Composition of Elementary Net Systems based on α-morphisms

Luca Bernardinello, Elisabetta Mangioni, and Lucia Pomello

Dipartimento di Informatica, Sistemistica e Comunicazione,
DISCo - Università degli Studi di Milano-Bicocca
viale Sarca, 336, I-20126 Milano, Italia
mangioni@disco.unimib.it

Abstract. In the development of distributed systems a central role is played by formal tools supporting various aspects of modularity such as compositionality, refinement and abstraction. One of the main challenges consists in developing methods allowing to derive properties of the composed system from properties of the components. In this context we consider Elementary Net Systems related by morphisms and compose them through an interface. Imposing structural constraints on the components, we obtain some structural properties of the composed system and, requiring additional local behavioural constaints, behavioural properties.

Keywords: Elementary Net Systems, morphisms, composition

1 Introduction

In the development of distributed systems a central role is played by formal tools supporting various aspects of modularity such as compositionality, refinement and abstraction. Several formal approaches are available. One of the main challenges consists in developing languages and methods allowing to derive properties of the refined or composed system from properties of the components.

In this paper we present a composition operator such that, by imposing on the components structural constraints and only local behavioural constraints, the composed system inherits behavioural properties of the components.

We consider systems modelled by State Machine Decomposable Elementary Net Systems, i.e.: Elementary Net Systems obtained by composing state machines through synchronized events.

Following the approach proposed in [13, 1, 4, 5], the basic idea consists in composing two different refinements of a common abstract view, obtaining a new model which describes the system comprising the details of both operands, while respecting the same abstract view.

The rules for identifying elements of the nets being composed are expressed by means of morphisms towards another net system, called interface. The interface can be seen as an abstraction of the whole system, shared by the components or, alternatively, it can be interpreted as the specification of the communication protocol with which the components agree. In this case, each operand can be seen as made of the actual, local, component, and of an interface to the rest of the system. Even if this operation is not a limit in the category of nets here considered, the composed system results to be related to both the components and the interface by means of morphisms, and the resulting diagram is commutative.

The use of products in a suitable category of nets as a way to model composition by synchronization has been studied by several authors. A variation on this theme, more similar to ours, proposed by Fabre in [8], applies to safe nets and is built on the notion of pullback.

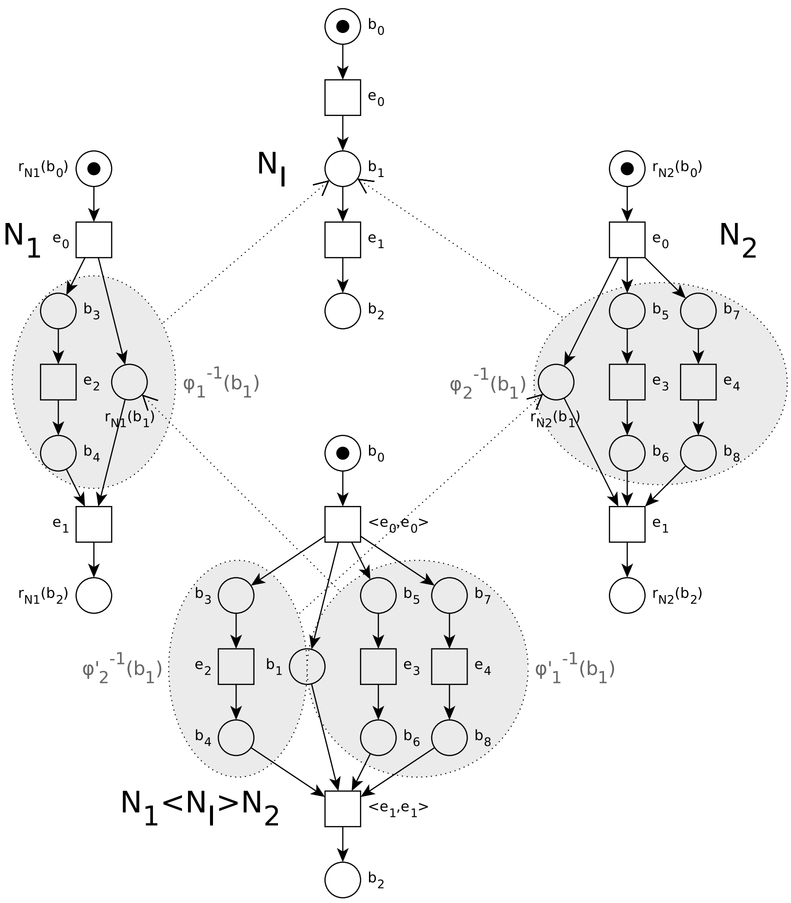


Fig. 1: An example of composition based on α-morphisms

Using morphisms to formalize the relation between a refined net and a more abstract one is not new. The majority of refinement approaches introduced in Petri net theory are mainly based on transition refinement and, less frequently, on place refinement; see [9] and for a survey [6]. Another survey paper, [12], describes a set of techniques which allow to refine transitions in Place/transition nets, so that the relation between the abstract net and its refinement is given by a morphism. There, the emphasis is on refinement rules that preserve specific behavioural properties, within the wider context of general transformation rules on nets.

A very general class of morphisms, interpreted as abstraction of system requirements, with less focus on strict preservation of behavioural properties, is defined in [7].

The refinement we use in this paper is similar in spirit to the one proposed in [11]. In that approach, refinement is defined on transition systems, however it is strictly related to refinement of local states in nets, through the notion of region.

The morphisms used in this paper, called α-morphisms, can be seen as a special case of those introduced by Winskel in [17]. Other morphisms in the same line of Winskel morphisms, are the ones given in [16] and [2].

