A Categorical Framework for Concurrent, Anticipatory Systems

P. Blauth Menezes *

Abstract

A categorical semantic domain is constructed for Petri nets which satisfies the diagonal compositionality requirement with respect to anticipations, i.e., Petri nets are equipped with a compositional anticipation mechanism (vertical compositionality) that distributes through net combinators (horizontal compositionality). The anticipation mechanism is based on graph transformations (single pushout approach). A finitely bicomplete category of partial Petri nets and partial morphisms is introduced. Classes of transformations stand for anticipations. The composition of anticipations (i.e., composition of pushouts) is defined, leading to a category of nets and anticipations which is also complete and cocomplete. Since the anticipation operation composes, the vertical compositionality requirement of Petri nets is achieved. Then, it is proven that the anticipation also satisfies the horizontal compositionality requirement. A specification grammar stands for a system specification and the corresponding induced subcategory of nets and anticipation's stands for all possible dynamic anticipation's of the system (objects) and their relationship (morphims).

Keywords: anticipatory systems, concurrency, compositionality, graph transformation, category theory.

1 INTRODUCTION

Petri nets (see, for instance, [19]) are one of the first models for concurrency developed and are widely used in many applications. Also, as stated in [17], Petri nets are able to distinguish clearly the basic concepts in the behavior of processes and the graphical representation visualizes these concepts. However, nets lack the following properties:

*blauth@inf.ufrgs.br, Instituto de Informática, UFRGS, Porto Alegre, RS, Brasil

a) Composition. Lack a way of composing larger nets from smaller ones through high level operators (net combinators). Complex systems are structured entities and can be better understood if we can reason and build on their parts separately.
b) Anticipation. Lack a way of modifying nets state or topology according to possible future causes, i.e., nets in the sense of [19] do not model anticipatory systems in the sense of [21, 5]. In fact, one of the drawbacks of Petri nets it that they have a static topology, i.e. a net may not change dynamically.

Moreover, we aim a mathematical theory for concurrent, anticipatory systems which is compositional with respect to combinators, anticipations or both and thus, should satisfy the *Diagonal Compositionality Requirement* with respect to anticipations which means both:

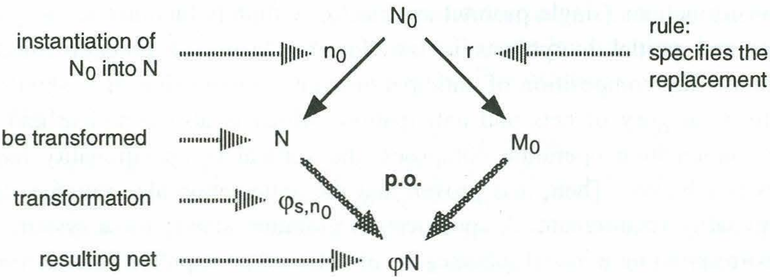

Figure 1 - Single pushout approach to net transformation

a) Vertical Compositionality. Anticipations compose.
b) Horizontal Compositionality. Anticipations distribute through net combinators, i.e., a concurrent system may anticipate before or after the joint behavior of component parts in order to obtain the same resulting system.

Recently, (categorical) frameworks based on Petri nets have been proposed for expressing the semantics of concurrent systems in the so-called true concurrency approach. The framework in [23] provides compositionality of nets where categorical constructions such as product and coproduct stand for system combinators. The approach in [16] provides abstraction mechanisms where a special kind of net morphism corresponds to a notion of implementation. Menezes and Costa [14] introduces a semantic domain on this abstraction mechanisms (which can be viewed as an anticipation). Menezes and Costa [13] provides a framework for synchronization of Petri nets inspired by [22, 16] where the categorical product stands for parallel combinator and a functorial operation defined using the fibration technique stands for synchronization.

The main goal of this paper is to achieve the diagonal compositionality for Petri nets. The anticipation mechanism proposed is based on graph transformation, introduced in [18], using the so-called single pushout approach [8] on a category of nets with partial morphisms. In this context, a graph transformation stands for possible system anticipation. The approach proposed is for net-based systems. However, since it uses categorical constructions, it can be generalized for several other models for concurrency (through adjunctions or property analysis).

The following graph transformation approach for Petri nets was first introduced in [11, 12] but in a different framework (for reifications). First, we introduce the category of partial Petri nets (with initial markings) and partial morphisms that is finitely bicomplete. The category defined is inspired by [16] and we claim that, with respect to partial morphisms, "Petri nets are semi-groups".

In most categorical frameworks for Petri nets, if an initial marking is added to the net structure, the resulting categories do not have colimits. For instance, in [16, 23], the categories are restricted in order the have coproducts. In this paper, we define categories of Petri nets with a set of initial markings (instead of a single initial marking) based on [10, 7]. The categories of Petri nets with partial morphisms and initial markings defined are finitely complete and *cocomplete*. Note that, it is a basic result for this work, since the anticipation mechanism proposed is defined using the pushout construction.

The graph transformation concept is extended for partial Petri nets with initial markings as follows: a rule $r: N_0 \to M_0$ is a partial net morphism specifying how N_0 (left-hand side) is replaced by M_0 (right-hand side) and an instantiation $n_0: N_0 \to N$ is a partial net morphism specifying how N_0 is instantiated ("matched") into N (the net to be transformed). Then, φN is the resulting net of the pushout construction of r along with n_0 and φ_{r,n_0} is the transformation morphism, as illustrated in Figure 1.

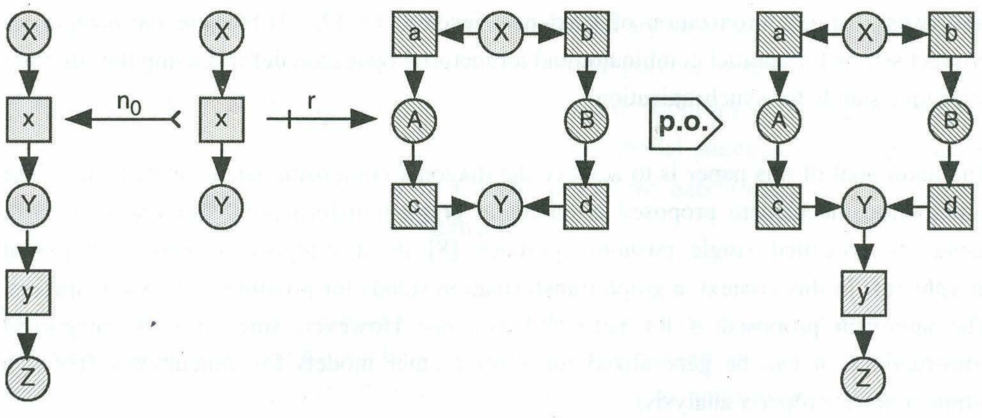

Figure 2 - Replacement of a transition by a net

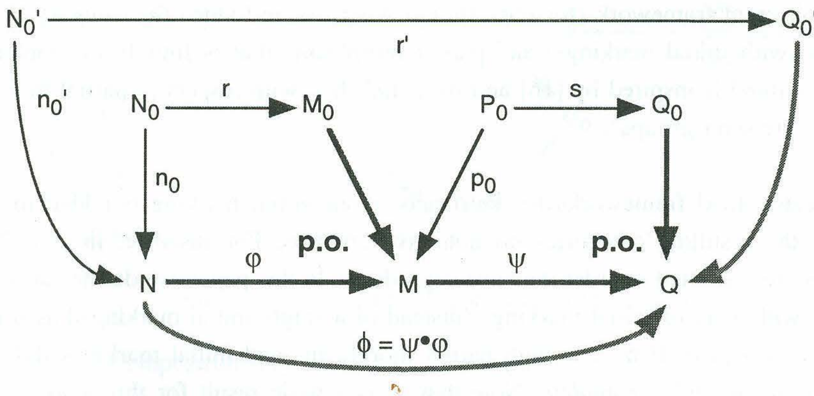

Figure 3 - Composition of pushouts with one vertex in common

For instance, Figure 2 illustrates the replacement of a transition (of a net) by a net preserving its source and target places. The rule r and the instantiation n_0 preserve the places X and Y and so, they are preserved in the resulting net (right most net in the figure). However, while the transition x is "forgotten" by r (partial morphism) it is preserved by n_0. Thus, in the resulting net, x is replaced by a, b, c, d and the nodes A and B are added.

