–120. Elsevier, 2005.
21. W. Vogler. Executions: A new partial-order semantics of Petri nets. *Theoretical Computer Science*, 91(2):205–238, 1991.
22. G. Winskel. Event structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, *LNCS* 255 pp. 325–392. Springer, 1987.