A simple example shows the main features of our proposal (see Fig. 1). The interface, N_I, is a simple sequence of two events. The two components, N_1 and N_2, refine the same local state, b_1, each by a subnet, shown on a gray background. The composed net, $N_1\langle N_I\rangle N_2$, contains both refinements of b_1, while the rest of the net, not refined by the components, is taken as it is.

The paper is structured as follows. In Section 2 we collect preliminary definitions related to Petri nets which are used in the rest of the paper. Section 3 contains the definition of α-morphisms and their properties. Section 4 contains the definition of $\widehat{N}$-morphisms [13] and their properties. Section 5 defines the composition guided by α-morphisms and the main result of the paper: under some structural and local behavioural properties the composed net is bisimilar to its components. Finally, in Section 6 we discuss some critical issues in our approach. Proofs omitted in this paper can be found in an extended version [3].

2 Preliminary definitions

In this section, we recall the basic definitions of net theory, in particular Elementary Net Systems [15].

We will use the symbol $\downarrow$ to denote the restriction of a function on a subset of its domain.

2.1 Petri Nets

In net theory, models of distributed systems are based on objects called nets which specify local states, local transitions and the relations among them. A *net* is a triple $N = (B, E, F)$, where B is a set of *conditions* or local states, E is a set of *events* or transitions such that $B \cap E = \emptyset$ and $F \subseteq (B \times E) \cup (E \times B)$ is the *flow relation*.

We adopt the usual graphical notation: conditions are represented by circles, events by boxes and the flow relation by arcs. The set of elements of a net will be denoted by $X = B \cup E$; note that we allow nets with isolated elements.

The *preset* of an element $x \in X$ is ${}^\bullet x = \{y \in X | (y, x) \in F\}$; the *postset* of x is $x^\bullet = \{y \in X | (x, y) \in F\}$; the *neighbourhood* of x is given by ${}^\bullet x^\bullet = {}^\bullet x \cup x^\bullet$. These notations are extended to subsets of elements in the usual way.

For any net N we denote the *in-elements* of N by ${}^\bigcirc N = \{x \in X_N : {}^\bullet x = \emptyset\}$ and the *out-elements* of N by $N^\bigcirc = \{x \in X_N : x^\bullet = \emptyset\}$.

A net is *simple* if for all $x, y \in X$, if ${}^\bullet x = {}^\bullet y$ and $x^\bullet = y^\bullet$, then $x = y$.

A net $N' = (B', E', F')$ is a *subnet* of $N = (B, E, F)$ if $B' \subseteq B, E' \subseteq E$, and $F' = F \cap ((B' \times E') \cup (E' \times B'))$. Given a subset of elements $A \subseteq X$, we say that $N(A)$ is the *subnet of* N *identified* by A if $N(A) = (B \cap A, E \cap A, F \cap (A \times A))$.

A *State Machine* is a connected net such that each event e has exactly one input condition and exactly one output condition: $\forall e \in E, |{}^\bullet e| = |e^\bullet| = 1$.

Elementary Net (EN) Systems are a basic system model in net theory. An *Elementary Net System* is a quadruple $N = (B, E, F, m_0)$, where (B, E, F) is a net such that B and E are finite sets, self-loops are not allowed, isolated elements are not allowed, and the *initial marking* is $m_0 \subseteq B$.

The elements in the initial marking are interpreted as the conditions which are true in the initial state.

A subnet of an Elementary Net System N identified by a subset of conditions A and all its pre and post events, $N(A \cup {}^\bullet A^\bullet)$, is a *Sequential Component* of N if $N(A \cup {}^\bullet A^\bullet)$ is a State Machine and if it has only one token in the initial marking.

An Elementary Net System is *covered* by Sequential Components if every condition of the net belongs to at least a Sequential Component. In this case we say that the system is *State Machine Decomposable*.

The behaviour of Elementary Net Systems is defined through the firing rule, which specifies when an event can occur, and how event occurrences modify the holding of conditions, i.e. the state of the system.

Let $N = (B, E, F, m_0)$ be an Elementary Net System, $e \in E$ and $m \subseteq B$. The event e is *enabled* at m, denoted $m\,[e\rangle$, if ${}^\bullet e \subseteq m$ and $e^\bullet \cap m = \emptyset$; the occurrence of e at m leads from m to m', denoted $m\,[e\rangle\, m'$, iff $m' = (m \setminus {}^\bullet e) \cup e^\bullet$.

Let ϵ denote the empty word in E^*. The firing rule is extended to sequences of events by $m\,[\epsilon\rangle\, m$ and $\forall e \in E, \forall w \in E^*, m\,[ew\rangle\, m' = m\,[e\rangle\, m''[w\rangle m'$; w is then called *firing sequence*.

A subset $m \subseteq B$ is a *reachable marking* of N if there exists a $w \in E^*$ such that $m_0\,[w\rangle\, m$. The *set of all reachable markings* of N is denoted by $[m_0\rangle$.

An Elementary Net System is *contact-free* if $\forall e \in E, \forall m \in [m_0\rangle$: ${}^\bullet e \subseteq m$ implies $e^\bullet \cap m = \emptyset$. If an Elementary Net System is covered by Sequential Components then it is contact-free. An event is called *dead* at a marking m if it is not enabled at any marking reachable from m. A reachable marking m is called *dead* if no event is enabled at m. An Elementary Net System is *deadlock-free* if no reachable marking is dead.

2.2 Unfoldings

The semantics of an Elementary Net System can be given as its *unfolding*. The unfolding is an acyclic net, possibly infinite, which records the occurrences of its elements in all possible executions.