Note that the transformation operation may be used not only to "explode" or "collapse" transitions/places but also to "add" or "delete" parts to an existing net.

To achieve the *Vertical Compositionality*, we have to compose transformations, i.e. for given transformations $\varphi N = M$ and $\psi M = Q$, we need a transformation $\varphi \bullet \psi$ such that $(\varphi \bullet \psi)N = Q$. Note that the matter is not only the composition of φ and ψ as partial net morphisms, but also as pushouts: given two pushouts with only one vertex in common, we have to determine a single pushout such that the resulting transformation is the composition of the component transformations, as illustrated in the Figure 3.

In fact, for some given rule r and instantiation n_0, the resulting transformation φ_{r,n_0} can be determined by several pushouts. Also, several rules and instantiations can give the composition of pushouts. Thus, we define classes of equivalencies of pairs of rules and instantiations such that the resulting transformation coincides. Therefore, a category of nets and partial morphisms leads to a category where objects are nets and morphisms are classes of equivalence called *Anticipations*. Both categories are isomorphic. Then, we show that the *Horizontal Compositionality* for the category of Petri nets and anticipations is achieved.

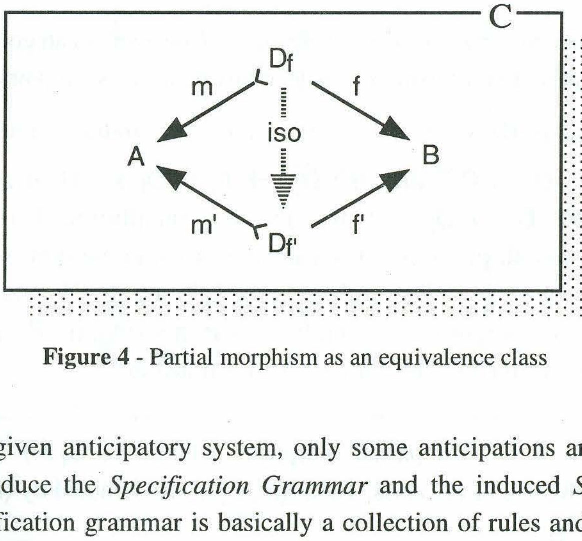

Figure 4 - Partial morphism as an equivalence class

Usually, to specify a given anticipatory system, only some anticipations are desired. For this purpose, we introduce the *Specification Grammar* and the induced *Subcategory of Anticipations*. A specification grammar is basically a collection of rules and instantiations and an initial net. Each specification grammar induces a subcategory of partial Petri nets and anticipations, reflecting all possible nets that can be derived from the initial one and the corresponding reachable markings. Therefore, a grammar can be considered as a specification of a system and the induced subcategory as all possible dynamic anticipations of the system. This means that depending on the specified system (given grammar) its dynamic behavior may anticipate in a deterministic (one future state), nondeterministic (more than one possible future state) or concurrent (multiple future states) ways. The anticipation mechanism proposed is able to deal with internal or predefined external aspects [5]. To deal with external aspects (without restrictions), it is enough to modify the grammar definition allowing rules as inputs.

2 PARTIAL PETRI NETS

First we define partial morphisms on a given category *C* as in [1]. Then, we introduce the concepts of graph as an element of a comma category over the base category *Set*, internal graph which is a graph where the base category is an arbitrary category *C,* and structured graph, which is an extension of the notion of internal graph where arcs and nodes may be objects of different categories, provided that there are functors from these categories to the base category. In this context, the category of Petri nets is defined as the category of partial morphisms on a category of structured graphs.

Definition 1. Category with Partial Morphisms. Consider a category $C = \langle Ob_C, Mor_C, \partial_0, \partial_1, \iota, \bullet \rangle$. A partial morphism on *C* is an equivalence class of pairs of morphisms

$\langle m: D_f \rightarrowtail A, f: D_f \rightarrowtail B \rangle$, where m is mono, with respect to the relation

$\langle m: D_f \rightarrowtail A, f: D_f \to B \rangle$ parc $\langle m': D_{f'} \rightarrowtail A, f': D_{f'} \rightarrowtail B \rangle$ if and only if there is an isomorphism iso: $D_f \to D_{f'}$ such that the diagram illustrated in Figure 4 commutes. Suppose that *C* has all pullbacks. The category of partial morphism on *C* is $pC = \langle Ob_C, {}_pMor_C, p\partial_0, p\partial_1, \iota, p\bullet \rangle$ where $pMor_C$, $p\partial_0$, $p\partial_1$ are partial morphisms on *C* and the composition of two morphisms $f = \langle m_f, f \rangle: A \to B$, $g = \langle m_g, g \rangle: B \to C$ is $g \bullet f = \langle m_g \bullet m_f, g \bullet f \rangle: A \to C$ determined by the pullback as illustrated in 5.

Let $[\langle m, f \rangle]: A \to B$ be a partial morphism where $\langle m: D_f \rightarrowtail A, f: D_f \to B \rangle$ is a representative element of the class. Then $[\langle m, f \rangle]$ is also denoted by $\langle m, f \rangle: A \to B$ or $f: A \leftarrowtail D_f \to B$.

Proposition 2. Consider the category *pC* and the partial morphisms $f: A \leftarrowtail D_f \to B$, $g: B \leftarrowtail D_g \to E$, $u: A \leftarrowtail D_u \to C$, $v: C \leftarrowtail D_v \to E$ such that $g \bullet f = v \bullet u$. Then, there are morphisms $p: D_f \leftarrowtail M \to D_v$, $q: D_u \leftarrowtail M \to D_g$ where the middle object M is unique (up to an isomorphism) and are such that the diagram illustrated in Figure 6 commutes. Moreover, ① and ② are pullback.

