COMPUTING SCIENCE

Localities in Systems with a/sync Communication

Jetty Kleijn and Maciej Koutny

TECHNICAL REPORT SERIES

No. CS-TR-1285 October 2011

TECHNICAL REPORT SERIES

No. CS-TR-1285 October, 2011

Localities in Systems with a/sync Communication

J. Kleijn, M. Koutny

Abstract

Localities and a/sync places are two recent extensions to the Petri net model. Whereas localities have been introduced as a modelling tool for membrane systems and more general GALS (globally asynchronous locally synchronous) systems, a/sync places make it possible to model synchronous communication between transitions. We investigate the interaction between locally synchronous execution and synchronous communication. Our focus is in particular on the causalities in the concurrent runs of a new Petri net model combining these features.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

KLEIJN, J,. KOUTNY, M.
Localities in Systems with a/sync Communication
[By] J. Kleijn, M. Koutny
Newcastle upon Tyne: Newcastle University: Computing Science, 2011.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1285)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1285

Abstract

Localities and a/sync places are two recent extensions to the Petri net model. Whereas localities have been introduced as a modelling tool for membrane systems and more general GALS (globally asynchronous locally synchronous) systems, a/sync places make it possible to model synchronous communication between transitions. We investigate the interaction between locally synchronous execution and synchronous communication. Our focus is in particular on the causalities in the concurrent runs of a new Petri net model combining these features.

About the authors

Jetty Kleijn is a visiting fellow within the School of Computing Science, Newcastle University.

Maciej Koutny obtained his MSc (1982) and PhD (1984) from the Warsaw University of Technology. In 1985 he joined the then Computing Laboratory of the University of Newcastle upon Tyne to work as a Research Associate. In 1986 he became a Lecturer in Computing Science at Newcastle, and in 1994 was promoted to an established Readership at Newcastle. In 2000 he became a Professor of Computing Science.

Suggested keywords

CONCURRENCY
PLACE TRANSITION NET
LOCALITY
SYNCHRONOUS AND ASYNCHRONOUS COMMUNICATION
PROCESS SEMANTICS
OCCURRENCE NET
BARB-EVENT
A/SYNC PLACE

Localities in Systems with a/sync Communication

Jetty Kleijn[1] and Maciej Koutny[2]

[1] LIACS, Leiden University
P.O.Box 9512, NL-2300 RA Leiden, The Netherlands
kleijn@liacs.nl

[2] School of Computing Science, Newcastle University
Newcastle upon Tyne, NE1 7RU, United Kingdom
maciej.koutny@ncl.ac.uk

Abstract. Localities and a/sync places are two recent extensions to the Petri net model. Whereas localities have been introduced as a modelling tool for membrane systems and more general GALS (globally asynchronous locally synchronous) systems, a/sync places make it possible to model synchronous communication between transitions. We investigate the interaction between locally synchronous execution and synchronous communication. Our focus is in particular on the causalities in the concurrent runs of a new Petri net model combining these features.
Keywords: concurrency, place transition net, locality, synchronous and asynchronous communication, process semantics, occurrence net, barb-event, a/sync place.

1 Introduction

Petri nets are a well-established framework for the modelling of concurrency in the behaviour of distributed systems. Introduced originally by C.A.Petri [13], they were conceived as a foundational model to describe information flows based on local states and local state changes. Over the years these basic insights have led to a vast area of research with a wide range of theories and applications, methodologies and tools. Nowadays, as reported in collected works [5, 15, 16] edited by J.Desel, W.Reisig and G.Rozenberg, Petri nets provide a framework of net-based models with many extensions and variations introduced for ease of modelling and adapted to the modelling needs of different applications like business process modelling, manufacturing systems, hardware circuits, and biological systems. This not only proves the flexibility of modelling with Petri nets, but also the robustness of the approach.

A key feature of Petri nets are the local transformation rules that support modelling of causality and concurrency in a direct way. The underlying structure of a Petri net is essentially a bipartite directed graph consisting of two types of nodes: *places* and *transitions*. Places are used to carry local information on a system's state whereas transitions represent actions that can occur if certain

local conditions (represented in neighbouring places) are satisfied. When a thus enabled transition occurs ('fires'), it 'consumes' from its input places and 'produces' in its output places. Hence each transition has only a limited and local effect on the (global) state of the system. This basic rule for the dynamics of a Petri net allows one to consider also concurrent and simultaneous occurrences of transitions and induces in a natural way a concept of (in)dependence. This then leads to definitions of step semantics and firing policies describing the allowed steps (simultaneously occurring transitions) at a given state which may depend on features added to the basic structure. All these many possible extensions give rise to different Petri net models each with their own operational semantics.