Definition 1. *Let $N = (B, E, F)$ be a net, and let $x, y \in X$. We say that x and y are in* conflict*, denoted by $x \#_N y$, if there exist two distinct events $e_x, e_y \in E$ such that $e_x F^* x$, $e_y F^* y$, and ${}^\bullet e_x \cap {}^\bullet e_y \neq \emptyset$.*

Definition 2. *An occurrence net is a net $N = (B, E, F)$ satisfying:*

1. *if $e_1, e_2 \in E, {e_1}^\bullet \cap {e_2}^\bullet \neq \emptyset$ then $e_1 = e_2$;*
2. *F^* is a partial order,*
3. *for any $x \in X, \{y : yF^*x\}$ is finite;*
4. *$\#_N$ is irreflexive,*
5. *the minimal elements with respect to F^* are conditions.*

A branching process of N is an occurrence net whose elements can be mapped to the elements of N.

Definition 3. *Let $N = (B, E, F, m_0)$ be an Elementary Net System, and $\Sigma = (P, T, G)$ be an occurrence net. Let $\pi : P \cup T \to B \cup E$ be a map.*
The pair (Σ, π) is a branching process of N if:

- *$\pi(P) \subseteq B$, $\pi(T) \subseteq E$;*
- *π restricted to the minimal elements of Σ is a bijection on m_0;*
- *for each $t \in T$, π restricted to ${}^\bullet t$ is injective and π restricted to $t^\bullet$ is injective;*
- *for each $t \in T$, $\pi({}^\bullet t) = {}^\bullet(\pi(t))$ and $\pi(t^\bullet) = ({}^\bullet \pi(t))$.*

The unfolding of an Elementary Net System N, denoted by $Unf(N)$, is the "maximal" branching process of N, namely the unique branching process such that any other branching process of N is isomorphic to a subnet of $Unf(N)$. The map associated to the unfolding will be denoted u and called *folding*.

3 A class of morphisms

In the rest of the paper, we consider the class of State Machine Decomposable Elementary Net Systems (SMD-EN Systems).

In this section we give the formal definition of α-morphisms for this class of systems, and present some of their properties, particularly with respect to the preservation of both structural and behavioural properties, as formally introduced in [4].

We start by giving the formal definition of a general morphism and then present the more specific restrictions.

Definition 4. *Let $N_i = (B_i, E_i, F_i, m_0^i)$ be a SMD-EN System, for $i = 1, 2$. An ω-morphism from N_1 to N_2 is a total surjective map $\varphi : X_1 \to X_2$ such that:*

1. $\varphi(B_1) = B_2$;
2. $\varphi(m_0^1) = m_0^2$;
3. $\forall e_1 \in E_1$, *if* $\varphi(e_1) \in E_2$, *then* $\varphi({}^\bullet e_1) = {}^\bullet\varphi(e_1)$ *and* $\varphi(e_1{}^\bullet) = \varphi(e_1)^\bullet$;
4. $\forall e_1 \in E_1$, *if* $\varphi(e_1) \in B_2$, *then* $\varphi({}^\bullet e_1{}^\bullet) = \{\varphi(e_1)\}$;

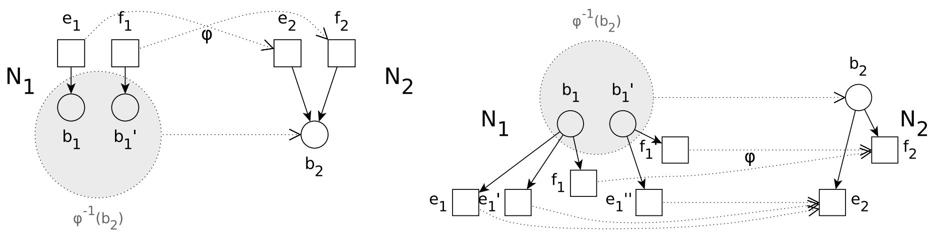
We require that the map is total and surjective because N_1 refines the abstract model N_2, and any abstract element must be related to its refinement.

In particular, a subset of nodes can be mapped on a single condition $b_2 \in B_2$, in this case, we will call *bubble* the subnet identified by this subset $N_1(\varphi^{-1}(b_2))$; if more than one element is mapped on b_2, we will say that b_2 is *refined* by φ. As example, we can see in Fig. 1 the refinement of condition b_1 of N_I with the bubble enclosed in the shaded oval on N_1.

The additional constraints listed in the next definition will be explained below through simple examples.

Definition 5. *Let* $N_i = (B_i, E_i, F_i, m_0^i)$ *be a SMD-EN System, for* $i = 1, 2$. *An* α*-morphism from* N_1 *to* N_2 *is an* ω*-morphism with the following additional constraints:*

5. $\forall b_2 \in B_2$:
   (a) $N_1(\varphi^{-1}(b_2))$ *is an acyclic net;*
   (b) $\forall b_1 \in {}^{\bigcirc}N_1(\varphi^{-1}(b_2))$, $\varphi({}^\bullet b_1) \subseteq {}^\bullet b_2$ *and* $({}^\bullet b_2 \neq \emptyset \Rightarrow {}^\bullet b_1 \neq \emptyset)$;
   (c) $\forall b_1 \in N_1(\varphi^{-1}(b_2))^{\bigcirc}$, $\varphi(b_1{}^\bullet) = b_2{}^\bullet$;
   (d) $\forall b_1 \in \varphi^{-1}(b_2) \cap B_1$,
   $(b_1 \notin {}^{\bigcirc}N_1(\varphi^{-1}(b_2)) \Rightarrow \varphi({}^\bullet b_1) = \{b_2\})$ *and* $(b_1 \notin N_1(\varphi^{-1}(b_2))^{\bigcirc} \Rightarrow \varphi(b_1{}^\bullet) = \{b_2\})$;
   (e) $\forall b_1 \in \varphi^{-1}(b_2) \cap B_1$, *there is a Sequential Component* N_{SC} *of* N_1 *such that* $b_1 \in B_{SC}$ *and* $\varphi^{-1}({}^\bullet b_2{}^\bullet) \subseteq E_{SC}$.