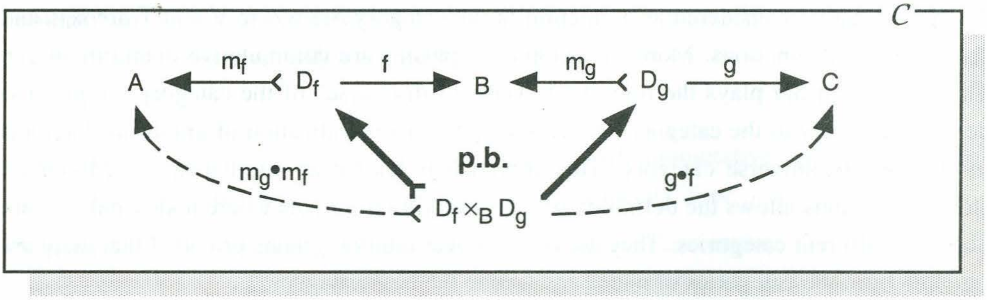

Figure 5 - Composition of partial morphisms (right)

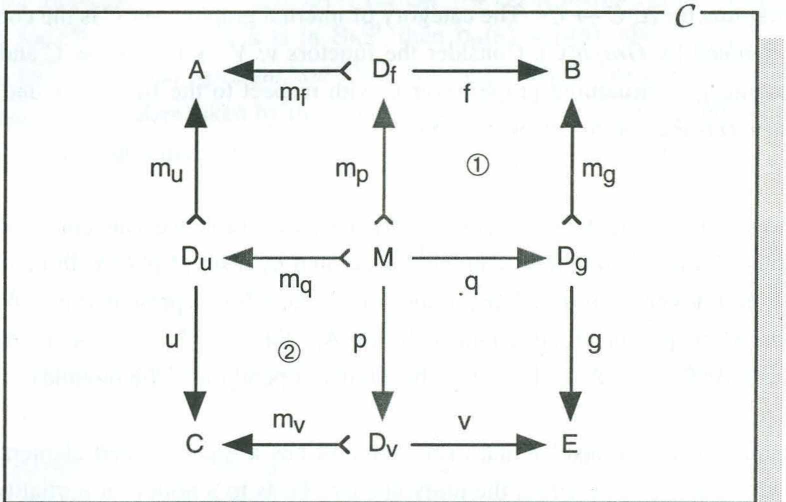

Figure 6 - Commutative square diagram in categories with partial morphisms

Proof: The compositions $g \bullet f$, $v \bullet u$ are given by pullbacks in ①, ② where $D_f \times_B D_g$, $D_u \times_C D_v$ are the pullback objects. Since $g \bullet f = v \bullet u$, there is an isomorphism iso: $D_f \times_B D_g \rightarrow D_u \times_C D_v$ and so, both objects represent the middle object M.

In what follows, the category *Graph* of graphs is defined as the comma category $\Delta\downarrow\Delta$ where Δ: *Set* $\rightarrow$ *Set*^{2} is the diagonal functor as in (0). Thus, a graph is a triple $G = \langle V, T, \partial \rangle$ where $\partial = \langle \partial_0, \partial_1 \rangle$ or just $G = \langle V, T, \partial_0, \partial_1 \rangle$. As stated in [1, 3], a (small) graph $G = \langle V, T, \partial_0, \partial_1 \rangle$ can be considered as a diagram in the category *Set* where V and T are sets and ∂_0, ∂_1 are total functions. Moreover, graph morphisms are commutative diagrams in *Set*. This means that *Set* plays the role of "universe of discourse" of the category *Graph*: it is defined internally to the category *Set*. This suggests a generalization of graphs as diagrams in an arbitrary universe category. This approach is known as internalization. Moreover, structured graphs allows the definition of a special kind of graphs where nodes and arcs are object of different categories. They are defined over internal graphs provided that there are functors from the categories of nodes and arcs to the base category.

Definition 3. Internal Graph, Structured Graph. Consider the (base) category *C* and the diagonal functor Δ: $C \rightarrow C^2$. The category of internal graphs over *C* is the comma category $\Delta\downarrow\Delta$, denoted by *Graph(C)*. Consider the functors v: $V \rightarrow C$, t: $T \rightarrow C$ and Δ: $C \rightarrow C^2$. The category of structured graphs over *C* with respect to the functors *v* and *t* denoted by *Graph(v, t)* is the comma category $\Delta \bullet t \downarrow \Delta \bullet v$.

As proposed in [16], to represent a Petri net as a graph we can consider the states as elements of a free commutative monoid generated by a set of places. In this case, for each transition, n tokens consumed or produced in a place A is represented by nA and n_i tokens consumed or produced simultaneously in A_i, for $i = 1, \ldots, k$ is represented by $n_1A_1 \oplus n_2A_2 \oplus \ldots \oplus n_kA_k$ (where $\oplus$ is the additive operation of the monoid).

Note that, we may consider that every monoid has a distinguished element which is the unity element. In some sense, the unity element leads to a notion of partiality: to forget an element in a monoid homomorphism it is enough to map this element to the unity of the target object. Considering that we need partial morphism in order to define graph transformations, partial monoid homomorphism can be seen as a partial category of a category that already behaves as a partial one. However, if we consider the category of semi-groups with partial morphisms instead of the category of monoids, the notion of Petri nets as graphs as in [16] is kept. Thus, we claim that, for partial morphisms "Petri nets are semi-groups". In what follows, the main reference for concrete categories is [2].

The category of free commutative semi-groups with partial morphisms *pCSem*, is concrete over the category of free commutative monoids *CMon*. In fact, any semi-group can be canonically extended as a monoid and a partial semi-group morphism can be viewed as a "pointed" morphism of monoids, where the distinguished element is the unity. Moreover, the limits and colimits of *CMon* are lifted to *pCSem*.

Definition 4. Category pCSem. Consider the category of commutative semi-groups *CSem*. The category *pCSem* is the category of partial morphisms on *CSem*.

Proposition 5. The category *pCSem* is finitely complete and cocomplete.

Proof: Consider the functor *sm: pCSem* $\rightarrow$ *CMon* such that for all *pCSem*-object $S^{\oplus}$, *sm* $S^{\oplus} = S_e^{\oplus}$, where $S_e^{\oplus}$ is the free monoid generated by the set S with e as the unity element and for all *pCSem*-morphism h: $S_1^{\oplus} \leftarrow\!\prec S_h^{\oplus} \rightarrow S_2^{\oplus}$, *sm* h = h_e where h_e: $S_{e_1}^{\oplus} \rightarrow S_{e_2}^{\oplus}$ and for all s in $S_1^{\oplus}$, if s is in $S_h^{\oplus}$, then $h_e(s) = h(s)$; else, $h_e(s) = e$. The functor *sm* is faithful and so, ⟨*pCSem, sm*⟩ is a concrete category over *CMon*. Also, for each finite diagram in *pCSem* taken by the functor *sm* into *CMon*, the limits and colimits in *CMon* can be lifted as an initial source and final sink, respectively, in *pCSem*.

The category of partial Petri nets is defined as follows: first consider the category of structured graphs where the base category is *pSet*, the category of arcs is *Set* and the category of nodes is *CSem* (and thus, the source and target functions are partial); then consider the category of partial morphisms of the previous category of structures graphs.

Definition 6. Partial Petri net. The category of partial Petri nets is *pPetri = pGraph(t, v)*, i.e., the category of partial morphisms on the category of structured graphs *Graph(t, v)*, where *t: Set* $\rightarrow$ *pSet* is the canonical embedding functor and *v: CSem* $\rightarrow$ *pSet* is the forgetful functor such that for all *CSem*-object $S^{\oplus} = \langle S^*, \oplus\rangle$, $v\, S^{\oplus} = S^*$ and for all *CSem*-morphism h: $S_1^{\oplus} \rightarrow S_2^{\oplus}$, v h: $S_1^* \rightarrow S_2^*$.