In [8, 9], J.Kleijn, M.Koutny and G.Rozenberg propose Petri nets as a model to describe what is going on during an evolution of a *membrane system*. Membrane systems, also known as P systems, are inspired by the compartmentisation of living cells and the effect this has on their functioning ([11, 12]). Within each compartment enclosed by a membrane, chemical reactions between molecules take place (modelled as transformations of multisets of objects into new objects). With both models based on multiset calculus, there is a direct connection between membrane systems and Petri nets which is also already obvious in the structure underlying Petri nets: a certain type of object in a specific compartment can be represented as a place, while reaction rules have a natural interpretation as Petri net transitions. To model the compartmentisation of membrane systems, transitions representing reaction rules are associated with specific compartments and so in [8] the concept of *locality* of a transition is proposed as a new extension to Petri nets. The resulting class of *Place/Transition nets with localities* or PTL-nets, is suitable for the modelling of membrane systems and actually also for more general so-called GALS (globally asynchronous, locally synchronous) systems [2, 3]. Such systems exhibit a mix of synchronous and asynchronous behaviour rather than the strict global synchronicity of the original membrane systems which assumes that as many reactions as possible occur in one time unit (of a global clock). The concept of localities allows one to discern subsystems on basis of actions. As a consequence, in PTL-nets locally maximal steps (no transitions belonging to an active locality can be added to the step) can be defined. With this step semantics, PTL-nets provide a faithful, operational model for asynchronous systems consisting of subsystems that are internally fully synchronised (according to a local clock).

To investigate the structure of the concurrent behaviour of PTL-nets, *processes* based on unfoldings of such nets are considered in [8, 9]. Processes characteristically belong to the classical Petri net approach aimed at obtaining a causality semantics for Petri net models (see, e.g., [1, 6, 17]). Processes are labelled occurrence nets, i.e., acyclic Petri nets, providing insight in the explicit, local causalities in a concurrent run of a Petri net. They can be derived from step sequences as representatives of such runs by unfolding the Petri net in accordance with the execution of the step sequence. An important property of such operationally defined process semantics is that it should be *consistent* [6] with the operational causal relations in the original net by satisfying two prop-

erties: (CONS-I) every step sequence executable in a process should correspond to a step sequence of the original net; and (CONS-II) every step sequence of a Petri net should be represented by a step sequence in each process it defines. For PTL-nets operating under the locally maximal step semantics, the standard unfolding strategy does not yield a consistent process semantics (in particular (CONS-I) is not satisfied). To overcome this problem caused by lack of information on potential executability of (co-located) transitions, it is proposed in [8, 9] to employ so-called *barb-events* in the unfolding procedure. Indeed this is then proved to be sufficient for a consistent process semantics.

On basis of different types of dependencies between evolving and communicating subsystems, occurrence nets can be combined into *structured occurrence nets* [10, 14]. In [7] the causality in *communication* structured occurrence nets is investigated and a corresponding Petri net model is defined. This model combines Place/Transition nets with explicit communication and interaction structures called *a/sync places*, to implement the specific causality expressed by communication in structured occurrence nets. These new places incorporate a possibility for both synchronous and asynchronous communication between transitions, meaning that a message sent through such a channel is handed over directly or may be left there to be picked up later. Thus, in particular, a/sync places affect the concept of enabledness of transitions. Using a/sync places as building blocks it is possible to force the synchronous occurrence of transitions. Consequently, a/sync places imply a proper extension to the modelling power of Petri nets which by themselves lack a structural possibility to express that a transition has to wait in order to synchronise with the occurrence of another transition.

In this paper, we further elaborate the idea of structured communication in concurrent systems by combining synchronous execution and synchronous communication in one Petri net model. We do this by lifting the constraints on a/sync communication. Rather than being channels between a pair of transitions (from different subsystems), a/sync places are now shared communication places for multisets of transitions. In addition, we use localities to model synchronously evolving subsystems. All this yields the new class of *Place/Transition nets with localities and a/sync places* or PTLAS-nets.

The combination of synchronous execution and a/sync communication leads to complex causal relationships. To shed light on the intricate interplay of these two phenomena we investigate how to unfold concurrent runs of PTLAS-nets. We use techniques from [7] and [9] to arrive as before in a natural way at occurrence nets with localities and a/sync places. In this case however, it is not sufficient to introduce barb-events in the same way as in [9]. There each barb-event is given a single location and is introduced to supply information about the existence of co-located enabled transitions in order to exclude illegal executions. Here, to cater for possible enforced synchronicity between (transitions from) different localities due to synchronous communication, barb-events need to have multiple localities.

