

# COMPUTING SCIENCE

Modelling Ambient Systems with Coloured Petri Nets

Alexandros Konios and Marta Pietkiewicz-Koutny

**TECHNICAL REPORT SERIES**

---

No. CS-TR-1377

March 2013

## **Modelling Ambient Systems with Coloured Petri Nets**

**A. Konios and M. Pietkiewicz-Koutny**

### **Abstract**

This paper is concerned with the formal modelling of Ambient Systems, focussing on the issues that emerge in their modelling using coloured Petri nets. A new class of coloured Petri nets, called Ambient Petri Nets (APNs), is introduced. Furthermore, a related step-modelling approach and its rationale are described. After that, the modularity and compositionality of the new class of Petri nets are discussed, and a method of composing APNs is introduced.

## Bibliographical details

KONIOS, A., PIETKIEWICZ-KOUTNY, M.

Modelling Ambient Systems with Coloured Petri Nets  
[By] A. Konios, M. Pietkiewicz-Koutny

Newcastle upon Tyne: Newcastle University: Computing Science, 2013.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1377)

### Added entries

NEWCASTLE UNIVERSITY  
Computing Science. Technical Report Series. CS-TR-1377

### Abstract

This paper is concerned with the formal modelling of Ambient Systems, focussing on the issues that emerge in their modelling using coloured Petri nets. A new class of coloured Petri nets, called Ambient Petri Nets (APNs), is introduced. Furthermore, a related step-modelling approach and its rationale are described. After that, the modularity and compositionality of the new class of Petri nets are discussed, and a method of composing APNs is introduced.

### About the authors

Mr A. Konios received his Bachelor degree in Computer Science from the University of Piraeus in 2009. In 2010, he graduated from Newcastle University, where he did his Master degree in Computer Security and Resilience. Currently he is studying for his PhD in Modelling and Verification of Ambient Systems under the supervision of Dr Marta Koutny. Other research interests include Formal Methods and System Validation.

Marta Pietkiewicz-Koutny is a Lecturer in the School of Computing Science, Newcastle University. Her areas of research interest include: modelling and validation of concurrent systems, synthesis of Petri nets from transition systems and abstraction and refinement in process networks.

### Suggested keywords

MODELLING  
AMBIENT SYSTEMS  
PETRI NETS  
MODULARITY  
COMPOSITIONALITY

# Modelling Ambient Systems with Coloured Petri Nets

Alexandros Konios and Marta Pietkiewicz-Koutny

School of Computing Science, Newcastle University  
Newcastle upon Tyne, NE1 7RU, U.K.

**Abstract.** This paper is concerned with the formal modelling of Ambient Systems, focussing on the issues that emerge in their modelling using coloured Petri nets. A new class of coloured Petri nets, called Ambient Petri Nets (APNs), is introduced. Furthermore, a related step-modelling approach and its rationale are described. After that, the modularity and compositionality of the new class of Petri nets are discussed, and a method of composing APNs is introduced.

## 1 Introduction

There is a growing need to introduce and develop formal techniques for computational models capable of faithfully modelling systems whose behaviour is of high complexity and concurrent.

One of these formal techniques is Petri nets. Petri nets are a modelling language that is used for the behavioural representation and analysis of real concurrent systems [11]. Since their introduction [10], Petri nets have been widely used in various areas, such as biology, software engineering and concurrent programming (see, e.g., [2, 6]).

Petri nets can be distinguished into two general categories, the low level and the high level Petri nets. The low level Petri nets consist of classes like Elementary nets or Condition/Event nets and Place/Transition nets [5]. Low level nets can usually represent the behaviour of systems with low or at most medium complexity. This does not mean that a complex system cannot be modelled with low level nets, but in such a case the complexity of the modelling and the analysis of that system would be higher. On the other hand, high level nets like Coloured Petri nets and Predicate/Transition nets, can model the behaviour of more complex systems. The advantage of high level nets over the low level nets is that they can depict the flow of different data types [4, 8].

An important and interesting subject of investigation in computing science are the recently introduced Ambient Systems which were originally developed in the late 1990's, and have already attracted the attention of the researchers. The tremendous advance of technology has contributed to the construction of such systems, which incorporate both ubiquitous and pervasive computing, resulting

in their rapid proliferation. This has resulted in the development of methods and techniques for the analysis of Ambient Systems [13, 14].

In this paper, we investigate the behaviour of Ambient Systems with the help of Petri nets. More precisely, we use a class of high level Petri nets to capture the behaviour of Ambient Systems.

In Section 3, we define a new class of coloured Petri nets, called Ambient Petri Nets (APNs). The intended application area for such nets are the general Ambient Systems. In Section 4, the modelling issues that derive from the development of the models of the Ambient Systems and the rationale of the modelling approach are discussed. Furthermore, modularity issues of the produced models, such as the building blocks, are presented in detail and the step-modelling approach is discussed. The last section deals with the composition of Ambient Petri Nets, by introducing the two composition operators for the APNs.

## 2 Preliminaries

In this section, we recall some basic notions related to multisets which we will use throughout the rest of the paper.

**Definition 1 (multisets).** A multiset  $m$  over a set  $S$  is a mapping  $m : S \rightarrow \mathbb{N}$ . (Intuitively,  $m(s)$  is the multiplicity of  $s \in S$  in  $m$ .) We will denote by  $\mu S$  the set of all the multisets over  $S$ .

We will use  $\mu^\infty S$  to denote the set of all extended multisets over  $S$ , i.e., mappings  $m : S \rightarrow \mathbb{N} \cup \{\infty\}$ . (Intuitively, the elements can now have an infinite multiplicity.)

For any (extended) multiset  $m$ ,  $\text{supp}(m) = \{s \in S \mid m(s) \neq 0\}$  is its support.

Sometimes we will use  $m = \{p_1, p_2, p_2\}$  to denote a multiset such that  $m(p_1) = 1$ ,  $m(p_2) = 2$  and  $\text{supp}(m) = \{p_1, p_2\}$ .

**Definition 2 (operations and relations on multisets).**

Let  $m$  and  $m'$  be multisets over  $S$ , and  $X \subseteq S$ .

- The sum of  $m$  and  $m'$  is the multiset  $m + m'$  over  $S$  such that, for all  $s \in S$ ,  $(m + m')(s) = m(s) + m'(s)$ .
- The difference between  $m$  and  $m'$  is the multiset  $m - m'$  over  $S$  such that, for all  $s \in S$ ,  $(m - m')(s) = \max\{0, m_1(s) - m_2(s)\}$ .
- $m$  is a sub-multiset of  $m'$  if, for all  $s \in S$ ,  $m(s) \leq m'(s)$ . We denote this by  $m \leq m'$ .
- $m|_X \in \mu X$  is the restriction of  $m$  to  $X$  if, for all  $s \in X$ ,  $m|_X(s) = m(s)$ .

## 3 Ambient Petri Nets

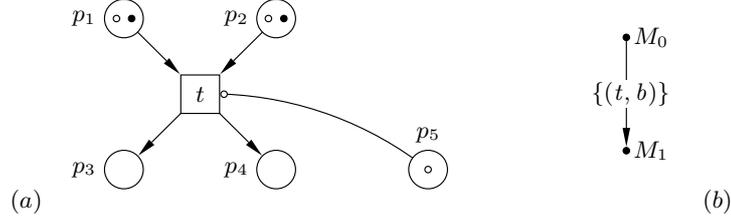
In this section, we introduce a class of nets aimed at the modelling of Ambient Systems.

**Definition 3 (Ambient Petri Net).** An Ambient Petri Net is a tuple

$$\mathcal{N} = (P, T, Pre, Post, I, Cl, C, K, M_0, G),$$

where:

- $P$  is a finite set of places.
- $T$  is a finite set of transitions disjoint from  $P$ .
- $Pre, Post : T \rightarrow \mu P$  are the pre and post mappings.
- $I \subseteq T \times P$  is a set of inhibitor arcs.
- $Cl$  is a non-empty finite set of non-empty colour sets. We will call it the structuring set.
- $C : P \cup T \rightarrow Cl$  is a colour function used to structure places and transitions:
  - $\tilde{P} = \{(p, g) \mid p \in P \wedge g \in C(p)\}$  is the set of structured places.
  - $\tilde{T} = \{(t, c) \mid t \in T \wedge c \in C(t)\}$  is the set of structured transitions.
- $K \in \mu^\infty \tilde{P}$  is an extended multiset defining the capacities of places.
- $M_0 \in \mu \tilde{P}$  is an initial marking satisfying  $M_0 \leq K$ . In general, any  $M \in \mu \tilde{P}$  such that  $M \leq K$  is a marking.
- $G \subseteq P$  is a set of gluing (or interface) places.



**Fig. 1.** An Ambient Petri Net  $\mathcal{N}$ , where the capacity  $K$  is a constant function 1 (a), and its concurrent reachability graph  $CRG(\mathcal{N})$  (b). In the diagram, each  $\circ$  token represents  $w$ , and each  $\bullet$  token represents  $b$ .