Thus, a partial Petri net N is a quadruple $N = \langle V^{\oplus}, T, \partial_0, \partial_0\rangle$ where $V^{\oplus}$ is a free commutative semi-group, T is a set and ∂_0, ∂_0: $t\,T \rightarrow v\,V^{\oplus}$ are partial functions. Let $N_1 = \langle V_1^{\oplus}, T_1, \partial_{0_1}, \partial_{1_1}\rangle$ and $N_2 = \langle V_2^{\oplus}, T_2, \partial_{0_2}, \partial_{1_2}\rangle$ be nets. From the definition of partial morphism, we infer that a *pPetri*-morphism $h: N_1 \leftarrow\!\!\prec D_h \rightarrow N_2$ is a pair $\langle h_V: V_1^{\oplus} \leftarrow\!\!\prec D_{h_V} \rightarrow V_2^{\oplus}, h_T: T_1 \leftarrow\!\!\prec D_{h_T} \rightarrow T_2 \rangle$ where h_V is a *pCSem*-morphism, h_T is a partial function and satisfies the commutative square diagram in Figure 6. Also, using the proposition about the square diagram in *pC*, it is easy to prove that the corresponding square diagram in is not a commutative diagram in *pSet*.

Proposition 7. The category *pPetri* is finitely complete e cocomplete.

Proof: The forgetful functor *v: pCSem → pSet* that takes each semi-group $S^{\oplus} = \langle S^*, \oplus \rangle$ into S^* has left adjoint which takes each set into the commutative semi-group freely generated. Thus, *v* preserves limits. Suppose k in $\{0, 1\}$. Then:

a) Let 0 and $0^{\oplus}$ be zero objects of *pSet* and *pCSem*, respectively. Then $\langle 0^{\oplus}, 0, !, ! \rangle$ where ! is the unique partial function, is a *zero object* of *pPetri*.

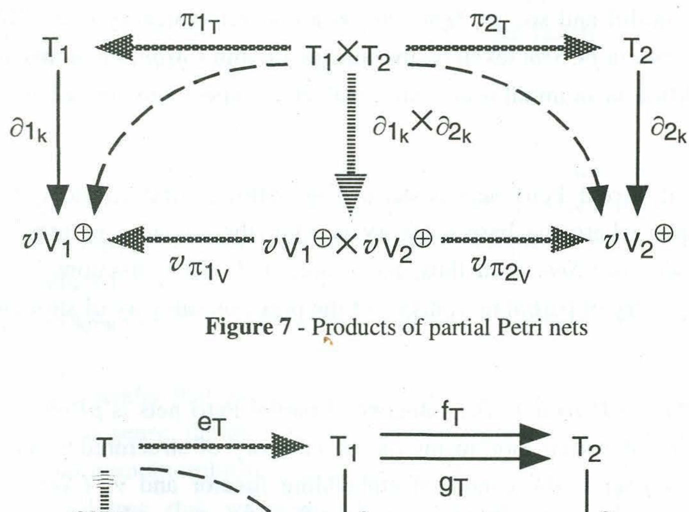

Figure 7 - Products of partial Petri nets

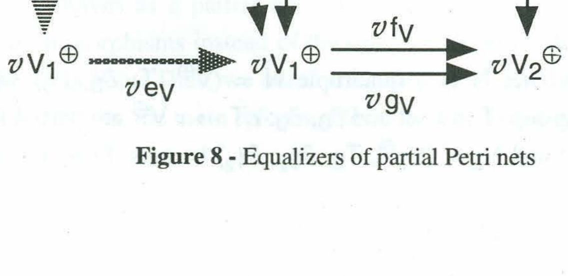

Figure 8 - Equalizers of partial Petri nets

b) Consider the nets $N_1 = \langle V_1^{\oplus}, T_1, \partial_{1_0}, \partial_{1_1}\rangle$ and $N_2 = \langle V_2^{\oplus}, T_2, \partial_{2_0}, \partial_{2_1}\rangle$. A *product* of N_1 and N_2 is the object $N_1 \times N_2 = \langle V_1^{\oplus} \times_{pCSem} V_2^{\oplus}, T_1 \times_{pSet} T_2, \partial_{1_0} \times \partial_{2_0}, \partial_{1_1} \times \partial_{2_1}\rangle$ together with the morphisms $\pi_1 = \langle \pi_{1_V}, \pi_{1_T}\rangle: N_1 \times N_2 \to N_1$ and $\pi_2 = \langle \pi_{2_V}, \pi_{2_T}\rangle: N_1 \times N_2 \to N_2$ where $\partial_{1_k} \times \partial_{2_k}$ are uniquely induced by the product in *pCSem*, taken into *pSet*, as illustrated in Figure 7 (remember that *v* preserves limits).

c) Consider the nets $N_1 = \langle V_1^{\oplus}, T_1, \partial_{1_0}, \partial_{1_1}\rangle$, $N_2 = \langle V_2^{\oplus}, T_2, \partial_{2_0}, \partial_{2_1}\rangle$ and a pair of parallel morphisms f, g: $N_1 \to N_2$ where $f = \langle f_V, f_T\rangle$, $g = \langle g_V, g_T\rangle$. Let $e_V: V^{\oplus} \to V_1^{\oplus}$ be a *pCSem*-equalizer of f_V, g_V and $e_T: T \to T_1$ be a *pSet*-equalizer of f_T, g_T. An *equalizer* of f, g is the net $N = \langle V^{\oplus}, T, \partial_0, \partial_1\rangle$ together with the morphism $e = \langle e_V, e_T\rangle: N \to N_1$ where $\Box_k$ are uniquely induced by the equalizer e_V in *pCSem*, taken into *pSet*, as illustrated in Figure 8 (again, remember that *v* preserves limits).

It is easy to prove that the above constructions are, in fact, zero object, product and equalizers in *pPetri*. The constructions for coproducts and coequalizers are analogous.

For instance, Figure 9 illustrates the resulting objects of a coproduct and product in *pPetri*.

3 PARTIAL PETRI NETS WITH INITIAL MARKING

A Partial Petri net with initial markings is a partial Petri net endowed with a set of initial markings where the choice of which initial marking is considered at run time is an external nondeterminism. The main advantage of considering a set of initial marking as in [7,10] instead of a single initial marking as in [23,16] is that the resulting category has finite colimits. This solution is more general than restricting the category for safe nets as in [23] or considering initial marking with one token at most in each place as in [16]. Moreover, the coproduct construction reflects the asynchronous composition of component nets.

Definition 8. Partial Petri Net with Initial Marking. Consider the category *pPetri*. Let *u: pPetri* $\to$ *pSet* be a functor such that each *pPetri*-net $N = \langle V^{\oplus}, T, \partial_0, \partial_1\rangle$ where $V^{\oplus} = \langle V^*, \oplus\rangle$ is taken into the set V^* and each *pPetri*-morphism $h = \langle h_V, h_T\rangle$ is taken into the partial function canonically induced by the *pCSem*-morphism h_V. The category of partial Petri nets with initial markings, denoted by *pMPetri*, is the comma category $id_{pSet} \downarrow u$, where id_{pSet} is the identity functor in *pSet*.