2 PTL-nets with a/sync places

In what follows, we use mostly standard mathematical notation. In addition, let us fix some notation for multisets.

Recall that a multiset over a set X is a function $U : X \to \mathbb{N} = \{0, 1, 2, \ldots\}$. Sets may be treated as multisets and multisets may be represented by listing their elements with repetitions, e.g., $U = \{y, y, z\}$ is a multiset such that $U(y) = 2$, $U(z) = 1$, and $U(x) = 0$ otherwise. The empty multiset $\mathbf{0}$ defined by $\mathbf{0}(x) = 0$, for all $x \in X$, may be written simply as $\varnothing$. For a multiset U (over X), we denote by $supp(U)$ its supporting set consisting of all elements (of X) for which $U(x) > 0$. If $supp(U)$ is finite, then the cardinality of $|U|$ is $U = \sum_{x \in supp(U)} U(x)$. Applying a function ℓ to a (multi)set $Z = \{z_1, \ldots, z_k\}$ of elements of X yields a multiset $\ell(Z) = \{\ell(z_1), \ldots, \ell(z_k)\}$. Then, for a sequence of (multi)sets $Z_1 \ldots Z_n$, we define $\ell(Z_1 \ldots Z_n) = \ell(Z_1) \ldots \ell(Z_n)$.

For two multisets U and V over X, their sum $U \oplus V$ and difference $U \ominus V$ are the multisets given respectively by $U \oplus V(x) = U(x) + V(x)$ and $U \ominus V(x) = max\{0, U(x) - V(x)\}$, for all $x \in X$. We write $U \geq V$ whenever $U(x) \geq V(x)$, for all $x \in X$, and $U > V$ if, in addition, $U \neq V$. The restriction of a multiset U to $Y \subseteq X$ is the multiset $U|_Y$ given by $U|_Y(x) = U(x)$, for all $x \in Y$, and $U|_Y(x) = 0$ otherwise. The multiplication of U by $k \in \mathbb{N}$ is the multiset V given by $V(x) = k \cdot U(x)$, for all $x \in X$.

We now introduce the class of Petri nets with localities and a/sync places extending the standard PT-net model [4].

A PT-*net with localities and a/sync places* (or PTLAS-net) is a tuple:

$$PTLAS = (\widetilde{P}, \widehat{P}, T, W, \ell, M_0) \ , \tag{1}$$

where: $\widetilde{P}$ is a set of *places* and $\widehat{P}$ is a set of *a/sync places* with their union being denoted by P; T is a set of *transitions*; $W : T \times P \cup P \times T \to \mathbb{N}$ is the *arc weight* function; $\ell : T \to \mathbb{N}$ is the *locality* mapping; and M_0 is a multiset over P called the *initial* marking (in general, any multiset over P is a *marking*). It is assumed that the sets $\widetilde{P}$, $\widehat{P}$, and T are finite and mutually disjoint. We refer to the elements of P and T as *nodes* (of $PTLAS$). In diagrams, places are represented by circles; a/sync places by thick circles; transitions by rectangles with their locality displayed in the middle; the arc weight function by directed arcs with the weight n annotated if $n \geq 2$ and arcs with weight 0 are omitted; and a marking by *tokens* (small black dots) drawn inside places.

Figure 1 depicts a PTLAS-net representing a producer, an unbounded a/sync buffer (the middle a/sync place p_0), and two consumers. The producer can execute one of three transitions: m (making item(s)), a (adding two new items to the buffer), and f (failing to add two items, but still adding one item to the buffer). Each of the two consumers represented by the two tokens in place p_3 can cyclically execute: g (getting an item), and u (using the item). Transitions modelling the actions of the producer belong to locality 1, and those representing the actions of the two consumers to locality 2. Initially, the system is in the marking $M_0 = \{p_0, p_1, p_3, p_3\}$.

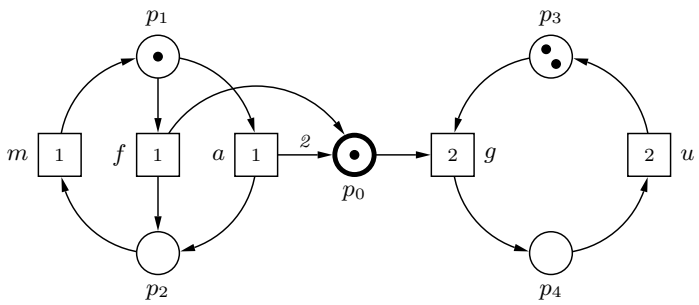

Fig. 1. A PTLAS-net modelling a one-producer/two-consumers system.