The meaning and graphical representation of  $P$  (places) and  $T$  (transitions) are as in the standard net theory. The set of places,  $G$ , specified as the last element of the tuple, will be important in the composition of Ambient Petri Nets. The directed arcs of the net are given by the  $Pre$  and  $Post$  mappings. For example, for the only transition of the APN of Figure 1 we have:  $Pre(t) = \{p_1, p_2\}$  and  $Post(t) = \{p_3, p_4\}$ . We will call  $p_1$  and  $p_2$  the *pre-places* of  $t$ , and  $p_3$  and  $p_4$  the *post-places* of  $t$ . In general, any  $p \in \text{supp}(Pre(t))$  is a pre-place of  $t$  and any  $p \in \text{supp}(Post(t))$  is a post-place of  $t$ . In this paper,  $Pre(t)$  and  $Post(t)$  will be sets for any  $t \in T$ . This means that every directed arc of an APN net

has the weight 1. We will also use the following ‘dot’ notation:

$$\begin{aligned} \bullet t &= \text{supp}(Pre(t)) && \text{(pre-set of } t \in T), \\ t \bullet &= \text{supp}(Post(t)) && \text{(post-set of } t \in T), \\ \bullet p &= \{t \in T \mid p \in \text{supp}(Post(t))\} && \text{(pre-set of } p \in P), \\ p \bullet &= \{t \in T \mid p \in \text{supp}(Pre(t))\} && \text{(post-set of } p \in P). \end{aligned}$$

An *inhibitor arc*  $(t, p) \in I$  will be represented in diagrams by an edge ending with a small circle, like  $(t, p_5)$  in Figure 1. We will call  $p_5$  an *inhibitor place* of transition  $t$ .

The  $Cl$  set is used for structuring places and transitions, by equipping them with colour sets. Places can hold coloured tokens, representing different kinds of resources or agents. The set of structured places is given by  $\tilde{P}$ . A structured transition  $(t, c) \in \tilde{T}$  represents an action that operates in a colour ‘mode’ given by  $c$ . In Figure 1,  $Cl = \{\{w\}, \{b, w\}\}$ , where  $b$  stands for the black colour and  $w$  stands for the white colour.  $\tilde{P}$  for the APN in Figure 1 is the set  $\{p_1, p_2, p_3, p_4\} \times \{b, w\} \cup \{(p_5, w)\}$  assuming that  $C(p_i) = \{b, w\}$  for  $i \in \{1, \dots, 4\}$  and  $C(p_5) = \{w\}$ . In diagrams, a structured place is represented by a coloured token that resides in this place. For example,  $(p_1, b)$  means that a black token was placed in place  $p_1$ . Also,  $\tilde{T}$  (for the APN in Figure 1) can be defined as  $\{(t, b), (t, w)\}$ , assuming that  $C(t) = \{b, w\}$ . Structured transitions only become apparent when we look at the net’s behaviour (see Figure 1(b)).

The initial marking  $M_0$  (which cannot exceed the capacities given by  $K$ ) specifies, for each place  $p$ , the number of tokens of each colour held in  $p$ . The initial marking  $M_0$  for the net in Figure 1 is  $\{(p_1, b), (p_1, w), (p_2, b), (p_2, w), (p_5, w)\}$ . In this paper, APNs will always be *safe coloured Petri nets*, because we will allow at most one token of a given colour per place. This is motivated by the fact that colours will be used as identities given to agents or other entities interacting within an ambient environment. As a consequence, the capacity  $K$  will be a constant function returning 1.

To define the semantics of an APN net we need to extend  $Pre$  and  $Post$  mappings using structured places and transitions.  $\widetilde{Pre}, \widetilde{Post} : \tilde{T} \rightarrow \mu\tilde{P}$  are defined as follows:

$$\widetilde{Pre}(t, c)(p, g) = \begin{cases} 0 & \text{if } c \neq g \\ Pre(t)(p) & \text{if } c = g \end{cases}$$

and

$$\widetilde{Post}(t, c)(p, g) = \begin{cases} 0 & \text{if } c \neq g \\ Post(t)(p) & \text{if } c = g \end{cases}$$

$\widetilde{Pre}(t, c)$  denotes, for every place  $p$ , the number of tokens of colour  $c$  that  $t$  needs from place  $p$  in order to fire.  $\widetilde{Post}(t, c)$  denotes, for every place  $p$ , the number of tokens of colour  $c$  that  $t$  will deposit in  $p$  after being fired. Notice that  $Pre(t)(p)$  and  $Post(t)(p)$  in the formulas above would be 1 for APN nets, as the weights of the directed arcs are all 1.

We can now extend  $\widetilde{Pre}$  and  $\widetilde{Post}$  functions to steps of structured transitions  $U = \{(t_1, c_1), \dots, (t_n, c_n)\} \subseteq \mu\widetilde{T}$  as follows:  $\widetilde{Pre}(U) = \widetilde{Pre}(t_1, c_1) + \dots + \widetilde{Pre}(t_n, c_n)$ , and similarly for  $\widetilde{Post}$ . The above extension is needed as the proposed semantics will be a *step semantics* rather than a *sequential semantics*.

Furthermore, we can define a set of *colour sensitive inhibitor arcs* based on the set of inhibitor arcs  $I$ :

$$\widetilde{I} = \{((t, c), (p, c)) \mid (t, p) \in I \wedge c \in C(t) \cap C(p)\}.$$

The meaning of a colour sensitive inhibitor arc will depend on the colours with which we equip transition  $t$  and the tokens in its inhibitor place  $p$ . The arc will be blocking transition  $t$  if  $t$  is in mode  $c$  and there is a token  $c$  in  $p$ , and will have no effect on  $t$  otherwise.

Now we can describe the semantics of an ambient Petri net  $\mathcal{N}$ . A step of structured transitions  $U$  is *enabled* at marking  $M$  if the following hold:

- $\widetilde{Pre}(U) \leq M$ .
- $M + \widetilde{Post}(U) \leq K$ .<sup>1</sup>
- $(U \times M) \cap \widetilde{I} = \emptyset$ .<sup>2</sup>

We denote this by  $M[U]$ . An enabled step  $U$  can fire producing the marking  $M' = M - \widetilde{Pre}(U) + \widetilde{Post}(U)$ . This will be denoted by  $M[U]M'$ . That means that a step is enabled if the pre-places of all the transitions in the step, working in certain colour modes (taking into account their multiplicities), have sufficient number of tokens of appropriate colour. Additionally, for the step to be enabled, none of the transitions of the step should have its inhibitor places (if any) marked by tokens of the colour equivalent to its working mode colour. For the APN in Figure 1, the step  $\{(t, b)\}$  is enabled at  $M_0$  and can fire, removing black tokens from  $p_1$  and  $p_2$ , and adding one black token to each of  $p_3$  and  $p_4$ . However, the steps  $\{(t, w)\}$  and  $\{(t, b), (t, w)\}$  are not enabled at  $M_0$  and cannot be fired as  $((t, w), (p_5, w)) \in \widetilde{I}$ . Furthermore, a step with more than one structured transition  $(t, b)$ , like  $\{(t, b), (t, b)\}$ , is also not enabled, because of a lack of sufficient number of black tokens in  $p_1$  and  $p_2$ .

The full execution semantics of an APN will be captured using a transition system where arcs are labelled by executed steps.

**Definition 4.** A step transition system over  $T$  is a triple  $STS = (Q, \Delta, q_0)$  consisting of a set of states  $Q$ , set of arcs  $\Delta \subseteq Q \times \mu\widetilde{T} \times Q$ <sup>3</sup>, and the initial state  $q_0 \in Q$ . We assume that each state  $q$  is reachable, i.e., there are steps  $U_1, \dots, U_n$  and states  $q_0, q_1, \dots, q_n = q$  such that  $n \geq 0$  and  $(q_{i-1}, U_i, q_i) \in \Delta$  for  $1 \leq i \leq n$ .

<sup>1</sup> If we want to allow self-loops in the net, this condition should take a weaker form:  $M - \widetilde{Pre}(U) + \widetilde{Post}(U) \leq K$ .

<sup>2</sup> This condition means that there are no  $(t, c) \in U$  and  $(p, c) \in M$  such that  $((t, c), (p, c)) \in \widetilde{I}$ .

<sup>3</sup> If we consider our steps to be sets rather than multisets, we can replace  $\mu\widetilde{T}$  with  $2^{\widetilde{T}}$ .

The *concurrent reachability graph*  $CRG(\mathcal{N})$  of  $\mathcal{N}$  is a step transition system  $CRG(\mathcal{N}) = ([M_0], \Delta, M_0)$  over  $T$  where:

$$[M_0] = \{M_n \mid \exists U_1, \dots, U_n \exists M_1, \dots, M_{n-1} \forall 1 \leq i \leq n : M_{i-1}[U_i]M_i\} \quad (1)$$

is the set of reachable markings and  $(M, U, M') \in \Delta$  iff  $M[U]M'$ . Figure 1(b) shows the concurrent reachability graph of the APN net in Figure 1(a). Furthermore, we will call  $U_1 \dots U_n$ , as in the formula (1), a *step sequence* and write  $M_0[U_1 \dots U_n]M_n$ .