(a) Pre events of an in-condition (b) Post events of an out-condition

Fig. 2: Pre and post event of a bubble

As we can see in Fig. 2a and 2b, in-conditions and out-conditions have different constraints, 5b and 5c respectively. As required by 5c, we do not allow that choices, which are internal to a bubble, constrain a final marking of that bubble: i.e., each out-condition of the bubble must have the same choices of the condition it refines. Instead, pre-events do not need this strict constraint (5b): hence it is sufficient that pre-events of any in-condition are mapped on a subset of the pre-events of the condition it refines. For example, in this particular case, we know that the choice between e_1 and f_1 of Figure 2a is made before the bubble, and this is implied also by the requirement 5e) on Sequential Components. Moreover, the conditions that are internal to a bubble must have pre-events and post-events which are all mapped to the refined condition b_2, as required by 5d.

By 5e, events in the neighbourhood of a bubble, as well as their images, can not be concurrent. However, within a bubble there can be concurrent events. By the combined effect of 5a-5e, in any execution, when a post-event of a bubble fires, in the next marking no local state within the bubble will be marked.

The α-morphisms are closed by composition, the identity function on X is an α-morphism, and the composition is associative. Hence, the family of SMD-EN Systems together with α-morphisms forms a category.

We now list some properties of α-morphisms which have been proved in [4]. Given an α-morphism $\varphi : N_1 \to N_2$ we can say that:

p1 the partition of the nodes of N_1 induced by φ can be lifted to a net structure: the class of nodes mapped to a place b becomes a place, while the class of nodes mapped to an event e becomes an event; the flow relation is defined in the obvious way. The resulting net is isomorphic to N_2;

p2 firing an output event of a bubble empties the bubble: Let $e_1 \in E_1, b_2 \in B_2$: $e_1 \in \varphi^{-1}(b_2{}^\bullet)$; $m_1, m_1' \in \left[m_0^1\right\rangle$: $m_1 \left[e_1\right\rangle m_1'$, then $m_1' \cap \varphi^{-1}(b_2) = \emptyset$;

p3 no input event of a bubble is enabled whenever a token is within the bubble: Let $e_1 \in E_1, b_2 \in B_2$: $e_1 \in \varphi^{-1}({}^\bullet b_2)$; $m_1, m_1' \in \left[m_0^1\right\rangle$: $m_1 \left[e_1\right\rangle m_1'$ then $m_1 \cap \varphi^{-1}(b_2) = \emptyset$;

p4 sequential components are reflected in the sense that the inverse image of a sequential component is covered by sequential components. Sequential components are not preserved;

p5 φ preserves reachable markings:
If $m_1 \in \left[m_0^1\right\rangle$ and $m_1 \left[e\right\rangle m_1'$ in N_1 then $\varphi(m_1) \in \left[m_0^2\right\rangle$ and
– if $\varphi(e) \in E_2$ then $\varphi(m_1) \left[\varphi(e)\right\rangle \varphi(m_1')$ else
– (if $\varphi(e) \in B_2$ then) $\varphi(m_1) = \varphi(m_1')$.

Stronger properties hold under additional constraints. Given an α-morphism $\varphi : N_1 \to N_2$, and a condition $b_2 \in B_2$ with its refinement $N_1(\varphi^{-1}(b_2))$, we define two new SMD-EN Systems. The first one, denoted $S_1(b_2)$, contains (a copy of) the subnet $N_1(\varphi^{-1}(b_2))$, its pre and post events in E_1 and two new conditions: b_1^{in}, which is pre of all the pre events, and b_1^{out}, which is post of all the post-events. The initial marking of $S_1(b_2)$ will be $\{b_1^{in}\}$ or, if there are no pre events, the initial marking of the bubble in N_1. The second system, denoted $S_2(b_2)$, contains b_2, its pre- and post-events and two new conditions: b_2^{in}, which is pre of all the pre-events, and b_2^{out}, which is post of all the post-events. The initial marking of $S_2(b_2)$ will be $\{b_2^{in}\}$ or, if there are no pre events, the initial marking of b_2. Define φ^S as a map from $S_1(b_2)$ to $S_2(b_2)$, which restricts φ to the elements of $S_1(b_2)$, and extends it with $\varphi^S(b_1^{in}) = b_2^{in}$ and $\varphi^S(b_1^{out}) = b_2^{out}$. Note that $S_1(b_2)$ and

$S_2(b_2)$ are SMD-EN Systems and that φ^S is an α-morphism. Let $Unf(S_1(b_2))$ be the unfolding of $S_1(b_2)$, with folding function $u : Unf(S_1(b_2)) \to S_1(b_2)$.

Consider the following additional constraints:

c1 the initial marking of each bubble is at the start of the bubble itself; formally: for each $b_2 \in B_2$ one of the following conditions hold
- $\varphi^{-1}(b_2) \cap m_0^1 = \emptyset$ or
- if ${}^\bullet b_2 \neq \emptyset$ then there is $e_1 \in \varphi^{-1}({}^\bullet b_2)$ such that $\varphi^{-1}(b_2) \cap m_0^1 = e_1{}^\bullet$ or
- if ${}^\bullet b_2 = \emptyset$ then $\varphi^{-1}(b_2) \cap m_0^1 = {}^\circ\varphi^{-1}(b_2)$;

c2 any condition is refined by a subnet such that, when a final marking is reached, this one enables events which correspond to the post-events of the refined condition, i.e.:
$\varphi^S \circ u$ is an α-morphism from $Unf(S_1(b_2))$ to $S_2(b_2)$;

c3 different bubbles do not "interfere" with each other:
we say that two bubbles interfere with each other when their images share, at least, a neighbour.

The first condition assures that the initial marking of a bubble, if present, is in the initial conditions of the bubble and is generated by one of the pre-events, if there are some of them. The second condition is necessary to give to each final marking of a bubble the same choices that the abstract condition has. The third one is not restrictive since the refinement of two interfering conditions can be done in two different steps.