For any transition $t \in T$, we define its inputs $\textsc{in}(t)$ and outputs $\textsc{out}(t)$, as multisets over P such that, for every $p \in P$: $\textsc{in}(t)(p) = W(p,t)$ and $\textsc{out}(t)(p) = W(t,p)$. The notions of inputs and outputs are lifted to arbitrary multisets U of transitions in the usual way, for example:

$$\textsc{in}(U) = \bigoplus_{t \in T} U(t) \cdot \textsc{in}(t) \ .$$

We also denote $\widetilde{\textsc{in}}(\alpha) = \textsc{in}(\alpha)|_{\widetilde{P}}$ and $\widehat{\textsc{in}}(\alpha) = \textsc{in}(\alpha)|_{\widehat{P}}$ (and similarly for outputs), for any transition or multiset of transitions α. It is then assumed that the place inputs as well as place outputs of any transition $t \in T$ are non-empty, i.e., $\widetilde{\textsc{in}}(t) \neq \varnothing \neq \widetilde{\textsc{out}}(t)$.

For the PTLAS-net in Figure 1, we have $\widetilde{\textsc{in}}(g) = \{p_3\}$ and $\widehat{\textsc{out}}(a) = \{p_0, p_0\}$, as well as:

$$\begin{aligned} \textsc{in}(\{a,g,g\}) &= \{p_0, p_0, p_1, p_3, p_3\} \\ \textsc{out}(\{a,g,g\}) &= \{p_0, p_0, p_2, p_4, p_4\} \ . \end{aligned} \tag{2}$$

The above notation is extended to (a/sync) places $p \in P$: $\textsc{in}(p)$ and $\textsc{out}(p)$ are multisets of transitions such that respectively $\textsc{in}(p)(t) = W(t,p)$ and $\textsc{out}(p)(t) = W(p,t)$, for every $t \in T$.

The operational behaviour of *PTLAS* is captured by its step sequences, where a *step* U is a multiset of transitions. We begin by defining what it means for such a step to be enabled, considering two such notions. The first (standard) one — called token-enabledness — only checks whether its transitions' input places contain enough tokens (for a/sync input places also counting the tokens that will be supplied synchronously by executing the step). The second one — called lmax-enabledness (or locally maximal enabledness) — also takes into account its transitions' localities by requiring that no extra transitions can be executed in the localities involved in the current step.

Given a marking M of *PTLAS*, a step U of transitions is:

- *token-enabled* at M if $M \oplus \widehat{\textsc{out}}(U) \geq \textsc{in}(U)$.
- *lmax-enabled* at M if U is token-enabled at M, and no step $V > U$ satisfying $\mathit{supp}(\ell(V)) = \mathit{supp}(\ell(U))$ is token-enabled at M.

Whenever U is lmax-enabled at M, we will use $M[U\rangle M'$ to denote that M' is the successor marking resulting from the *execution* of U at M, where:

$$M' = M \oplus \textsc{out}(U) \ominus \textsc{in}(U) \ .$$

For the PTLAS-net in Figure 1 and $U = \{a, g, g\}$, we have:

$$M_0 \oplus \widehat{\textsc{out}}(U) = \{p_0, p_0, p_0, p_1, p_3, p_3\} \geq \{p_0, p_0, p_1, p_3, p_3\} = \textsc{in}(U)$$

and so U is token-enabled at M_0. In fact, U is lmax-enabled at M_0 and we have, by (2), $M_0[U\rangle\{p_0, p_2, p_4, p_4\}$.

The condition for token-enabledness, $M \oplus \widehat{\textsc{out}}(U) \geq \textsc{in}(U)$, and the subsequent definition of the successor marking imply that tokens in a/sync places produced by U can be consumed in the same executed step. However, the same does not hold for the (ordinary) places in $\widetilde{P}$ since we have that $M \geq \widetilde{\textsc{in}}(U)$ as $\widetilde{P}$ and $\widehat{P}$ are disjoint. We further observe that one can replace the condition for token-enabledness by $M \geq \textsc{res}(U)$, where $\textsc{res}(U) = \textsc{in}(U) \ominus \widehat{\textsc{out}}(U)$ captures precisely the tokens needed to make the execution of U possible. As we have seen in the PTLAS-net in Figure 1, the step $U = \{a, g, g\}$ is token-enabled at M_0 which shows that one of the customers will get a completely fresh item via the buffer.