Therefore, a partial Petri net with initial markings M is a triple $M = \langle N, I, init\rangle$ where $N = \langle V^{\oplus}, T, \partial_0, \partial_1\rangle$ is a partial Petri net, I is the set of initial states or initial markings and init is the partial function which instantiates the initial states into the net N. Thus, a net M may also be considered as $M = \langle V^{\oplus}, T, \partial_0, \partial_1, I, init\rangle$. If init is the canonical inclusion, it may be omitted, i.e., $\langle V^{\oplus}, T, \partial_0, \partial_1, I, inclusion\rangle$ is abbreviated by $\langle V^{\oplus}, T, \partial_0, \partial_1, I\rangle$. A *pMPetri*-morphism is a pair $h = \langle h_N, h_I\rangle$. Since h_N is a pair $h_N = \langle h_V, h_T\rangle$, we also represent a *pMPetri*-morphism as a triple $h = \langle h_V, h_T, h_I\rangle$.

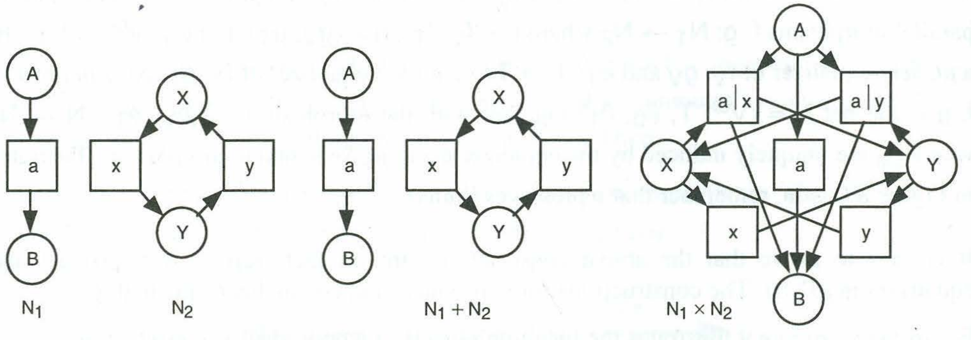

Figure 9 - Coproduct and product of partial Petri nets

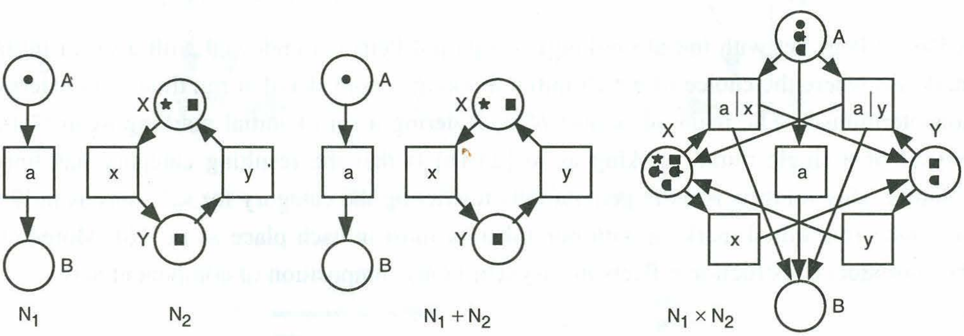

Figure 10 - Coproduct and product of partial Petri nets

Proposition 9. The category *pMPetri* is finitely complete and cocomplete.

Proof: Since *pMPetri* is the comma category $id_{pSet} \downarrow u$, we have only to prove that the functor *u: pPetri* → *pSet* preserves limits. Consider the initial object { } and the functor *p: pSet* → *pPetri* such that for all set V, *p* V is the net $\langle V^{\oplus}, \{\}, !, !\rangle$ where $V^{\oplus}$ is the semi-group freely generated from V. The functor *p* is left adjoint to *u*.

The product and coproduct in *pMPetri* have the same interpretation as in *pPetri*, i.e., the parallel composition and asynchronous composition, respectively. Figure 10 illustrates the resulting objects of a coproduct and product *pMPetri* where the set of initial markings are the following: $I_1 = \{A\}$, $I_2 = \{X, X+Y\}$, $I_1 + I_2 = \{A, X, X+Y\}$, $I_1 \times I_2 = \{A, X, X+Y, A+X, A+X+Y\}$. The possible initial markings in $I_1 \times I_2$ are represented using the following symbols:

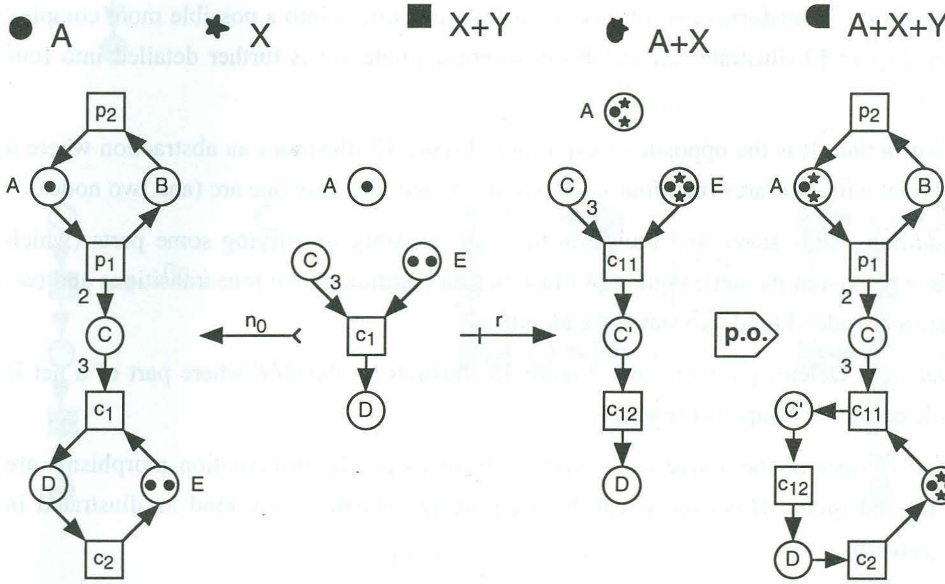

Figure 11 - Rule, instantiation and the transformed net

4 ANTICIPATORY PETRI NETS

The anticipation mechanism defined extends the single pushout approach of graph transformation to partial Petri nets.

Definition 10. Rule, Instantiation, Transformation. A rule $r: N_0 \to M_0$ and an instantiation $n_0: N_0 \to N$ are just *pMPetri*-morphisms. The transformation of a net N determined by a rule r and an instantiation n_0 is given by the pushout of r along with n_0 and $\varphi_{r,n_0}: N \to M$ is the transformation morphism where M is the transformed net.

For instance, consider the rule r, the instantiation n_0 and the transformed net, as in Figure 11. Entities preserved by morphisms are identified with the same label. Note that c_1 is replaced by a sequence of transitions c_{11}, c_{12} and that the state C' is introduced in the resulting net. With respect to the initial markings, the original one is preserved and a second marking is introduced.

A transformation of a Petri net may be classified in one of the follows cases:

a) *Expansion*. Transforms part of a net (usually a transition) into a possible more complex net. Figure 12 illustrates an expansion where a single arc is further detailed into four arcs.

b) *Abstraction*. It is the opposite of expansion. Figure 13 illustrates an abstraction where a sub-net with four arcs (and four nodes) is abstracted into only one arc (and two nodes).

c) *Addition*. Adds states and transition to a net, possibly identifying some parts (which already exist in the net). Figure 14 illustrates an addition where four transitions and two states are added (and two states are identified).

d) *Deletion*. Deletes parts of a net. Figure 15 illustrates a deletion where part of a net is deleted but the shape is preserved.

e) *Mix*. Neither of the above cases. In the above cases, the instantiation morphisms are total and mono. However, it can be a partial morphism of any kind as illustrated in Figure 16.