Under **c1**, **c2**, and **c3**, the following properties can be proved [4]:

p6 reachable markings of N_2 are reflected:
for all $m_2 \in [m_0^2\rangle$, there is $m_1 \in [m_0^1\rangle$ such that $\varphi(m_1) = m_2$;

p7 N_1 and N_2 are weakly bisimilar:
by using φ, define two labelling functions such that E_2 are all observable, i.e.: l_2 is the identity function, and the invisible events of N_1 are the ones mapped to conditions; then (N_1, l_1) and (N_2, l_2) are weakly bisimilar $(N_1, l_1) \approx (N_2, l_2)$.

For a definition of weak bisimulation of EN Systems see [14].

4 Relations with $\widehat{N}$-morphisms

The ω and α-morphisms here defined are related to $\widehat{N}$-morphisms, introduced in [13] and studied in [5], that are a restriction of N-morphisms defined in [10].

Here, we are interested in pointing out the precise relation, because we will apply to α-morphisms some results previously shown for $\widehat{N}$-morphisms.

First, let us recall the definition of $\widehat{N}$-morphisms.

Definition 6. *Let $N_i = (B_i, E_i, F_i, m_0)$ be an EN system for $i = 1, 2$.*
A $\widehat{N}$-morphism from N_1 to N_2 is a pair (β, η), where:

1. *$\beta \subseteq B_1 \times B_2$ and $\beta^{-1} : B_2 \to B_1$ is a total and injective function;*

2. $\eta : E_1 \to_* E_2$ *is a partial and surjective function;*
3. if $\eta(e_1)$ *is undefined, then* $\beta({}^\bullet e_1) = \emptyset = \beta(e_1{}^\bullet)$*;*
4. if $\eta(e_1) = e_2$*, then* $\beta({}^\bullet e_1) = {}^\bullet e_2$ *and* $\beta(e_1{}^\bullet) = e_2{}^\bullet$*;*
5. $\forall (b_1, b_2) \in \beta : [b_1 \in m_0^1 \Leftrightarrow b_2 \in m_0^2]$.

In order to compare ω- and α-morphisms with $\widehat{N}$-morphisms, we need some auxiliary notions. Given an ω-morphism φ from N_1 to N_2, we say that N_1 is *canonical* with respect to φ if, for each bubble induced by φ, it contains a local state corresponding to the image of the bubble.

Definition 7. *Let* $\varphi : X_1 \to X_2$ *be an* ω*-morphism from* N_1 *to* N_2*.* N_1 *is* canonical *with respect to* φ *if for each* $b_2 \in B_2$*, there exists a unique* $b_1 \in \varphi^{-1}(b_2) \cap B_1$ *satisfying:*

- $b_1 \in m_0^1 \Leftrightarrow b_2 \in m_0^2$*;*
- ${}^\bullet b_1 = \varphi^{-1}({}^\bullet b_2)$*;*
- $b_1{}^\bullet = \varphi^{-1}(b_2{}^\bullet)$*.*

In this case, b_1 *is said to be the* representation *of* b_2*, denoted* $r_{N_1}(b_2)$*. We define the subnet of a bubble, obtained by removing the representation:* $N_1^{-rep}(b_2) = N_1(\varphi^{-1}(b_2) \setminus \{r_{N_1}(b_2)\})$*.*

If N_1 is not canonical, it is always possible to construct its unique canonical version, $N_1^{\mathcal{C}}$, either by adding the missing representations (and marking them as their images) or by deleting multiple representations. The corresponding morphism, $\varphi^{\mathcal{C}}$, coincides with φ, plus the mapping of new conditions on the corresponding conditions of N_2. It is easy to verify that the canonical version of a system, with respect to an α-morphism to another SMD-EN System, is unique up to isomorphisms.

We have proved in [4] that $\varphi^{\mathcal{C}}$ is an ω-morphism from $N_1^{\mathcal{C}}$ to N_2. Here, we need to prove that, if φ is an α-morphism, then $\varphi^{\mathcal{C}}$ is also an α-morphism, as needed in Section 5.

Proposition 1. *Let* $\varphi : N_1 \to N_2$ *be an* α*-morphism, then* $\varphi^{\mathcal{C}}$ *is an* α*-morphism from* $N_1^{\mathcal{C}}$ *to* N_2*.*

Given an ω-morphism from N_1 to N_2, take $N_1^{\mathcal{C}}$, N_2 and $\varphi^{\mathcal{C}}$. Now, restrict $\varphi^{\mathcal{C}}$ to all the nodes of $N_1^{\mathcal{C}}$ that are not in a bubble $N_1^{-rep}(b_2)$ for some $b_2 \in B_2$ and call it $(\varphi^{\mathcal{C}})^{rep}$: this is a $\widehat{N}$-morphism.

Proposition 2. $((\varphi^{\mathcal{C}})^{rep} \downarrow B_1^{\mathcal{C}}, (\varphi^{\mathcal{C}})^{rep} \downarrow E_1^{\mathcal{C}})$ *is a* $\widehat{N}$*-morphism.*

Every α-morphism is obviously an ω-morphism. Adding the representation for each condition of N_2 does not modify its behaviour, because of the constraint on sequential components. Hence, the results achieved here hold for α-morphisms. In this sense, we consider them as a special case of $\widehat{N}$-morphisms.

The converse is not true, as shown in Fig. 3, where an $\widehat{N}$-morphism from N_1 to N_2 is given by identical names of elements; it is easy to see that there is no α-morphism from N_1 to N_2, since there is no way to map b_3 and b_5.