PTLAS-nets extend the PTL-nets of [9] by including a/sync places in addition to the standard ones. Moreover, their semantics is a conservative extension of that given for PTL-nets. There is, however, an important difference in the actual formulation of lmax-enabledness which, in the case of PTL-nets, takes $V > U$ such that $|V| = |U|+1$, i.e., assumes that V is extended by (a single copy of) one transition. Due to the intended behaviour of a/sync places, such an assumption would be too restrictive for PTLAS-nets and $|V| = |U| + 1$ is no longer wanted. Consider, for example, the PTLAS-net in Figure 2(a). The step $U = \{t, z\}$ is not lmax-enabled at M_0 as $V = \{t, z, u, v\}$ is token-enabled and $\mathit{supp}(\ell(V)) = \mathit{supp}(\ell(U)) = \{1, 2\}$. On the other hand, the steps $\{t, z, v\}$, $\{t, z, u\}$, $\{t, t, z\}$ and $\{t, z, z\}$ are not token-enabled at the initial marking. Note that this marking forces a synchronous execution of u and v.

As it was already the case for PTL-nets [9], due to conflicts between transitions coming from different localities, an lmax-enabled step does not necessarily consist of maximal steps w.r.t. the individual localities. In other words, restricting such a step to transitions coming from a single locality may yield a step which fails to be lmax-enabled. In the case of PTLAS-nets this observation can be further strengthened as the latter steps may even fail to be token-enabled. Consider again the PTLAS-net in Figure 2(a) and the step $U = \{t, z, u, v\}$ which is lmax-enabled in the initial marking. By projecting U onto the two localities, 1 and 2, we obtain steps $U' = \{t, u\}$ and $U'' = \{v, z\}$ which are not token-enabled at M_0. Furthermore, the same U can be used to demonstrate that an lmax-enabled step does not have to consist of transitions which are individually token-enabled as $\{u\}$ and $\{v\}$ are not token-enabled at the initial marking. This is in contrast with the step semantics of PTL-nets whose lmax-enabled steps are composed out of token-enabled transitions.

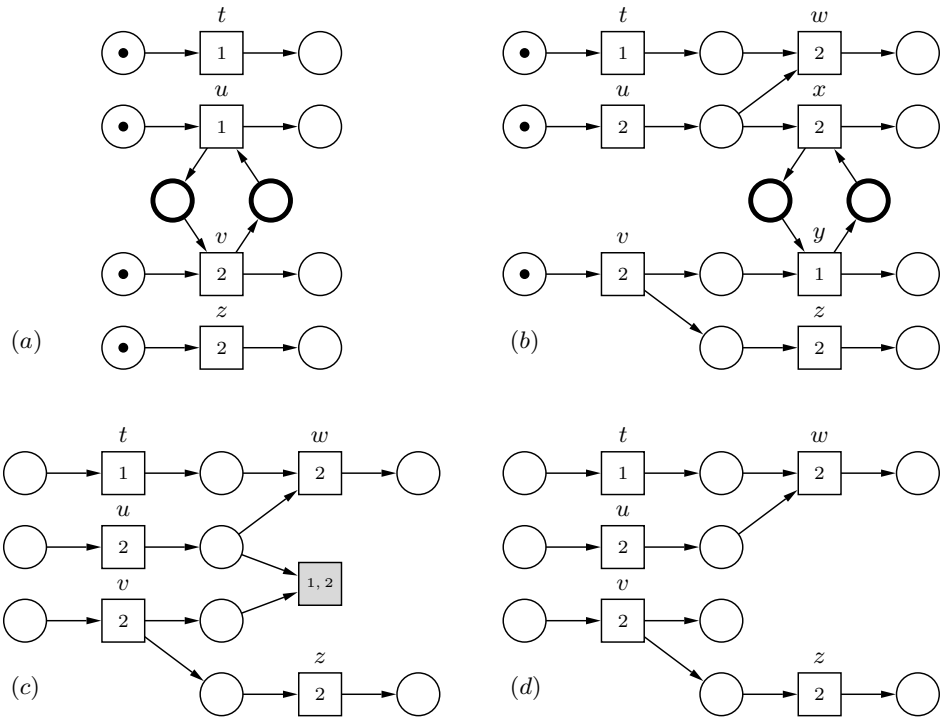

Fig. 2. Two PTLAS-nets (a, b); and two attempts to construct a process of the latter net: with barb-events (c), and without barb-events (d).

Finally, a *step sequence* of $PTLAS$ is a sequence $\sigma = U_1 \dots U_n$ $(n \geq 0)$ of steps such that there are markings $M_1, \dots, M_n$ satisfying $M_0[U_1\rangle M_1, \dots, M_{n-1}[U_n\rangle M_n$. Moreover, $M_0 U_1 M_1 \dots M_{n-1} U_n M_n$ is a *mixed step sequence* of $PTLAS$, and each M_i is a *reachable* marking.