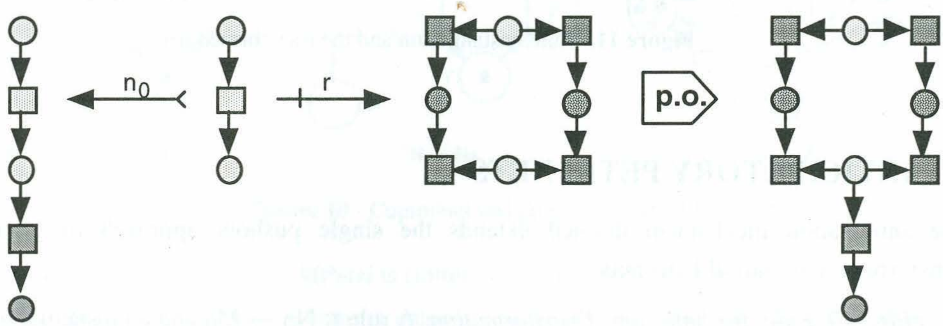

Figure 12 - Expansion

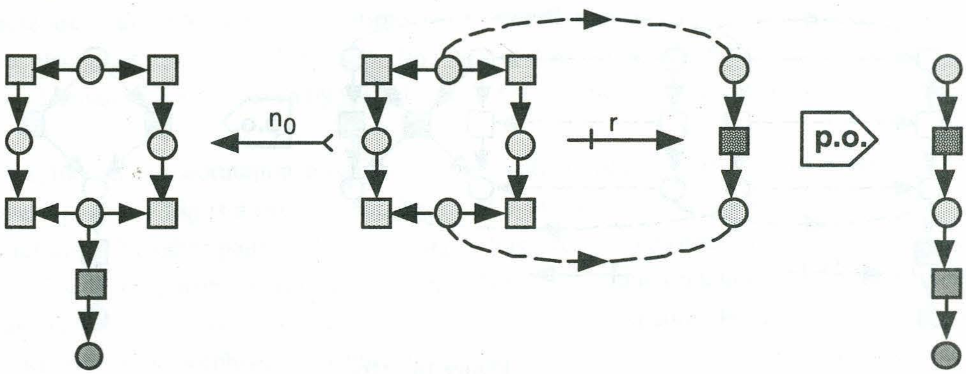

Figure 13 - Abstraction

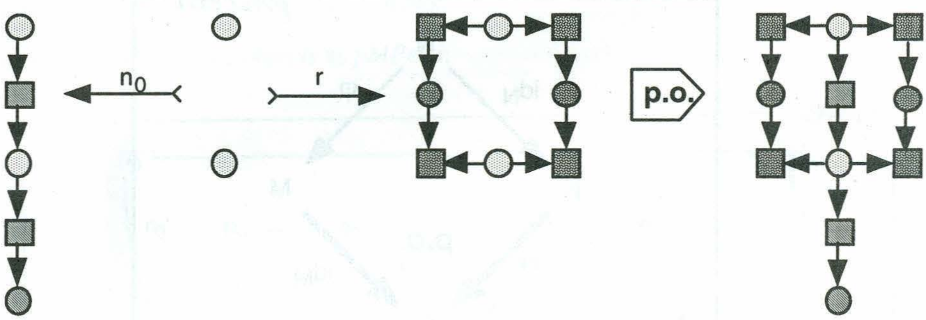

Figure 14 - Addition

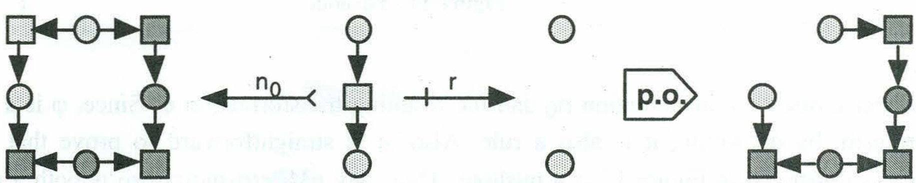

Figure 15 - Deletion

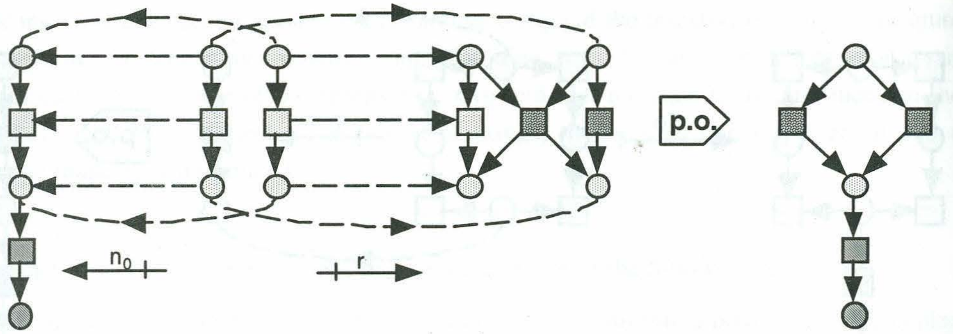

Figure 16 - Mix

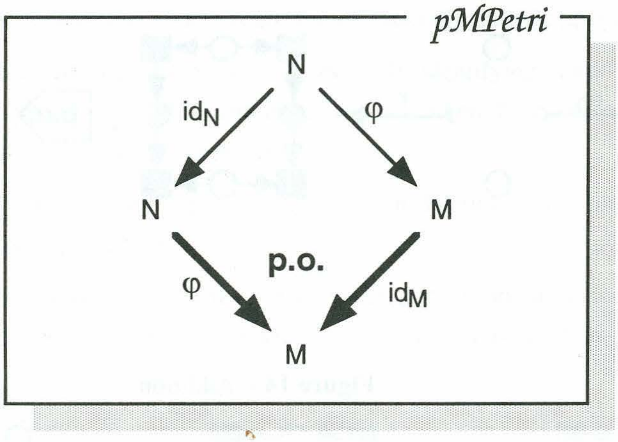

Figure 17 - Pushout

Consider a rule r, an instantiation n_0 and the resulting transformation φ. Since, φ is a net morphism, by definition, it is also a rule. Also, it is straightforward to prove that the diagram illustrated in Figure 17 is a pushout. Thus, any *pMPetri*-morphism is both a rule and a transformation.

Consider the rules r: $N_0 \rightarrow M_0$, s: $P_0 \rightarrow Q_0$, the instantiations n_0: $N_0 \rightarrow N$, p_0: $P_0 \rightarrow M$ and the transformations φ, ψ illustrated in Figure 18. The composition of ψ • φ should also be given by a pushout with rule r' and instantiation n_0' determined by r, s, n_0, p_0. In fact, there are many rules and instantiations which satisfy this requirement. But, since φ and ψ are also rules (determined by r, s, n_0, p_0), a pushout which results in the composed transformation $\psi \bullet \varphi$ is given by the rule $r' = \psi \bullet \varphi$ and the instantiation $n_0' = id_N$.