Notice that although in many introduced notions we used multisets, only sets were really required (like in *Pre* and *Post* mappings, for example).

## 4 Modelling with APNs

In the previous section, we described Ambient Petri Nets (APNs), a subclass of the Colour Petri Nets resulting from the modelling of the case studies in [9]. We will now consider how this new class can be used in the modelling of the Ambient Systems.

In the modelling of Ambient Systems with APNs, we intend to concentrate on the interactivity between the system and its user, as the functioning of Ambient Systems is based on such interaction. Therefore, it is crucial to examine, through formal models, how the user can affect the system's actions and vice versa.

The interactivity of Ambient Systems will be expressed through the detection or prediction of the user's action, or by notifications or feedbacks that the system provides to the users after their actions. Apart from interactivity, we need to examine another interesting characteristic of these systems, viz. context-awareness, which affects the ability of the system to interact with the user. In order to model the interactivity and context-awareness effectively, we had to decide what system components should be explicitly modelled.

Our study of Ambient Systems led to a realisation that the interaction between the user and the system takes place through components like the sensors as well as the input and output devices. Basically, these are the means of the expression of the interactivity that the system has in its disposal in order to communicate with the user. Thus, we will model them in the context of the user's behaviour. Another direction of our research is the study of modularity issues that could be inspired by the modelling of our case studies. In particular, we extracted building blocks that could be easily reused in the construction of general Ambient System.

As a result of our previous investigation, we now will introduce a 'step-modelling' approach for Ambient Systems aimed at the construction of APNs from basic building blocks. In the following sections we will also describe in detail the rationale behind such an approach.

### 4.1 Step-modelling Approach

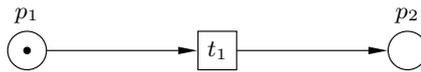
The name of the step-modelling approach is derived from the logic that governs the majority of Ambient Systems, namely the user cannot be at two different

places or perform two different important tasks at the same time. So, a user, as an entity usually executes actions *step by step*. On the other hand, the user can execute tasks (actions) concurrently with the system's actions and with other users' actions.

During the modelling of our case studies [9], we extracted a fundamental building block that can be used for the development of models of other Ambient Systems. This building block describes a single action of the user, and the way in which this particular action leads to the change of the state for both the system and the user. The following subsections will explain the structure of the building block and the overall rationale behind the step-modelling approach.

**Unidirectional Step.** Starting the description of the step-modelling approach, we introduce the unidirectional step, or unidirectional step system, which is part of the extracted building block.

*A unidirectional step is used when an action is permitted to happen only in one direction. In this case, the user or the system cannot return to the previous condition/state. An example of such a step is given in Figure 2.*



**Fig. 2.** A unidirectional step.

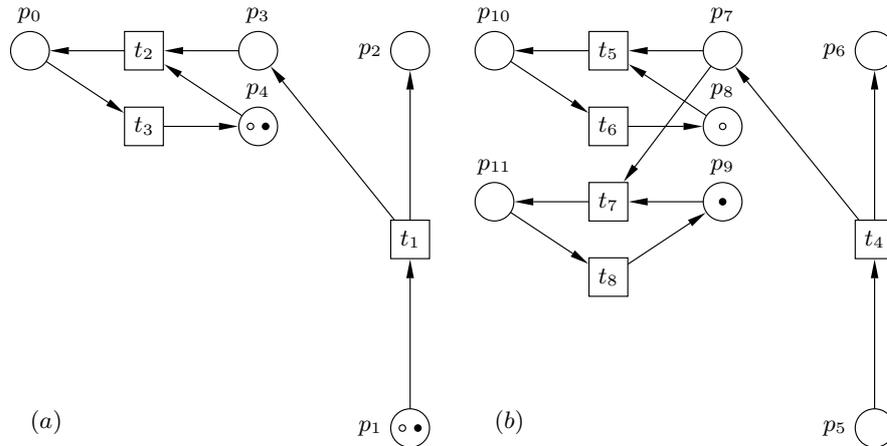
Note that the unidirectional step in Figure 2 is a very simple net, consisting of two places and one transition. In order to distinguish these places and the transition, we will use the notation  $P_{ST} = \{p_1, p_2\}$  and  $T_{ST} = \{t_1\}$ . A formal definition of a unidirectional step will be given later in this section.

**Unidirectional Step with Control System.** Having introduced the unidirectional step — the first part of the extracted building block — we now specify the construction of the control system which decides how the system should respond depending on the information received through the action of the user.

*The control system of a unidirectional step is used to synchronise the user's action with the response of the system.*

The structure and the function of the control system will be described through examples.

Figure 3 shows two unidirectional steps with control systems. Assume that the net (b) represents a step that follows the step described by the net (a). We will now explain the working of the system composed of these two unidirectional steps, shown in Figure 4 (the composition of steps will be formally defined in Section 5).



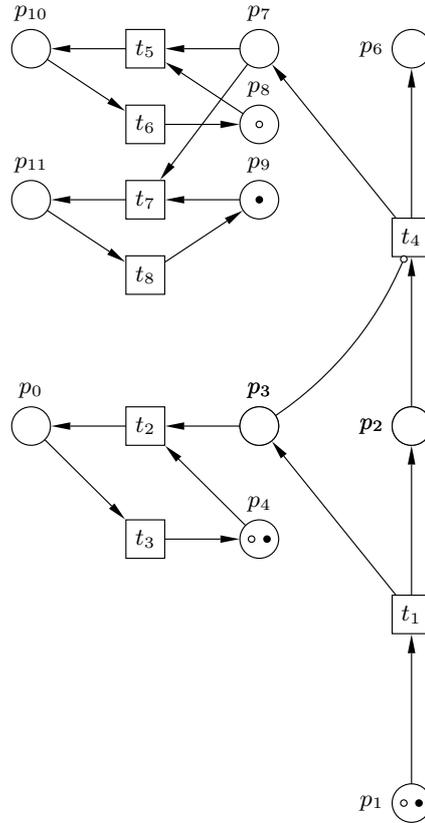
**Fig. 3.** A unidirectional step with control system (a), and unidirectional step with control system and multiple decision points (b).

The following simulation of the combined steps demonstrates the functioning of the control system. We begin the simulation from place  $p_1$ , assuming that the token game starts with the firing of transition  $t_1$  for the black token. By executing the transition, the black token is removed from place  $p_1$  and one black token is deposited in each of the  $p_2$  and  $p_3$  places. At this point, we will say that  $p_3$  ‘controls’  $p_2$  as transition  $t_4$  cannot fire when there are tokens of the same colour in  $p_3$  and  $p_2$  due to the colour-sensitive inhibitor arc between  $p_3$  and  $t_4$  (see Figure 4). The only way to execute transition  $t_4$  (in the black mode) is to remove the black token from place  $p_3$  (by executing  $t_2$ ). In other words, the only way for the user to execute the next task is to obtain first the appropriate response from the system. This will become clearer when we describe the control system of the building block.

The control system (CS) of a unidirectional step building block is a component of the net consisting of:

- the *control place* ( $p_c$ ),
- the *database places* ( $P_d$ ),
- the *response places* ( $P_r$ ) and
- transitions whose pre- and post-places are either control, database or response places.

The control place ( $p_c$ ) is used for the control of the corresponding place of the step (i.e., the post-place of the transition of the step). In other words, there is a one-to-one correspondence between the control places and the post-places of transition of the steps. The control place is also used for the depositing of tokens (reflecting the users actions) that will help the system to give an appropriate response to the users. It is important to note that each control system has only one control place.



**Fig. 4.** A composition of two unidirectional step building blocks.

The database places ( $P_d$ ) store in the form of tokens all the useful information about the configuration of the system. For instance, after firing a transition that has as pre-places the control place and a database place, the system can respond to the user's action according to the information that is given through the token that resides in the control place. For example, in Figure 3(b), if we assume that a black token is stored in the control place, in this example  $p_7$ , then the system will check the colour of the token, will also check its database places, and will then execute the transition that is enabled giving the proper feedback to the user. In this case, the system realises that the colour of the token is black, checks the database place  $p_9$  (where the information about the black token is located) and finds out that the user associated with the black token must be provided with the information that is indicated by the post-place  $p_{11}$  of the executed transition  $t_7$ .