3 Processes of PTLAS-nets

We now show how to construct processes of PTLAS-nets from their step sequences. Each process of the net $PTLAS$ as in (1), will be formalised as an *occurrence net with localities and a/sync conditions (or* OLAS*-net)*:

$$OLAS = (\widetilde{B}, \widehat{B}, E, F, \mathcal{L}, \psi, \mathfrak{E}, \mathfrak{F}, \mathfrak{L})$$

which employs suitably adapted notions from PTLAS-nets, but also includes other components, as described next:

- $\widetilde{B}$ is a set of places (or *conditions*).
- $\widehat{B}$ is a set of a/sync places (or *a/sync conditions*).
- E is a set of transitions (or *events*).

- $F \subseteq B \times E \cup E \times B$, where $B = \widetilde{B} \cup \widehat{B}$, is a *flow relation* (or arc weight function which returns only 0 or 1).
- $\mathcal{L} : E \to \mathbb{N}$ is a locality mapping.
- $\psi : B \cup E \to P \cup T$ is a mapping preserving the types of various kinds of nodes, i.e., $\psi(\widetilde{B}) \subseteq \widetilde{P}$, $\psi(\widehat{B}) \subseteq \widehat{P}$ and $\psi(E) \subseteq T$.
- $\mathfrak{E}$ is a set of *barb-events* (represented by shaded rectangles).
- $\mathfrak{F} \subseteq B \times \mathfrak{E}$ are arrows leading to barb events (defining $\textsc{in}(\mathfrak{e})$, for $\mathfrak{e} \in \mathfrak{E}$).
- $\mathfrak{L} : \mathfrak{E} \to 2^{\mathbb{N}}$ gives non-empty sets of localities (shown inside rectangles representing barb-events).

The default *initial* marking of $OLAS$ is the set M_0^{OLAS} comprising all (a/sync) places without incoming arcs in F.

Markings and execution rules for an OLAS-net as above, are defined as before for PTLAS-nets under the following assumptions. A step U is a multiset of events (and does not include any barb-events), and U is lmax-enabled whenever, in addition to the previously stated requirements, we have that:

- there is no barb-event $\mathfrak{e} \in \mathfrak{E}$ such that $M \oplus \widehat{\textsc{out}}(U) \geq \textsc{in}(U) \oplus \textsc{in}(\mathfrak{e})$ and $\mathfrak{L}(\mathfrak{e}) \subseteq \mathit{supp}(\mathcal{L}(U))$.

The role of barb-events should now be clear. They are not 'real' events, but rather indicators of the enabledness of some transitions. Hence, although they are never executed, they can still influence lmax-enabledness of steps made-up of the standard events.

We will now describe how to construct a process of a PTLAS-net by following the execution of one of its step sequences.

Let $\sigma = U_1 \ldots U_n$ be a step sequence of $PTLAS$ as in (1). An OLAS-net

$$\begin{aligned} OLAS &= (\widetilde{B}, \widehat{B}, E, F, \mathcal{L}, \psi, \mathfrak{E}, \mathfrak{F}, \mathfrak{L}) \\ &= \Big(\bigcup_{j=0}^{n} \widetilde{B}_j, \bigcup_{j=0}^{n} \widehat{B}_j, \bigcup_{j=0}^{n} E_j, \bigcup_{j=0}^{n} F_j, \bigcup_{j=0}^{n} \mathcal{L}_j, \bigcup_{j=0}^{n} \psi_j, \bigcup_{j=0}^{n} \mathfrak{E}_j, \bigcup_{j=0}^{n} \mathfrak{F}_j, \bigcup_{j=0}^{n} \mathfrak{L}_j \Big) \end{aligned} \tag{3}$$

generated by σ is the last net in the sequence $OLAS_0, \ldots, OLAS_n$ where each

$$OLAS_j = (\widetilde{B}_j, \widehat{B}_j, E_j, F_j, \mathcal{L}_j, \psi_j, \mathfrak{E}_j, \mathfrak{F}_j, \mathfrak{L}_j)$$

is constructed as described below, where:

- each non-barb-event node of $OLAS$ is of the form $z = x^i$, where x is a node of $PTLAS$ and $i \in \mathbb{N}$; we also set $\psi(z) = x$ and, if z is an event, $\mathcal{L}(z) = \ell(x)$.
- each barb-event is of the form $\mathfrak{e} = \mathfrak{e}_C^L$, where $L \subseteq \mathbb{N}$ is a non-empty set of localities and $C \subseteq B$ is a non-empty set of (a/sync) conditions; we also set $\textsc{in}(\mathfrak{e}) = C$ and $\mathfrak{L}(\mathfrak{e}) = L$.