Therefore, a transformation morphism φ: N $\rightarrow$ M is fully determined by a pair $\langle r, n_0 \rangle$ where r: $N_0 \rightarrow M_0$ is a rule and n_0: $N_0 \rightarrow$ N is an instantiation. However, φ may also be determined by other pairs such as $\langle \varphi, id_N \rangle$. Thus, we may consider classes of equivalence of pairs of morphisms with respect to the relation "the transformations determined by the pushouts coincide". A class of equivalence is called a anticipation. Petri nets as objects and anticipation's as morphisms constitute the category *aMPetri.*

Definition 11. Category of Petri Nets and Anticipations. Consider the category *pMPetri.* The category *aMPetri* is defined as follows:

a) *aMPetri* has the same *objects* as *pMPetri*;

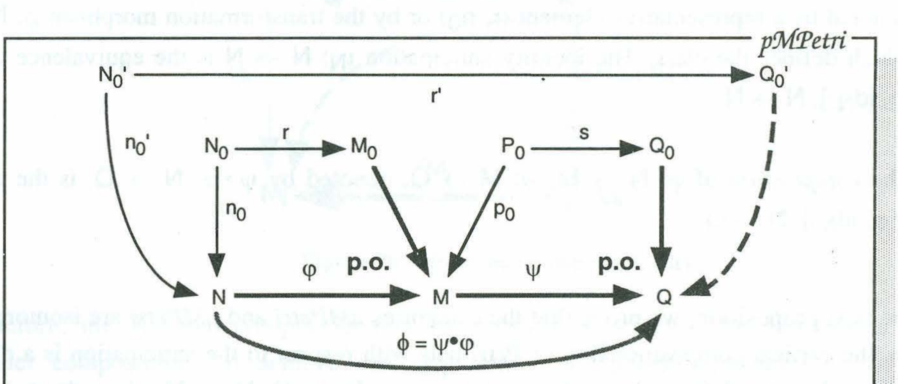

Figure 18 - Composition of transformations

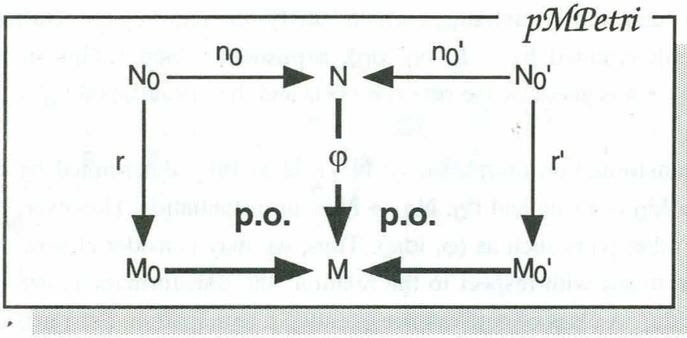

Figure 19 -Pushouts

b) A *morphism* in *aMPetri*, called *anticipation*, is an equivalence class of pairs of morphisms $\langle r: N_0 \to M_0, n_0: N_0 \to N\rangle: N \to M$ with respect to the relation $\langle r: N_0 \to M_0, n_0: N_0 \to N\rangle$ ant $\langle r': N_0' \to M_0', n_0': N_0' \to N\rangle$ if and only if the resulting pushouts determine the commutative diagram illustrated in Figure 19. A class $[\langle r, n_0\rangle]: N \to M$ may be denoted by a representative element $\langle r, n_0\rangle$ or by the transformation morphism $\varphi: N \to M$ which defines the class. The identity anticipation $\iota_N: N \to N$ is the equivalence class $[\langle id_N, id_N\rangle]: N \to N$;

c) The *composition* of $\varphi: N \to M$, $\psi: M \to Q$, denoted by $\psi \bullet \varphi: N \to Q$, is the class $[\langle \psi \bullet \varphi, id_N\rangle]: N \to Q$.

In the next proposition, we prove that the categories *aMPetri* and *pMPetri* are isomorphic. Thus, the vertical compositionality of Petri nets with respect to the anticipation is a direct corollary. In what follows, note that, for any class $[\langle r, n_0\rangle]: N \to M$ where $\langle r, n_0\rangle$ is a representative element, the pair $\langle \varphi_{r,n_0}, id_N\rangle$ is also an element of the class.

Proposition 12. The categories *aMPetri* and *pMPetri* are isomorphic.

<u>Proof</u>: Let *pa: pMPetri* $\to$ *aMPetri* be a functor such that for all net P , $\varphi: N \to M$ and $\psi: M \to Q$ we have that *pa* P = P, *pa* $id_P = [\langle id_P, id_P\rangle]$, *pa* $\varphi = [\langle \varphi, id_N\rangle]$ and *pa* $(\psi \bullet \varphi) = [\langle \psi, id_M\rangle \bullet \langle \varphi, id_N\rangle] = [\langle \psi \bullet \varphi, id_N\rangle]$. Let *ap: aMPetri* $\to$ *pMPetri* be a functor such that for all net P and for all $\varphi: N \to M$ and $\psi: M \to Q$ we have that *ap* P = P, *ap* $[\langle id_P, id_P\rangle] = id_P$, *ap* $[\langle \varphi, id_N\rangle] = \varphi$ and *ap* $[\langle \psi \bullet \varphi, id_N\rangle] = \psi \bullet \varphi$. Then $ap \bullet pa = id_{pMPetri}$ and $pa \bullet ap = id_{aMPetri}$.

Since *aMPetri* and *pMPetri* are isomorphic the composition of anticipations is straightforward and thus, the *vertical compositionality is achieved.* Also, we identify both categories by *pMPetri* and use the terms anticipation and transformation indifferently. A morphism φ: A $\rightarrow$ B which is a anticipation may also be represent as φ: A $\Rightarrow$ B. In the following proposition, we prove that the *horizontal compositionality of Petri nets is achieved,* i.e., the anticipation of nets distributes through the parallel composition (categorical product) of component nets.

Proposition 13. Let $\{\varphi_i: N_i \Rightarrow M_i\}_{i\in I}$ be an indexed set of *aMPetri*-anticipations, where I is a set. Then $\times_{i\in I}\varphi_i: \times_{i\in I} N_i \Rightarrow \times_{i\in I} M_i$.

Proof: Since *pMPetri* is complete, $\times_{i\in I}\varphi_i: \times_{i\in I} N_i \Rightarrow \times_{i\in I} M_i$ is the morphism uniquely induced by the product construction in *pMPetri*, as illustrated in Figure 20.

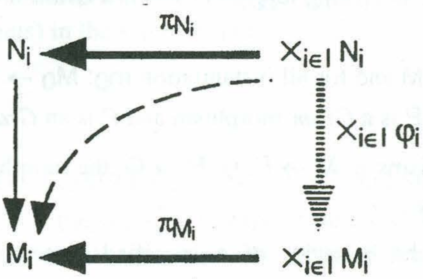

Figure 20 - Horizontal compositionality

Therefore, the diagonal compositionality requirement is achieved for anticipation and parallel composition. To achieve this requirement for general constructions of net combinators it is enough to extend the proposed approach using the synchronization mechanism introduced in [13] inspired by [22,16] where the categorical product stands for parallel combinator and a functorial operation defined using the fibration technique stands for synchronization. Basically, induced functor restricts the parallel composition according to given table of synchronizations.

Usually, for some given system, only some anticipations are desired. For this purpose, we introduce the specification grammar and the induced subcategory of anticipations. A specification grammar is basically an initial net and a collection of possible rules and instantiations. Each specification grammar induces a subcategory of *pMPetri* reflecting all possible nets that can be derived from the initial one. A specification grammar can be viewed as a specification of a given system and the induced subcategory as all possible dynamic anticipations of the system (objects) and their relationship (morphisms).