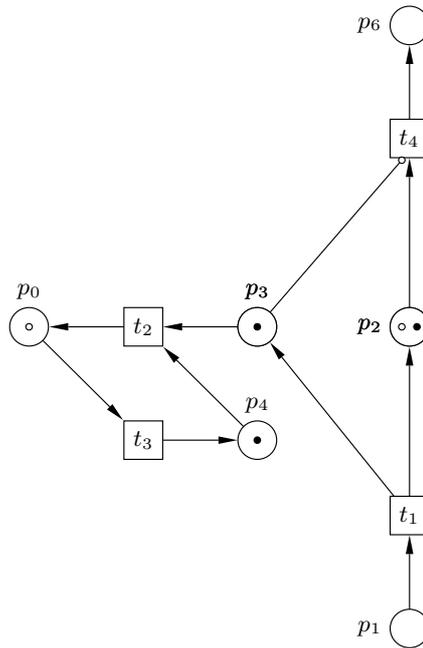
The response places ( $P_r$ ) usually represent the feedback that is given to the user through the output devices of the system (public displays, private displays, etc.). An example of a response place could be  $p_{11}$ , which was mentioned earlier.

The number of the response and database places of a control system is determined by the number of different options that the user has after making the associated step. For instance, suppose that the first step of Figure 4 represents a car that moves from the entrance of a garage ( $p_1$ ) to another position ( $p_2$ ). Suppose also that when the car is at  $p_2$ , it can only go forward. Then the system will have only one database place that includes all the users and one response place that shows the same information to all of them. Continuing with the second step of Figure 4, assume that at place  $p_6$  each user has three options (go left, right or forward). When a (black) user is at place  $p_6$ , the system must provide a correct feedback regardless of what will be the choice of the user after they receive it. In this example, the system has allocated a parking spot to the black user and depending on where that parking place is (according to the layout of the garage), it gives directions to that user. Thus, being at place  $p_6$ , the feedback of the system should advise the user to go in the appropriate direction, by executing transition  $t_7$ . However, as explained in the description of Ambient Systems [9], users have always the initiative, which means that they are not obliged to follow the advice of the system. On the other hand, the system always has to provide the correct information to the users. Regarding the feedback that is given to the users through the response places of the control system, it should be noticed that the labels of the places might represent the nature of the feedback (e.g. instructions like: forward, backward, right or left in case of a guidance system) and the marking of them indicates whether the feedback was given or not.

To complete the description of the control system, we need to explain the role its transitions. There can be two types of transitions: the retrieve transitions ( $T_r$ ) and the emptying transitions ( $T_{em}$ ). The retrieve transitions have as pre-places the response places and as post-places the database places. These transitions are used for the refill of the database places with the tokens that have been removed recently. The importance of the refill process lies in the fact that the database places can be used more than once by the same user if that user wants to repeat the same action again and again. On the other hand, the emptying transitions have as pre-places the database places and the control place and as post-places the response places. The emptying transitions are responsible for the feedback that is given to the user as their firing results in the depositing of the token into the response places. Furthermore, the emptying transitions control the state of the control place.

Another important issue is the colour-sensitive inhibitor arc, the formal definition of which is given in Section 3. Here we simply describe its usefulness for our model, and the difference between it and the ordinary inhibitor arc [7]. The colour sensitive inhibitor arc prevents the user from taking another action before the response of the system. We could say that this restriction (the use of colour-sensitive inhibitor arc) synchronises the system's actions with those of the user (the user acts and the system reacts). The difference between the colour-sensitive inhibitor arc and the standard inhibitor arc is that a colour-sensitive arc permits the firing of a transition for all the data types (colours) of the tokens except for those that reside in the inhibitor places of the transition. The example

of Figure 5 demonstrates the operation of a colour-sensitive inhibitor arc. The transition  $t_4$  can be fired in the white mode but not in the black one. This is due to the fact that a black token is in place  $p_3$  and blocks the execution of the transition for that particular colour of token. We can interpret this situation by saying that the ‘white’ user has obtained a response from the system (see place  $p_0$ ) and can proceed to the next step, while the ‘black’ user still awaits system’s instruction and must wait. Contrary to the colour-sensitive inhibitor arc, the standard inhibitor arc blocks the firing of a transition for any token that would be present in its inhibitor places. Generally, the standard inhibitor arcs lack this extra flexibility that is not needed in the P/T nets [3], where there is only one type of tokens.



**Fig. 5.** A colour-sensitive inhibitor arc.

## 4.2 Basic Step Nets

We will now define formally the basic step nets which capture the intuitive concept of a unidirectional step. Their definition follows from the above discussion about an unidirectional step and its components: the step system and the control system. As basic step nets will be used as building blocks of Ambient Systems, we will leave the places of a step sub-system unmarked, only requiring the database places to be marked. The initial state of an ambient system will be provided by a *root net* described at the end of this section.

**Definition 5 (basic step net).** A basic step net is an APN net

$$\mathcal{N}_S = (P_S, T_S, Pre_S, Post_S, I_S, Cl, C_S, K_S, M_0^S, G_S)$$

consisting of two parts: step system and control system satisfying the following structural conditions:

- $P_S = P_{ST} \uplus P_{CS}$ , where  $P_{ST}$  is a set of places of the step system and  $P_{CS}$  is a set of places of the control system. More specifically,  $P_{ST} = \{p_s, p_f\}$  contains the starting ( $p_s$ ) and finishing ( $p_f$ ) place of the unique step transition  $t_s$ , while  $P_{CS} = P_r \uplus P_d \uplus \{p_c\}$  is built out of three subsets such that (below  $n \geq 1$ ):
  1.  $P_r = \{p_r^1, \dots, p_r^n\}$  is a set of response places.
  2.  $P_d = \{p_d^1, \dots, p_d^n\}$  is a set of database places.
  3.  $p_c$  is a unique control place of the finishing place  $p_f$  associated with transition  $t_s$ . We will denote this by  $p_c = cp^{t_s}(p_f)$ .
- $T_S = T_{ST} \uplus T_{CS}$ , where  $T_{ST}$  is a set of transitions of the step system and  $T_{CS}$  is a set of transitions of the control system. More specifically,  $T_{ST} = \{t_s\}$  contains a unique transition, called a step transition.  $T_{CS} = T_r \uplus T_{em}$  is built out of two subsets:
  1.  $T_r = \{t_r^1, \dots, t_r^n\}$  is a set of retrieve transitions.
  2.  $T_{em} = \{t_{em}^1, \dots, t_{em}^n\}$  is a set of emptying transitions.
- $Pre_S, Post_S : T_S \rightarrow \mu P_S$  are such that:

$$\begin{aligned} Pre_S(t_{em}^i) &= \{p_c, p_d^i\} && \text{for } i = 1, \dots, n. \\ Post_S(t_{em}^i) &= \{p_r^i\} && \text{for } i = 1, \dots, n. \\ Pre_S(t_r^i) &= \{p_r^i\} && \text{for } i = 1, \dots, n. \\ Post_S(t_r^i) &= \{p_d^i\} && \text{for } i = 1, \dots, n. \\ Pre_S(t_s) &= \{p_s\}. \\ Post_S(t_s) &= \{p_f, p_c\}. \end{aligned}$$

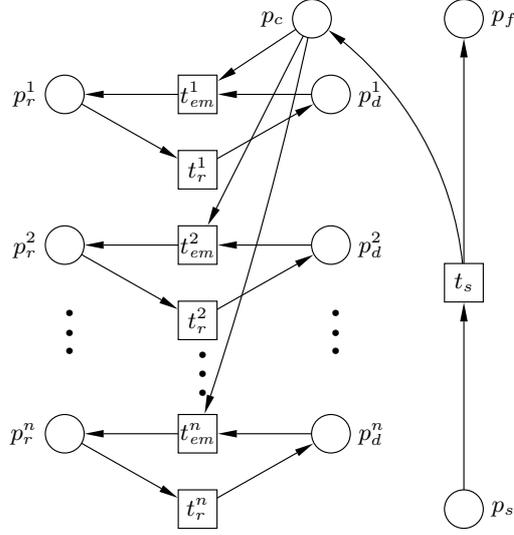
- $I_S = \emptyset$ .
- All the places except for the database places are unmarked. The database places are marked by at least one token of some allowed colour.

$$M_0^S(p, c) = 0, \text{ for every } p \in P_{ST} \cup P_r \cup \{p_c\} \text{ and } c \in C_S(p).$$

$$M_0^S(p, c) \neq 0, \text{ for every } p \in P_d \text{ and some } c \in C_S(p).$$

- $G_S = \{p_s\}$ .

**Definition 6 (root net).** A root net is defined as an APN net that has two places: a place  $p$  that acts as both starting and finishing place of a ‘collapsed’



**Fig. 6.** A basic step net.

step and its control place  $p_c = cp^\epsilon(p)$ , where  $\epsilon$  denotes the ‘collapsed’ transition of the collapsed step.<sup>4</sup> The root net will be denoted by:

$$\mathcal{N}_R = (P_R, T_R, Pre_R, Post_R, I_R, Cl, C_R, K_R, M_0^R, G_R),$$

where

- $P_R = \{p, p_c\}$ ,
- $T_R = \emptyset$ ,
- $Pre_R, Post_R$  are empty functions,
- $I_R = \emptyset$ ,
- $G_R = \{p\}$ .