Step 0 of the process construction. We set:

$$B_0 = \{p^m \mid p \in P \wedge 1 \leq m \leq M_0(p)\} \quad \text{and} \quad E_0 = F_0 = \varnothing \, .$$

Moreover, $\mathfrak{E}_0$ comprises all barb-events $\mathfrak{e}^L_C$ (where $C \subseteq B_0$) for which there exists a multiset V of transitions of $PTLAS$ satisfying $\textsc{res}(V) = \psi(C)$ and $supp(\ell(V)) = L$.

Step j ($1 \leq j \leq n$) of the process construction. To construct $OLAS_j$, we extend the components of $OLAS_{j-1}$, as follows (below $\triangle x$ denotes the number of nodes of $OLAS_{j-1}$ labelled by $x \in P \cup T$):

$$B_j = B_{j-1} \cup \{p^{m+\triangle p} \mid p \in P \wedge 1 \leq m \leq \textsc{out}(U_j)(p)\}$$
$$E_j = E_{j-1} \cup \{t^{m+\triangle t} \mid t \in T \wedge 1 \leq m \leq U_j(t)\} \ .$$

Then, for every new event $e = t^i \in E_j \backslash E_{j-1}$, we *arbitrarily* choose[1] four sets of (a/sync) conditions: $\widetilde{I}_e \subseteq \widetilde{B}_{j-1}$, $\widehat{I}_e \subseteq \widehat{B}_j$, $\widetilde{O}_e \subseteq \widetilde{B}_j \backslash \widetilde{B}_{j-1}$ and $\widehat{O}_e \subseteq \widehat{B}_j \backslash \widehat{B}_{j-1}$ in such a way that:

$$\psi(\widetilde{I}_e \cup \widehat{I}_e) = \textsc{in}(t) \ \text{ and } \ \psi(\widetilde{O}_e \cup \widehat{O}_e) = \textsc{out}(t) \ . \tag{4}$$

and after setting:

$$F_j = F_{j-1} \ \cup \bigcup_{e \in E_j \backslash E_{j-1}} (I_e \cup \widehat{I}_e) \times \{e\} \ \cup \bigcup_{e \in E_j \backslash E_{j-1}} \{e\} \times (\widetilde{O}_e \cup \widehat{O}_e)$$

it is the case that $|\textsc{in}(b)| \leq 1 \geq |\textsc{out}(b)|$, for every $b \in B_j$.

Finally, the set $\mathfrak{E}_j$ extends $\mathfrak{E}_{j-1}$ by including barb-events $\mathfrak{e}^L_C$ ($C \subseteq B_j$ and $C \not\subseteq B_{j-1}$), for which there exists a multiset V of transitions of $PTLAS$ satisfying $\textsc{res}(V) = \psi(C)$ and $supp(\ell(V)) = L$.

We will provide an example of the above construction later on.

3.1 Properties of the construction

The above construction is a conservative extension of that developed for PT-nets and occurrence nets [4], allowing us to import some useful properties to the current framework. Let $PT = (\widetilde{P}, T, \widetilde{W}, \widetilde{M_0})$, where $\widetilde{W} = W|_{\widetilde{P} \times T \cup T \times \widetilde{P}}$ and $\widetilde{M_0} = M_0|_{\widetilde{P}}$, be the PT-net underlying $PTLAS$ in (1), and $ON = (\widetilde{B}, E, \widetilde{F}, \widetilde{\psi})$, where $\widetilde{F} = F|_{\widetilde{B} \times E \cup E \times \widetilde{B}}$ and $\widetilde{\psi} = \psi|_{\widetilde{B} \cup E}$, be the occurrence net underlying $OLAS$ in (3). The step semantics of PT and ON is essentially that of token-enabled step semantics of PTLAS-nets and OLAS-nets with no a/sync places/conditions and barb-events. It is clear that each step sequence of $PTLAS$ is a step sequence of PT, and the construction described above can be used to derive ON from PT and (the same) σ. Moreover, as shown next, the semantics of $OLAS$ is set-based rather than multiset-based, similarly as that of ON.

Proposition 1. *If $G_0 H_1 G_1 \ldots G_{m-1} H_m G_m$ is a mixed step sequence of OLAS, then $G_0, \ldots, G_m$ are sets, and $H_1, \ldots, H_m$ are mutually disjoint sets.*

[1] This means that, in general, more than one process can be constructed for σ. We will later show that suitable sets $\widetilde{I}_e$, $\widehat{I}_e$, $\widetilde{O}_e$ and $\widehat{O}_e$ can always be found.