Definition 14. Specification Grammar. A specification grammar or just grammar is Gram = ⟨R, I, N⟩ where R, I are collections of *pMPetri*-morphisms representing the rules and instantiations of the grammar and N is an *pMPetri*-object called initial net.

Definition 15. Subcategory Induced by a Grammar. Let Gram = ⟨R, I, N⟩ be a grammar. The subcategory *Gram* of *pMPetri* induced by the grammar Gram is inductively defined as follows:

a) N is an *Gram*-object and $[\langle id_N, id_N \rangle]: N \to N$ is a *Gram*-morphism;

b) for all *Gram*-object M and for all instantiation $m_0: M_0 \to M$ and for all rule $r: M_0 \to P_0$, $[\langle r, m_0 \rangle]: M \to P$ is a *Gram*-morphism and P is an *Gram*-object;

c) for all *Gram*-morphisms $\varphi: M \to P$, $\psi: P \to Q$, the morphism $[\langle \psi \bullet \varphi, id_M \rangle]: M \to Q$ is a *Gram*-morphism.

The understanding of the behavior of a specified system, according to all possible reachable marking for all possible anticipation can be achieved through an adjunction to a category Nonsequential Automata as introduced in [15]. The details of this approach is left as a future work.

5 CONCLUSION

We construct a categorical semantic domain for Petri nets which satisfies the diagonal compositionality requirement, i.e., anticipations compose and distribute through net combinators. The anticipation mechanism is based on graph transformations using the single pushout approach. For this purpose, we introduce a finitely bicomplete category of partial Petri nets and partial morphisms and we claim that, with respect to partial morphisms, "Petri nets are semi-groups". Classes of net transformations stand for anticipations. The composition of anticipations (i.e., composition of pushouts) is defined, leading to a category of nets and anticipations which is also bicomplete. In this context, the diagonal compositionality is achieved. A specification grammar stands for a system specification and the induced subcategory of nets and anticipations can be viewed as all possible dynamic anticipations of the system (objects) and their relationship (morphims). Currently we are working on an adjunction between the categories in the proposed approach and categories of Nonsequential Automata as introduced in [15] to give a better understanding of a specified system, according to all possible reachable marking for all possible anticipations.

If an anticipation replaces part of a net by another net we can not expect that the original net and the resulting one are equivalent, according to some notion of direct simulation between all component transitions. Following the same idea, if an anticipation introduces new places we can not expect an equivalence according to some notion of observation of state changes. We are working on a notion of equivalence based on computations. Also, we will investigate the diagonal compositionality of anticipatory, concurrent systems equipped with reifications (refinements) in the sense of [15].

References

[1] A. ASPERTI and G. LONGO, Categories, Types and Structures - An Introduction to category Theoryfor the Working Computer Scientist, MIT Press, 1991.

[2] J. ADÁMEK, H. HERRLICH, and G. STRECKER, Abstract and Concrete Categories, Wiley, 1990.

[3] A. CORRADINI, An Algebraic Semantics for Transition Systems and Logic Programming, Ph.D. thesis, technical report TD-8/90, University of Pisa, 1990

[4] D. M. DUBOIS, Introduction to the Aristotle's Final Causation in CAST: Concept and Method of Incursion and Hyperincursion, EUROCAST'95, LNCS 1030, Springer-Verlag, pp. 477-493, 1996.

[5] D. M. DUBOIS, Introduction to Computing Anticipatory Systems, International Journal of Computing Anticipatory Systems Vol. 2, CHAOS, pp. 3-23, 1998.

[6] H. EHRIG, Introduction to the Algebraic Theory of Graph Grammars. LNCS 73, Springer-Verlag, pp. 1-69, 1979.

[7] B. JONSSON, A Hierarchy of Compositional Models of I/O-Automata, Symposium on Mathematical Foundations of Computer Science, pp. 347-354, LNCS 452, Springer-Verlag, 1990.

[8] M. LÖWE, Algebraic Approach to Single-Pushout Graph Transformation, TCS 109, pp. 181-224, 1993.

[9] S. Mac LANE, Categories for the Working Mathematician, Springer-Verlag, 1971.

[10] P. B. MENEZES, Marked Petri Nets Within a Categorial Framework, RITA - Revista de Informática Teórica e Aplicada, Vol. 2, No. 2, pp. 71-92, UFRGS/Brazil, 1995.

[11] P. B. MENEZES, Compositional Reification of Concurrent, Interacting Systems, PDPTA'98 - International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, pp. 1754-1761, CSREA Press, 1998.

[12] P. B. MENEZES, Diagonal Compositionality of Partial Petri Nets , 27 pp. submitted, 1998b.

[13] P. B. MENEZES & J. F. COSTA, Synchronization in Petri Nets, Fundamenta Informaticae Vol. 26.1, Annales Societatis Mathematicae Polonae, pp. 11-22, IOS Press, 1996.

[14] P. B. MENEZES & J. F. COSTA, Systems for System Implementation, Advances in Modeling of Anticipative Systems (G. E. Lasker, D. Dubois, B. Teiling, Eds.), pp. 84-88, International Institute for Advanced Studies in Systems Research and Cybernetics, Canada, 1996.

[15] P. B. MENEZES & J. F. COSTA, Compositional Refinement of Concurrent Systems, Journal of the Brazilian Computer Society - Special Issue on Parallel Computation, No. 1, Vol. 2, pp. 50-67, SBC/Brazil, 1995.

[16] J. MESEGUER & U. MONTANARI, Petri Nets are Monoids, Information and Computation 88, pp. 105-155, Academic Press, 1990.

[17] E.-R. OLDEROG, Nets, Terms and Formulas, Computer Tracts in Theoretical Computer Science 23, Cambridge University Press, 1991.

[18] J. PADBERG, H. EHRIG and L. RIBEIRO, Algebraic High-Level Net Transformation Systems, Mathematical Structures in Computer Science, Vol. 5, pp. 217-256, 1995.

[19] W. REISIG, Petri Nets: An Introduction, EATCS Monographs on Theoretical Computer Science 4, Springer-Verlag, 1985.

[20] E. ROBINSON & G. ROSOLINI, Categories of Partial Maps, Information and Computation 79, pp. 95-130, 1988.

[21] ROSEN. R., Anticipatory Systems, Pergamon Press, 1985.

[22] V. SASSONE, M. Nielsen & G. Winskel, A Classification of Models for Concurrency, CONCUR 93: 4th International Conference of Concurrency (E. Best, Ed.), pp. 82-96, LNCS 715, Springer-Verlag, 1993.

[23] G. WINSKEL, Petri Nets, Algebras, Morphisms and Compositionality, Information and Computation 72, pp. 197-238, Academic Press, 1987.

Acknowledgements

This work is partially supported by: FAPERGS (Project QaP-For), CNPq (Projects HoVer CAM, GRAPHIT) and CAPES (Project TEIA) in Brazil.

Este artigo recebeu o prêmio de Best Paper Award for The Symposium of Computing Systems and Soft Computing, durante a CASSYS'98, Second International Conference on Computing Anticipatory Systems, Liege, Bélgica, agosto de 1998, tendo sido publicado nos *proceedings* da conferência, editados pelo American Institute of Physics.