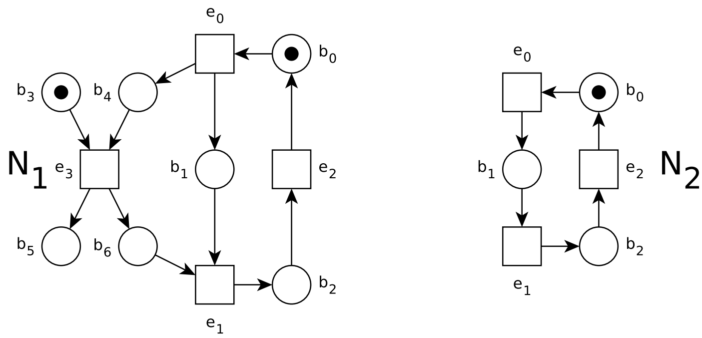


Fig. 3: An example of $\widehat{N}$-morphism which is not an α-morphism

$\widehat{N}$-morphisms are suitable to drive an operation of *composition* of nets. Let N_1 and N_2 be a pair of EN Systems, each one related to another EN System, called interface N_I, by $\widehat{N}$-morphisms, (β_i, η_i). We can see N_I as the protocol of the interaction between them. The morphisms are surjective so that each system cannot ignore a part of the protocol. The composition of N_1 and N_2 on the interface N_I, denoted $N = N_1\langle N_I\rangle N_2$, is given by the union of the local part of each system N_i and the common part corresponding to the protocol. The composition induces $\widehat{N}$-morphisms, (β_i', η_i'), from the composed system to its components.

This composition has several properties, proved in [5], which will be used later and which we informally resume here:

n1 *if the components reflect the sequences of the interface*, the composed net reflects the sequences of the two components;
n2 *if one component is weakly bisimilar to the interface*, then the composed net is weakly bisimilar to the other component.

In particular, **n2** says that if a component is bisimilar to the interface, then only the other component can add behavioural constraints to the composed system.

5 Composition based on α-morphisms

In this section, we define a way of composing SMD-EN systems, in a similar way as in [5], but based on α-morphisms.

The starting point is a set of three SMD-EN systems; one of them, N_I, plays the role of an interface between the other two, N_1 and N_2. A pair of α-morphisms, one from N_1 to N_I, the other from N_2 to N_I, determine how the two components refine the local states of the interface, and which events in the two components have to synchronize.

The crucial point in the definition concerns the choice of synchronizing events. Suppose that the morphisms onto the interface map bubbles A_1 and A_2 to the same local state b (where A_i is taken in N_i). Then, the representations of A_1 and A_2 are local states which are identified in composing the two nets. This implies that any event in N_1 which puts a token in the representation of A_1 must be synchronized with any event doing the same in the representation of A_2. This explains the definition of the sets E_{sync}, below.

It is assumed that N_1, N_2 and N_I are disjoint and that N_1 and N_2 are canonical with respect to the corresponding morphisms.

Definition 8. *Let $N_i = (B_i, E_i, F_i, m_0^i)$ be an SMD-EN System for $i = 1, 2, I$. Let φ_i, with $i = 1, 2$, be an α-morphism from N_i to N_I. Let N_i be canonical with respect to φ_i.*

We define $N = N_1\langle N_I\rangle N_2 = (B, E, F, m_0)$ such that

$$B = \bigcup_{b_I \in B_I} B_{Bubble(b_I)} \qquad E = \left(\bigcup_{e_I \in E_I} E_{sync}(e_I)\right) \cup \left(\bigcup_{b_I \in B_I} E_{Bubble(b_I)}\right)$$

$$F = \bigcup_{b_I \in B_I} \left(F(b_I) \cup F_{Bubble(b_I)}\right)$$

Where:

$$E_{sync}(e_I) = \{e = \langle e_1, e_2\rangle : e_1 \in E_1, e_2 \in E_2, \varphi_1(e_1) = e_I = \varphi_2(e_2)\}$$

Let $b_I \in B_I$:

$$\begin{aligned} Bubble(b_I) = ((&B_{N_1^{-rep}(b_I)} \cup \{b_I\} \cup B_{N_2^{-rep}(b_I)}), \\ &(E_{N_1^{-rep}(b_I)} \cup E_{N_2^{-rep}(b_I)}), \\ &(F_{N_1^{-rep}(b_I)} \cup F_{N_2^{-rep}(b_I)})) \end{aligned}$$

$$F(b_I) = {}^{\bullet}F(b_I) \cup F^{\bullet}(b_I)$$

Let $e = \langle e_1, e_2\rangle \in \bigcup_{e_I \in {}^{\bullet}b_I} E_{sync}(e_I)$,

$$\begin{aligned} {}^{\bullet}F(b_I) = &\{(e, b) : b \in {}^{\bigcirc}Bubble(b_I), (e_1, b) \in F_1\} \cup \\ &\{(e, b_I)\} \cup \\ &\{(e, b) : b \in {}^{\bigcirc}Bubble(b_I), (e_2, b) \in F_2\} \end{aligned}$$

Let $e = \langle e_1, e_2\rangle \in \bigcup_{e_I \in b_I{}^{\bullet}} E_{sync}(e_I)$,

$$\begin{aligned} F^{\bullet}(b_I) = &\{(b, e) : b \in Bubble(b_I)^{\bigcirc}, (b, e_1) \in F_1\} \cup \\ &\{(b_I, e)\} \cup \\ &\{(b, e) : b \in Bubble(b_I)^{\bigcirc}, (b, e_2) \in F_2\} \end{aligned}$$

Note that in order to simplify the notation, $N_1\langle N_I\rangle N_2$ does not refer to the morphisms φ_i. By construction, $N = N_1\langle N_I\rangle N_2$ as defined above is an EN System. Moreover, it is covered by sequential components. To see this, take $b \in B$. If $b \in B_I$, then b belongs to a sequential component in N_I, and all the conditions in this component are also in N, and these, together with their neighbourhood, identify a sequential component in N. If $b \in B_i$, then b belongs to a sequential component in N_i, and all the conditions in this component have a corresponding condition in N. It is easy to check that these, together with their neighbourhood, identify a sequential component in N.