A root net is used to provide the initial information of the system by marking each of its places with some chosen set of tokens. The place  $p$  will be marked with the set of initial users of the system. For example, in Figure 7(a, b), we have two users represented by black and white tokens. The control place,  $p_c$ , is provided to define the initial restrictions. The root net of Figure 7(a) states no restrictions about any of the allowed users, while the root net of Figure 7(b) states restrictions for the ‘black’ user: this user will be stopped from proceeding any further. For example, in the Ambient Parking Garage model [9] this might mean that the owner of the black car has not paid the fee and is not allowed to use the garage.

<sup>4</sup> We can extend our notation to:  $p \in \bullet \epsilon$  and  $p \in \epsilon \bullet$ .



**Fig. 7.** Root nets  $(a, b)$ .

## 5 Composition of APNs

In the previous section, we introduced the building block (or basic step net) that was extracted from the modelling process of our case studies [9]. This simple net is composed of the Control System part and the Step part, and will be used for the compositional construction of APNs.

The basic step net (see Section 4) represents the user's actions and the system's response to these actions. The Control System of the net corresponds to the system's response and the Step part to the user's action. The general structure of a basic step net is shown in Figure 6. It will be used later on in examples that demonstrate how the formal definition of the composition of the APNs works. The repetitive composition of 'steps' represented by the structure of Figure 6 will result in the construction of behavioural models of Ambient Systems.

Composition is a fundamental technique for constructing large system models out of simpler ones. A number of different approaches to the composition of Petri nets have been developed in the past years, based on composition of places or transitions [1, 12]. In this section, we will introduce two composition operators for the APNs: the *forward* composition operator, and the *backward* composition operator.

The composition of APNs described in this section is based on the gluing of net places. Moreover, during the composition colour-sensitive inhibitor arcs between some of the control places of the first net and the step transition of the second net are generated.

### Definition 7 (forward composition).

Let us assume that

$$\mathcal{N} = (P, T, Pre, Post, I, Cl, C, K, M_0, G)$$

is an ambient Petri net such that, for every place  $g \in G$  and  $t \in \bullet g$ , there is a control place in  $t^\bullet$  denoted by  $cp^t(g)$ . Moreover, let

$$\mathcal{N}_S = (P_S, T_S, Pre_S, Post_S, \emptyset, Cl, C_S, K_S, M_0^S, G_S)$$

be a basic step net (as in Definition 5) such that  $P \cap P_S = T \cap T_S = \emptyset$ , and let  $g \in G$  be a place satisfying:  $C(g) = C_S(p_s) = C_{SET}$  and  $K(g, c) = K_S(p_s, c)$ , for every  $c \in C_{SET}$ .<sup>5</sup>

<sup>5</sup> Note that  $p_s$  is the only gluing place of  $\mathcal{N}_S$ .

The forward composition of  $\mathcal{N}$  and  $\mathcal{N}_S$  w.r.t. the gluing pair  $(g, p_s)$  of places is an ambient Petri net

$$\mathcal{N} \oplus^g \mathcal{N}_S = \mathcal{N}' = (P', T', Pre', Post', I', Cl, C', K', M'_0, G'),$$

where the different components are defined as follows:

- $P' = P \cup (P_S \setminus \{p_s\})$  and  $T' = T \cup T_S$ .
- $Pre', Post' : T' \rightarrow \mu P'$  are defined by:

$$Pre'(t)(p) = \begin{cases} Pre(t)(p) & \text{for } t \in T \text{ and } p \in P, \\ Pre_S(t)(p) & \text{for } t \in T_S \text{ and } p \in P_S \setminus \{p_s\}, \\ Pre_S(t)(p_s) & \text{for } t \in T_S \text{ and } p = g, \\ 0 & \text{otherwise.} \end{cases}$$

and

$$Post'(t)(p) = \begin{cases} Post(t)(p) & \text{for } t \in T \text{ and } p \in P, \\ Post_S(t)(p) & \text{for } t \in T_S \text{ and } p \in P_S \setminus \{p_s\}, \\ Post_S(t)(p_s) & \text{for } t \in T_S \text{ and } p = g, \\ 0 & \text{otherwise.} \end{cases}$$

- $I' \subseteq T' \times P'$  is defined by:  $I' = I \cup \{(t_s, cp^t(g)) \mid t \in \bullet g \text{ in } \mathcal{N}\}$ , where  $t_s \in T_S$  is  $\mathcal{N}_S$ 's unique step transition.
- $C' : P' \cup T' \rightarrow Cl$  is defined by:  $C'|_{P \cup T} = C$  and  $C'|_{(P_S \setminus \{p_s\}) \cup T_S} = C_S$ .
- $K' \in \mu^\infty \widetilde{P'}$  is defined by:  $K'|_{\widetilde{P}} = K$  and  $K'|_{\widetilde{P_S \setminus \{p_s\}}} = K_S|_{\widetilde{P_S \setminus \{p_s\}}}$ .
- $M'_0 \in \mu \widetilde{P'}$  is defined by:  $M'_0|_{\widetilde{P}} = M_0$  and  $M'_0|_{\widetilde{P_S \setminus \{p_s\}}} = M_0^S|_{\widetilde{P_S \setminus \{p_s\}}}$ .
- $G' = G \cup \{p_f\}$ ; moreover, the control places of the gluing places in  $G'$  are inherited from the component nets.

Although in the above definition  $\mathcal{N}$  is a general ambient Petri net, for the composition operator to work as desired,  $\mathcal{N}$  should have a specific structure. More precisely, it should be a *forward composite step net*. In what follows,  $X_{Basic}$  denotes all the basic step nets, and  $X_{Root}$  all the root nets.

**Definition 8 (forward composite step nets).** *The set of forward composite step nets  $X_{Com}^F$  is defined inductively as follows:*

- $X_{Root} \subset X_{Com}^F$ .
- If  $\mathcal{N} \in X_{Com}^F$  and  $\mathcal{N}_S \in X_{Basic}$ , then  $\mathcal{N} \oplus^g \mathcal{N}_S \in X_{Com}^F$ , provided that the composition is well-defined.

Note that for every gluing place of a net  $\mathcal{N} \in X_{Com}^F$  there will be at least one control place. Indeed, the only gluing place of a root net has its unique control place associated with collapsed transition  $\epsilon$ , and this transition can be considered both as pre- and post-transition of this gluing place. Also, the place

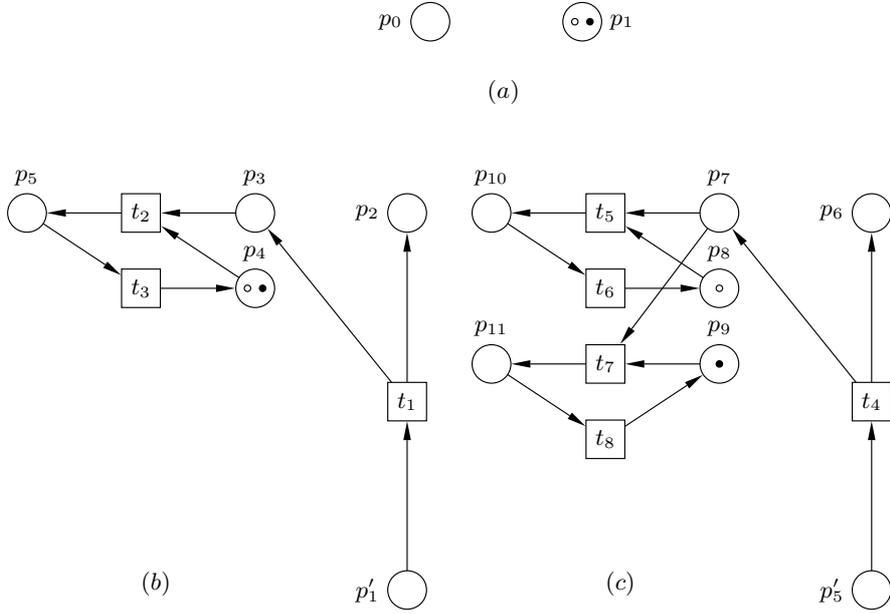
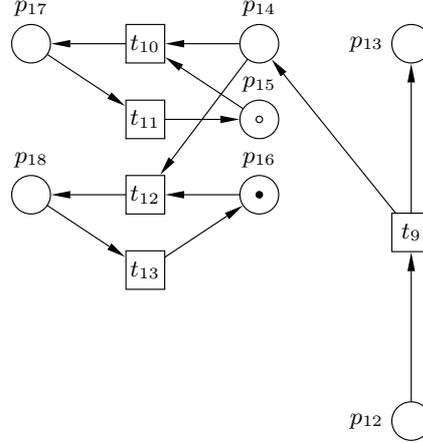


Fig. 8.  $\mathcal{N}_1 \in X_{Root}$  (a),  $\mathcal{N}_2 \in X_{Basic}$  (b) and  $\mathcal{N}_3 \in X_{Basic}$  (c).

added to the set of gluing places after a forward composite net is extended by a basic step net ( $p_f$  of the basic step net) comes with its control place as well.