We now define a map φ'_i from N onto N_i, and we will show in Theorem 1 that it is an α-morphism.

Definition 9. *Define φ'_i as follows, for each $x \in X$:*

$$\varphi'_i(x) = \begin{cases} x, & \text{if } x \in X_i \\ r_{N_i}(x), & \text{if } x \in B_I \\ r_{N_i}(\varphi_{3-i}(x)), & \text{if } x \in B_{3-i} \\ e_i, & \text{if } x = \langle e_1, e_2\rangle \\ r_{N_i}(\varphi_{3-i}(x)), & \text{if } x \in E_{3-i} \end{cases}$$

Theorem 1. *The map φ'_i is an α-morphism from $N = N_1\langle N_I\rangle N_2$ to $N_i, i = 1, 2$.*

By construction we get the following result:

Proposition 3. *The system $N = N_1\langle N_I\rangle N_2$ is canonical with respect to φ'_1 and to φ'_2.*

These results say that the composed system refines both the components, as well as the interface. For each abstract condition there is a corresponding condition in the composed system.

$$\begin{tikzcd}
& N_I & \\
N_1 \arrow[ur, "\varphi_1"] & & N_2 \arrow[ul, "\varphi_2"'] \\
& N_1\langle N_I\rangle N_2 \arrow[ul, "\varphi'_1"] \arrow[ur, "\varphi'_2"'] &
\end{tikzcd}$$

To show that the diagram above commutes, we prove that the operation essentially coincides with the composition based on $\widehat{N}$-morphisms. Since in that case the diagram commutes, the same holds for α-morphisms.

The following proposition is the direct consequence of the definitions of composition.

Proposition 4. *Let $N_i = (B_i, E_i, F_i, m_0^i)$ be an SMD-EN System for $i = 1, 2, I$. Let φ_i, with $i = 1, 2$, be an α-morphism from N_i to N_I. Let N_i be canonical with respect to φ_i. Let $N^\alpha = N_1\langle N_I\rangle^\alpha N_2 = (B, E, F, m_0)$ be the composition of N_1 and N_2 using φ_1 and φ_2. Let φ_i' be the α-morphism from N to N_i created by the composition operation.*

Now, consider the $\widehat{N}$-morphism $((\varphi_i)^{rep} \downarrow B_i, (\varphi_i)^{rep} \downarrow E_i)$. Let $N^{\widehat{N}} = N_1\langle N_I\rangle^{\widehat{N}} N_2 = (B, E, F, m_0)$ be the composition of N_1 and N_2 using $((\varphi_1)^{rep} \downarrow B_1, (\varphi_1)^{rep} \downarrow E_1)$ and $((\varphi_2)^{rep} \downarrow B_2, (\varphi_2)^{rep} \downarrow E_2)$. Let (β_i', η_i') be the $\widehat{N}$-morphism from N to N_i created by the composition operation.

The systems N^α and $N^{\widehat{N}}$ are isomorphic, $\beta_i' = (\varphi_i')^{rep} \downarrow B_i$ and $\eta_i' = (\varphi_i)^{rep} \downarrow E_i$.

The diagram in Fig. 1 is an example of composition which is not a pull-back diagram. It is still an open problem whether, in general, the diagram of a composition operation is a pushout.