In the forward composite step nets, there is always a unique control place for every gluing place of  $\mathcal{N} \in X_{Com}^F$ . As a result, the added set of inhibitor arcs (in Definition 7) will be just a singleton set. However, this will change when we introduce the second composition operator allowing transitions to ‘come back’ to some of the existing places.

**An example of forward net composition.** Suppose that we have two nets,  $\mathcal{N}_1 \in X_{Root}$  and  $\mathcal{N}_2 \in X_{Basic}$ , as shown in Figure 8(a, b). We can (forward) composed them by choosing  $(p_1, p'_1)$  as a gluing pair of places and constructing  $\mathcal{N}_1 \oplus^{p_1} \mathcal{N}_2$ . In the process of composing them, a colour-sensitive inhibitor arc will be inserted between place  $p_0$  and transition  $t_1$ . The resulting net will belong to the set  $X_{Com}^F$ . We can extend it further by a step defined by the net  $\mathcal{N}_3 \in X_{Basic}$  (see Figure 8(c)), by gluing places  $p_2$  and  $p'_5$ . The resulting net will contain an inserted colour-sensitive inhibitor arc between  $p_3$  and  $t_4$ . Again the resulting net belongs to the set  $X_{Com}^F$ . Let’s call it  $\mathcal{N}$ . At this point the forward composite step net  $\mathcal{N}$  has a set of gluing points with three elements:  $\{p_1, p_2, p_6\}$ . Any of them can be a starting point of future extensions. Suppose we select  $p_2$  as the



**Fig. 9.**  $\mathcal{N}_4 \in X_{Basic}$ , an example of a basic step net.

next gluing point with a net  $\mathcal{N}_4$  of Figure 9 describing the next step. We will use  $(p_2, p_{12})$  as the next gluing pair.

The resulting net is shown in Figure 10. The two component nets,  $\mathcal{N}$  and  $\mathcal{N}_4$ , can be composed according to Definition 7, and their composition fuses the places of the gluing pair and removes  $p_{12}$ . A new colour-sensitive inhibitor arc is added between the control place  $p_3 = cp^{t_1}(p_2)$  and the step transition  $t_9$ . Note that the set  $\bullet p_2$  has only one element,  $t_1$ , and so only one inhibitor arc is added. Finally,  $p_{13}$ , which is the  $p_f$  place of the net  $\mathcal{N}_4$ , is added to the gluing places of the resulting net.

Having defined the operator for the forward composition of APNs, one can notice that it does not allow a step to return to an already existing place. Therefore, we will introduce another composition operator which will use two gluing pairs of places. Since the basic step net has only one gluing place, we extend its definition.

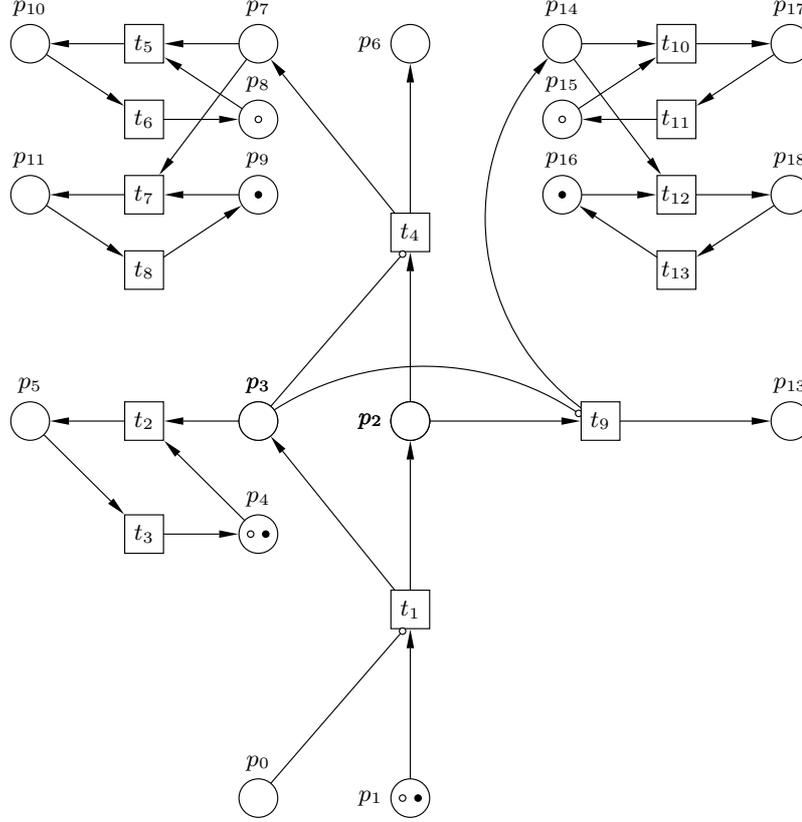
**Definition 9 (one step net).** A one step net is defined as a basic step net, with the only difference, namely its set of gluing places contains both the starting and finishing place:  $G_S = \{p_s, p_f\}$ .

We will denote by  $X_{One}$  the set of all one step nets.

**Definition 10 (backward composition).**

Let  $\mathcal{N}$  be an ambient Petri net, and  $\mathcal{N}_S$  be a one step net, as in Definitions 7 and 9, respectively (the two nets should have disjoint sets of places and transitions). Moreover, let  $g_1, g_2 \in G$  be places satisfying:

$$C(g_1) = C(p_s) = C_{SET}^1, \quad C_S(g_2) = C_S(p_f) = C_{SET}^2$$



**Fig. 10.** A forward composition of a forward composite net  $\mathcal{N}$  and a basic step net  $\mathcal{N}_4$ .

as well as, for all  $c \in C_{SET}^1$  and  $c' \in C_{SET}^2$ ,

$$K(g_1, c) = K_S(p_s, c), \quad K(g_2, c') = K_S(p_f, c').$$

The backward composition of  $\mathcal{N}$  and  $\mathcal{N}_S$  w.r.t. the gluing pairs  $(g_1, p_s)$  and  $(g_2, p_f)$  of places is an ambient Petri net:

$$\mathcal{N} \oplus_{g_2}^{g_1} \mathcal{N}_S = \mathcal{N}' = (P', T', Pre', Post', I', Cl, C', K', M'_0, G')$$

where the different components are defined as follows:

- $P' = P \cup (P_S \setminus \{p_s, p_f\})$  and  $T' = T \cup T_S$ .
- $Pre', Post' : T' \rightarrow \mu P'$  are defined by:

$$Pre'(t)(p) = \begin{cases} Pre(t)(p) & \text{for } t \in T \text{ and } p \in P, \\ Pre_S(t)(p) & \text{for } t \in T_S \text{ and } p \in P_S \setminus \{p_s, p_f\}, \\ Pre_S(t)(p_s) & \text{for } t \in T_S \text{ and } p = g_1, \\ Pre_S(t)(p_f) & \text{for } t \in T_S \text{ and } p = g_2, \\ 0 & \text{otherwise.} \end{cases}$$

and

$$Post'(t)(p) = \begin{cases} Post(t)(p) & \text{for } t \in T \text{ and } p \in P, \\ Post_S(t)(p) & \text{for } t \in T_S \text{ and } p \in P_S \setminus \{p_s, p_f\}, \\ Post_S(t)(p_s) & \text{for } t \in T_S \text{ and } p = g_1, \\ Post_S(t)(p_f) & \text{for } t \in T_S \text{ and } p = g_2, \\ 0 & \text{otherwise.} \end{cases}$$

–  $I' \subseteq T' \times P'$  is defined by:

$$I' = I \cup \{(t_s, cp^t(g_1)) \mid t \in \bullet g_1 \text{ in } \mathcal{N}\} \cup \{(t, cp^{t_s}(p_f)) \mid t \in g_2^\bullet \text{ in } \mathcal{N}'\},$$

where  $t_s \in T_S$  is  $\mathcal{N}_S$ 's unique step transition.