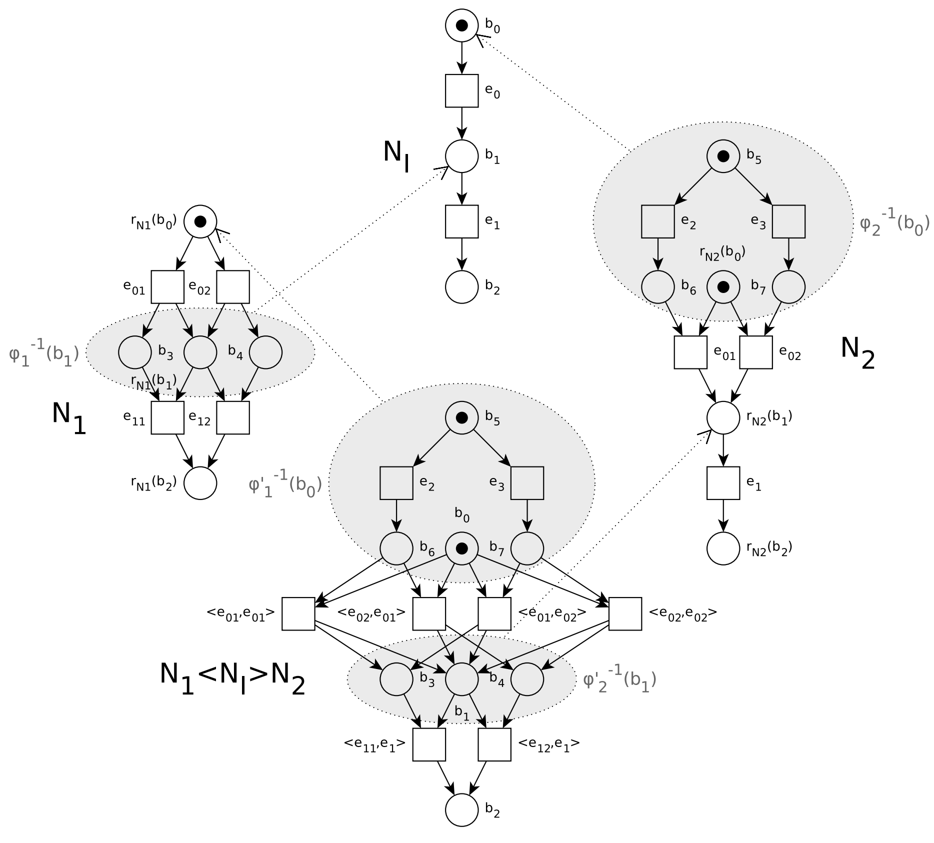


Fig. 4: An example of composition based on α-morphisms

From results in Section 3 and 4 we can derive a property valid for composition based on α-morphisms. We know that, if N_1 is weakly bisimilar to N_I then N is weakly bisimilar to N_2. By **p7** we can check weak bisimilarity between N_1 and N_I using **c1**, **c2** and **c3**. These constraints are either structural or locally behavioural, while, in the case of $\widehat{N}$-morphisms, checking bisimilarity must be made globally. Fig. 4 shows an example in which N_1 and N_2 are weakly bisimilar to N_I. Hence $N_1\langle N_I\rangle N_2$ is weakly bisimilar to N_1, N_2 and N_I.

6 Conclusions

We have proposed a way to compose State Machine Decomposable EN Systems, by identifying elements of the components. The identification is ruled by morphisms from the components to a net, which can be seen as an interface or as a common abstraction of the overall system.

We have proved that α-morphisms can be seen as a particular case of $\widehat{N}$-morphisms [5] and that, composing two systems using α-morphisms or using $\widehat{N}$-morphisms, we obtain isomorphic systems.

Here, we have looked at the properties of the composed net which can be deduced from properties of the components. In particular, the constraints of α-morphisms allow to check bisimilarity between a component and the interface by using only structural and local behavioural constraints. By a property holding also in the case of $\widehat{N}$-morphisms, this can be lifted to bisimilarity between the composed net and the components.

We plan to explore the extension of these ideas to P/T nets and to colored nets that can be unfolded to State Machine Decomposable EN Systems.

Acknowledgments

Work partially supported by MIUR.

References

1. Marek A. Bednarczyk, Luca Bernardinello, Benoît Caillaud, Wiesław Pawłowski, and Lucia Pomello. Modular system development with pullbacks. In Wil M. P. van der Aalst and Eike Best, editors, *ICATPN*, volume 2679 of *Lecture Notes in Computer Science*, pages 140–160. Springer, 2003.
2. Marek A. Bednarczyk and Andrzej M. Borzyszkowski. On concurrent realization of reactive systems and their morphisms. In Hartmut Ehrig, Gabriel Juhás, Julia Padberg, and Grzegorz Rozenberg, editors, *Unifying Petri Nets*, volume 2128 of *Lecture Notes in Computer Science*, pages 346–379. Springer, 2001.
3. Luca Bernardinello, Elisabetta Mangioni, and Lucia Pomello. Composition of elementary net systems based on α-morphisms. Internal report (2012), available at `http://www.mc3.disco.unimib.it/pub/bmp2012-compo.pdf`.
4. Luca Bernardinello, Elisabetta Mangioni, and Lucia Pomello. Local state refinement on elementary net systems: an approach based on morphisms. In Proc. Workshop PNSE 2012, Hamburg 2012.

5. Luca Bernardinello, Elena Monticelli, and Lucia Pomello. On preserving structural and behavioural properties by composing net systems on interfaces. *Fundam. Inform.*, 80(1-3):31–47, 2007.
6. Wilfried Brauer, Robert Gold, and Walter Vogler. A survey of behaviour and equivalence preserving refinements of Petri nets. *Advances in Petri Nets 1990*, pages 1–46, 1991.
7. Jörg Desel and Agathe Merceron. Vicinity respecting homomorphisms for abstracting system requirements. *Transactions on Petri Nets and Other Models of Concurrency*, 4:1–20, 2010.
8. Eric Fabre. On the construction of pullbacks for safe Petri nets. In Susanna Donatelli and P. S. Thiagarajan, editors, *ICATPN*, volume 4024 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2006.
9. Claude Girault and Rüdiger Valk. *Petri nets for systems engineering - a guide to modeling, verification, and applications*. Springer, 2003.
10. Mogens Nielsen, Grzegorz Rozenberg, and P. S. Thiagarajan. Elementary transition systems. *Theor. Comput. Sci.*, 96(1):3–33, 1992.
11. Mogens Nielsen, Grzegorz Rozenberg, and P. S. Thiagarajan. Elementary transition systems and refinement. *Acta Inf.*, 29(6/7):555–578, 1992.
12. Julia Padberg and Milan Urbásek. Rule-based refinement of Petri nets: A survey. In Hartmut Ehrig, Wolfgang Reisig, Grzegorz Rozenberg, and Herbert Weber, editors, *Petri Net Technology for Communication-Based Systems*, volume 2472 of *Lecture Notes in Computer Science*, pages 161–196. Springer, 2003.
13. Lucia Pomello and Luca Bernardinello. Formal tools for modular system development. In Jordi Cortadella and Wolfgang Reisig, editors, *ICATPN*, volume 3099 of *Lecture Notes in Computer Science*, pages 77–96. Springer, 2004.
14. Lucia Pomello, Grzegorz Rozenberg, and Carla Simone. A survey of equivalence notions for net based systems. In Grzegorz Rozenberg, editor, *Advances in Petri Nets: The DEMON Project*, volume 609 of *Lecture Notes in Computer Science*, pages 410–472. Springer, 1992.
15. Grzegorz Rozenberg and Joost Engelfriet. Elementary net systems. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Petri Nets*, volume 1491 of *Lecture Notes in Computer Science*, pages 12–121. Springer, 1996.
16. Walter Vogler. Executions: A new partial-order semantics of Petri nets. *Theor. Comput. Sci.*, 91(2):205–238, 1991.
17. Glynn Winskel. Petri nets, algebras, morphisms, and compositionality. *Inf. Comput.*, 72(3):197–238, 1987.