- $C' : P' \cup T' \rightarrow Cl$  is defined by:  $C'|_{P \cup T} = C$  and  $C'|_{(P_S \setminus \{p_s, p_f\}) \cup T_S} = C_S$ .
- $K' \in \mu^\infty \widetilde{P}'$  is defined by:  $K'|_{\bar{p}} = K$  and  $K'|_{P_S \setminus \{p_s, p_f\}} = K_S|_{P_S \setminus \{p_s, p_f\}}$ .
- $M'_0 \in \mu \widetilde{P}'$  is defined by:  $M'_0|_{\bar{p}} = M_0$  and  $M'_0|_{P_S \setminus \{p_s, p_f\}} = M_0^S|_{P_S \setminus \{p_s, p_f\}}$ .
- $G' = G$ ; moreover, the control places of the gluing places in  $G'$  are inherited from  $\mathcal{N}$  except for  $g_2$  that has an additional control place  $cp^{t_s}(g_2)$  which was  $cp^{t_s}(p_f)$  in  $\mathcal{N}_S$ .<sup>6</sup>

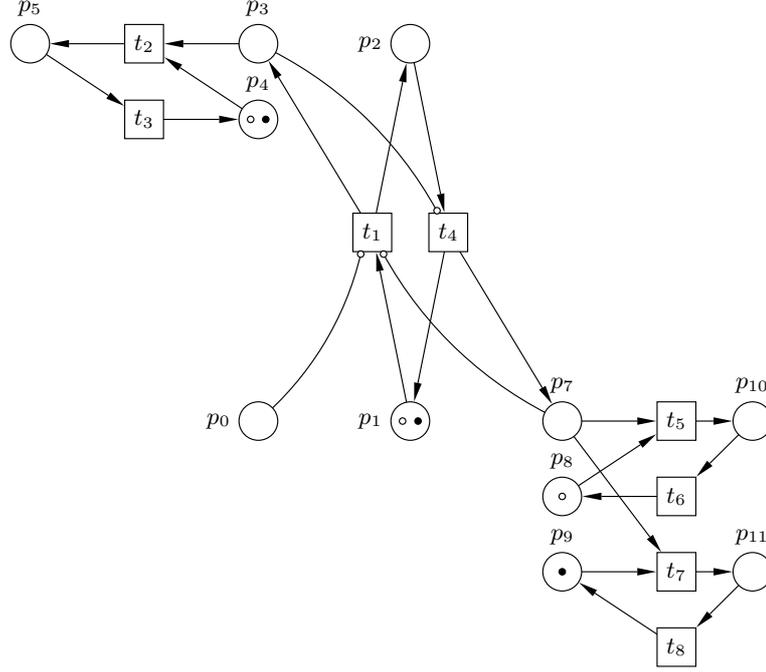
The basic difference between Definitions 7 and 10 is in the generation of the colour-sensitive inhibitor arcs and the set of gluing places in the resulting net. In the backward composition, the set of colour-sensitive inhibitor arcs includes the colour-sensitive inhibitor arcs of  $\mathcal{N}$  together with the two sets of colour-sensitive inhibitor arcs created during the composition. One of the sets is the same as in Definition 7; it contains arcs that are needed to make sure that the user takes instructions (from the system) after executing some transition  $t$  and proceeding to execute  $t_s$ . Now more inhibitor arcs are needed in order for the user to take advices (from the system) after executing transition  $t_s$  and proceeding to the execution of some subsequent transition  $t$ . This is due to the fact that backward composition introduces ‘cycles’ into the resulting net, and we not only need to consider execution sequences containing  $tt_s$ , but as well execution sequences containing  $t_s t$ . The gluing places of  $\mathcal{N}'$  are simply the gluing places of  $\mathcal{N}$ . Note that we do not require that  $g_1$  and  $g_2$  are different places. If they coincide, then both  $p_s$  and  $p_f$  can be glued with the same place ( $g_1 = g_2$ ), creating a self-loop.

In the above definition, as in Definition 7,  $\mathcal{N}$  is a general ambient Petri net. However, we are really interested in nets which can be derived compositionally forming the set  $X_{Com}$  of composite step nets.

**Definition 11 (composite step nets).** *The set of composite step nets  $X_{Com}$  is defined inductively as follows:*

- $X_{Com}^F \subset X_{Com}$ .

<sup>6</sup> So now the user executing some existing transition  $t \in g_2^\bullet$  in  $\mathcal{N}'$  will not be able to proceed without ‘taking the instructions’ from the system following the execution of transition  $t_s$ .



**Fig. 11.** A bidirectional step resulting from the composition of unidirectional steps of Figure 3.

- If  $\mathcal{N} \in X_{Com}$  and  $\mathcal{N}_S \in X_{Basic}$ , then  $\mathcal{N} \oplus^g \mathcal{N}_S \in X_{Com}$ , provided that the forward composition is well-defined.
- If  $\mathcal{N} \in X_{Com}$  and  $\mathcal{N}_S \in X_{One}$ , then  $\mathcal{N} \oplus_{g_2}^{g_1} \mathcal{N}_S \in X_{Com}$ , provided that the backward composition is well-defined.

**Bidirectional Step.** The backward composition of nets enables the creation of *bidirectional steps*, which are composed of two unidirectional steps.

A bidirectional step allows the user to retrieve the previous state by undoing the last action (note that a unidirectional step does not allow the retrieval of previous states). The construction of bidirectional steps turned out to be necessary for the modelling of ambient systems. The simplest way to create a bidirectional step is by gluing two unidirectional steps w.r.t. the places of their step systems (see Definition 5).

An example of a bidirectional step is given in Figure 11. The net of this figure is generated by following the definition of backward composition for the forward composite net  $\mathcal{N} = \mathcal{N}_1 \oplus^{p_1} \mathcal{N}_2$  (see Figure 8 for nets  $\mathcal{N}_1$  and  $\mathcal{N}_2$ ) and  $\mathcal{N}_3$  from Figure 8 treated as a one step net. For the composition of these two nets,  $(g_1, p_s) = (p_2, p'_5)$  and  $(g_2, p_f) = (p_1, p_6)$ <sup>7</sup> were chosen as the two gluing

<sup>7</sup>  $p_s$  and  $p_f$  are the starting and finishing places of  $\mathcal{N}_3$ .

pairs  $(\mathcal{N} \oplus_{p_1}^{p_2} \mathcal{N}_3 = \mathcal{N}')$ . In this case, the places that are removed during the composition of the two nets are  $p'_5$  and  $p_6$ , respectively. Furthermore, the set of colour-sensitive inhibitor arcs of the resulting net consists of the colour-sensitive inhibitor arcs that connect the transitions  $t_4$  and  $t_1$  with the places  $p_3$  and  $p_7$ , respectively, together with the colour-sensitive inhibitor arc (between  $t_1$  and  $p_0$ ) that was added after  $\mathcal{N}_1$  and  $\mathcal{N}_2$  were composed. Indeed, knowing that  $p_f$  is glued with  $g_2$ ,  $p_f = p_6$ ,  $t_s = t_4$ ,  $g_1 = p_2$ ,  $g_2 = p_1$  and  $\bullet p_2 = \{t_1\} = p_1^\bullet$  in  $\mathcal{N}$  (and  $\mathcal{N}'$ ), we obtain, from Definition 10, the following:

$$\begin{aligned} I' &= I \cup \{(t_s, cp^t(g_1)) \mid t \in \bullet g_1 \text{ in } \mathcal{N}\} \cup \{(t, cp^{t_s}(p_f)) \mid t \in g_2^\bullet \text{ in } \mathcal{N}'\} \\ &= \{(t_1, p_0)\} \cup \{(t_4, cp^t(p_2)) \mid t \in \bullet p_2 \text{ in } \mathcal{N}\} \cup \{(t, cp^{t_4}(p_6)) \mid t \in p_1^\bullet \text{ in } \mathcal{N}'\} \\ &= \{(t_1, p_0)\} \cup \{(t_4, cp^{t_1}(p_2))\} \cup \{(t_1, cp^{t_4}(p_6))\} \\ &= \{(t_1, p_0)\} \cup \{(t_4, p_3)\} \cup \{(t_1, p_7)\} \end{aligned}$$

Notice that the place  $p_1$  of the resulting net has two associated control places:  $p_0$  and  $p_7$ . Place  $p_0$  ‘controls’ the movements of the users starting from place  $p_1$  after they arrive there by executing an empty transition  $\epsilon$  ( $p_0 = cp^\epsilon(p_1)$ ). Place  $p_7$ , on the other hand, was a control place of the place  $p_6$  in the net  $\mathcal{N}_3$  ( $cp^{t_4}(p_6)$ ). Now, after composing  $\mathcal{N}$  and  $\mathcal{N}_3$ , where  $p_1$  and  $p_6$  were glued,  $p_7$  ‘controls’ the movements of the users starting from place  $p_1$  after they arrive there by executing transition  $t_4$  ( $p_7 = cp^{t_4}(p_1)$ ). So,  $p_7 = cp^{t_4}(p_6)$  in  $\mathcal{N}_3$ , but  $p_7 = cp^{t_4}(p_1)$  in  $\mathcal{N}'$ .

**Another example of Backward Composition.** We will now present a more general example of the backward composition.

Let  $\mathcal{N}$  be a composite step net obtained by using the forward composition operator on nets  $\mathcal{N}_1$ ,  $\mathcal{N}_2$  and  $\mathcal{N}_3$  from Figure 8:  $\mathcal{N} = (\mathcal{N}_1 \oplus^{p_1} \mathcal{N}_2) \oplus^{p_2} \mathcal{N}_3$ . A net similar to  $\mathcal{N}$  appears in Figure 4 (we only need to add an empty place  $p_0$  and a colour-sensitive inhibitor arc between  $t_1$  and  $p_0$ ). We now backward compose  $\mathcal{N}$  with net  $\mathcal{N}_4$  from Figure 9 treated as one step net, obtaining  $\mathcal{N}'$ . The two chosen gluing pairs for these two nets are  $(g_1, p_s) = (p_2, p_{12})$  and  $(g_2, p_f) = (p_1, p_{13})$ <sup>8</sup>. Applying Definition 10 and gluing the two nets w.r.t. these places results in the net of Figure 12. By carrying out the composition, the places  $p_{12}$  and  $p_{13}$  are removed from the structure of the resulting net as they are fused with the places  $p_2$  and  $p_1$  respectively. The set of transitions of the resulting net consists of the transitions of  $\mathcal{N}$  and  $\mathcal{N}_4$ . We recall that  $p_f$  is glued with  $g_2$  and  $p_f = p_{13}$ . Since  $t_s = t_9$ ,  $g_1 = p_2$ ,  $g_2 = p_1$  and  $\bullet p_2 = \{t_1\} = p_1^\bullet$  in  $\mathcal{N}$  (and  $\mathcal{N}'$ ), the set of colour-sensitive inhibitor arcs of the resulting net can be computed as follows:

$$\begin{aligned} I' &= I \cup \{(t_s, cp^t(g_1)) \mid t \in \bullet g_1 \text{ in } \mathcal{N}\} \cup \{(t, cp^{t_s}(p_f)) \mid t \in g_2^\bullet \text{ in } \mathcal{N}'\} \\ &= I \cup \{(t_9, cp^t(p_2)) \mid t \in \bullet p_2 \text{ in } \mathcal{N}\} \cup \{(t, cp^{t_9}(p_{13})) \mid t \in p_1^\bullet \text{ in } \mathcal{N}'\} \\ &= I \cup \{(t_9, cp^{t_1}(p_2))\} \cup \{(t_1, cp^{t_9}(p_{13}))\} \\ &= I \cup \{(t_9, p_3)\} \cup \{(t_1, p_{14})\}, \end{aligned}$$

<sup>8</sup>  $p_s$  and  $p_f$  are starting and finishing places of  $\mathcal{N}_4$  here.

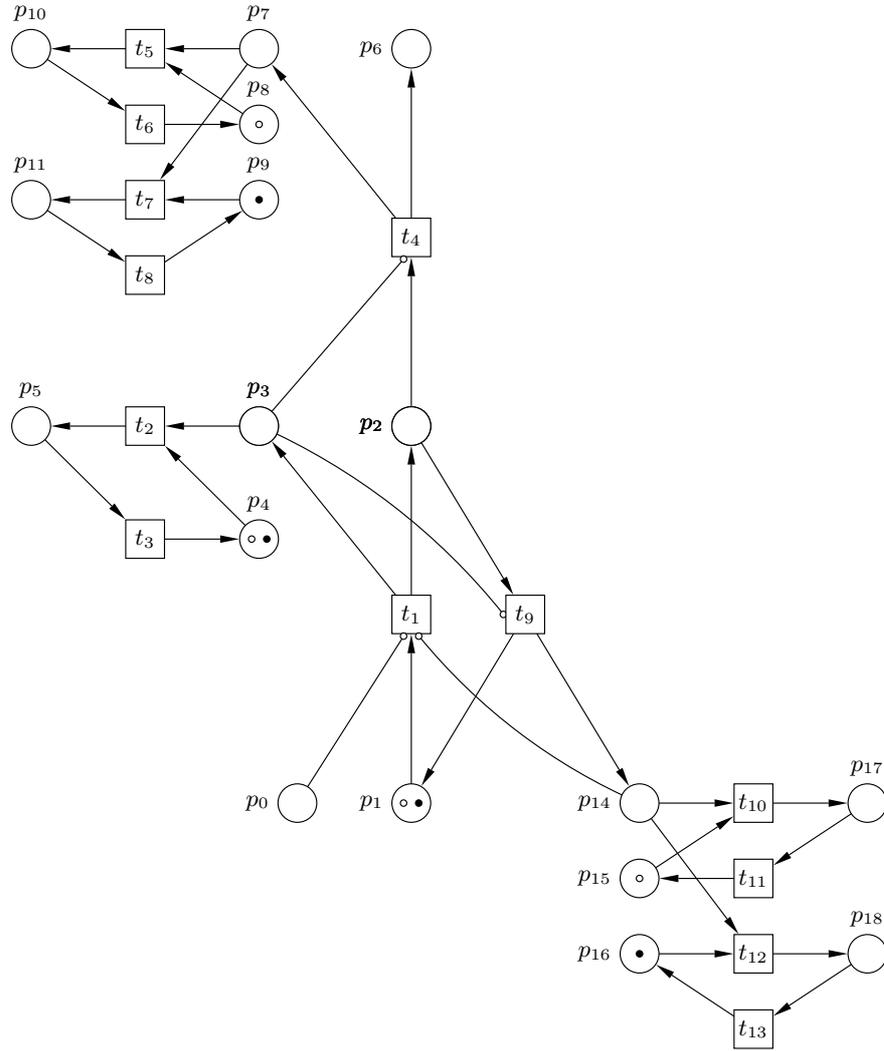


Fig. 12. A composition of a forward composite step net and a one step net.

where  $I = \{(t_1, p_0), (t_4, p_3)\}$ . Note that  $p_{14} = cp^{t_9}(p_{13})$  in  $\mathcal{N}_4$ , but now, in the context of  $\mathcal{N}'$ ,  $p_{14} = cp^{t_9}(p_1)$ .

## 6 Conclusion

Ambient Petri Nets are a class of Petri nets that can provide readable and understandable models of Ambient Systems. The main motivation behind their introduction was to reflect the characteristics of Ambient Systems in an accurate and flexible way.

The application of the new class to the modelling of the case studies (from [9]) led to the creation of the step-modelling approach, which enables the development of models of general Ambient Systems through the composition of specific structural building blocks that are glued together. The composition of the APNs uses two operators, one for the extension of the nets by a step forward and another for the creation of ‘cyclic nets’ or ‘nets with loops’ in the case of going backward to some already existing state. These two operators lead to a flexible compositional approach as the nets can be extended by gluing on every ‘step’ place and not only on specific places (as it happens in other compositional approaches [12]).

As we are really interested in a subset of APNs, namely the composite step nets, we plan in our future work to introduce two notions — *completeness* and *faithfulness* — which would help one to establish that the obtained net model is valid. Completeness would make sure that all the system’s ‘advices’ can be realised by the user, while faithfulness would guarantee that the user’s options are limited to those suggested by the system.

## References

1. Anisimov, N.A., Kovalenko, A.A.: Asynchronous composition of Petri nets via places. In: Proceedings of the Andrei Ershov Second International Memorial Conference on Perspectives of System Informatics, pp. 214-219, (1996)
2. Campos, J., Merseguer, J.: On the Integration of UML and Petri Nets in Software Development. In: Donatelli, S., Thiagarajan, P.S. (eds.) ICATPN 2006. LNCS, vol. 4024, pp. 19-36. Springer, Heidelberg (2006)
3. Desel, J., Reisig, W.: Place/transition Petri Nets. In: Reisig, W., Rozenberg, G. (eds.) Lectures on Petri Nets I: Basic Models. LNCS, vol. 1491, pp. 122-173. Springer, Heidelberg (1998)
4. DiCesare, F., Harhalakis, G., Proth, J.M., Silva, M., Vernadat, F.B.: Practice of Petri Nets in Manufacturing. Chapman and Hall (1993)
5. Girault, C., Valk, R.: Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications. Springer-Verlag (2003)
6. Heiner, M., Gilbert, D., Donaldson, R.: Petri nets for systems and synthetic biology. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 215-264. Springer, Heidelberg (2008)
7. Janicki, R., Koutny, M.: Semantics of inhibitor nets. Information and Computation 123 (1), 1-16 (1995)
8. Jensen, K., Kristensen, L.M.: Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Springer-Verlag (2009)
9. Konios, A.: Ambient Systems and Taxonomy Approaches. Technical Report 1281, School of Computing Science, Newcastle University (2011)
10. Petri, C.A.: Kommunikation mit Automaten. PhD thesis, University of Bonn, West Germany (1962)
11. Reisig, W.: Petri Nets: An Introduction. EATCS Monographs on Theoretical Computer Science. Springer, Heidelberg (1985)
12. Reisig, W.: Simple Composition of Nets. In: Franceschinis, G., Wolf, K. (eds.) PETRI NETS 2009. LNCS, vol. 5606, pp. 23-42. Springer, Heidelberg (2009)

13. Russ, A., Hesse, W., Muller, D.: Ambient information systems - do they open a new quality of IS? In: Proceedings of the 3rd EuroSIGSAND symposium 2008, Marburg/Lahn. LNI 129, pp. 87-101. Koellen-Verlag (2008)
14. Silva, J.L., Ribeiro, O.R., Fernandes, J.M., Campos, J.C., Harrison, M.D.: The APEX framework: prototyping of ubiquitous environments based on Petri nets. In: Bernhaupt, R., Forbrig, P., Gulliksen, J., Lrusdttir, M.K. (eds.) Human-Centred Software Engineering. LNCS, vol. 6409, pp. 6-21. Springer (2010)