Proof It is easily seen that $G_0|_{\widetilde{B}}H_1G_1|_{\widetilde{B}}\ldots G_{m-1}|_{\widetilde{B}}H_mG_m|_{\widetilde{B}}$ is a mixed step sequence of ON which, in particular, means that $H_1,\ldots,H_m$ are mutually disjoint sets. Moreover, by construction, $G_0 = M_0^{OLAS}$ is a set, $\textsc{in}(M_0^{OLAS}) = \varnothing$, and $|\textsc{in}(b)| \leq 1$ for all $b \in B$. Hence $G_1,\ldots,G_m$ are sets. □

Our aim is to show that the process construction we presented is consistent, i.e., it satisfies (CONS-I) and (CONS-II). We first show (CONS-I) which means that every step sequence of the constructed OLAS-net corresponds to a legal step sequence of the original PTLAS-net.

Theorem 1. *If ζ is a (mixed) step sequence of OLAS, then $\psi(\zeta)$ is a (mixed) step sequence of PTLAS.*

Proof We first observe that, by (4), for every event $e \in E$:

$$\psi(\textsc{in}(e)) = \textsc{in}(\psi(e)) \;\text{ and }\; \psi(\textsc{out}(e)) = \textsc{out}(\psi(e))\;. \tag{5}$$

We only need to prove that the result is satisfied for a mixed step sequence $\zeta = G_0H_1G_1\ldots G_{m-1}H_mG_m$. We proceed by induction on m. For $m = 0$ the result follows from the definition of $OLAS_0$.

In the induction step, we assume $\zeta' = \zeta HG$ is a mixed step sequence of $OLAS$ and $\psi(\zeta)$ is a mixed step sequence of $PTLAS$. Let $U = \psi(H)$, $M = \psi(G_m)$ and $K = \psi(G)$. As H is token-enabled at G_m, $G_m \geq \textsc{in}(H) \ominus \widehat{\textsc{out}}(H)$. Hence:

$$\begin{aligned} M = \psi(G_m) &\geq \psi(\textsc{in}(H) \ominus \widehat{\textsc{out}}(H)) \\ &= \psi(\textsc{in}(H)) \ominus \psi(\widehat{\textsc{out}}(H)) \qquad \text{(by (5))} \\ &= \textsc{in}(\psi(H)) \ominus \widehat{\textsc{out}}(\psi(H)) = \textsc{in}(U) \ominus \widehat{\textsc{out}}(U)\;. \end{aligned}$$

Thus U is token-enabled at M. If U is not lmax-enabled at M, then there is a step $U \oplus V > U$ which is token-enabled at M and $\mathit{supp}(\ell(V)) = \mathit{supp}(\ell(U))$. From the fact that both U and $U \oplus V$ are token-enabled at M it follows that $M \oplus \widehat{\textsc{out}}(U) \ominus \widehat{\textsc{in}}(U) \geq \textsc{res}(V)$. Hence

$$\psi(G_m \oplus \widehat{\textsc{out}}(H) \ominus \widehat{\textsc{in}}(H)) \geq \textsc{res}(V)\;.$$

Consequently, as G_m is a set by Proposition 1, there is a set

$$C \subseteq G_m \oplus \widehat{\textsc{out}}(H) \ominus \widehat{\textsc{in}}(H)$$

such that $\psi(C) = \textsc{res}(V)$. Hence, by (5) and the construction, there is a barb-event $\mathfrak{e} = \mathfrak{e}_C^L$ such that $L = \mathit{supp}(\ell(V))$ and $C \oplus \widehat{\textsc{out}}(H) \geq \textsc{in}(H) \oplus \textsc{in}(\mathfrak{e})$. This produces a contradiction with H being lmax-enabled at G_m. Hence U is lmax-enabled at M.

We then observe that since $G = G_m \oplus \textsc{out}(H) \ominus \textsc{in}(H)$, we have:

$$\begin{aligned} K = \psi(G) &= \psi(G_m \oplus \textsc{out}(H) \ominus \textsc{in}(H)) \\ &= \psi(G_m) \oplus \psi(\textsc{out}(H)) \ominus \psi(\textsc{in}(H)) \qquad \text{(by (5))} \\ &= \psi(G_m) \oplus \textsc{out}(\psi(H)) \ominus \textsc{in}(\psi(H)) \\ &= M \oplus \textsc{out}(U) \ominus \textsc{in}(U)\;. \end{aligned}$$

Hence $M[U\rangle K$, and so $\psi(\zeta')$ is a mixed step sequence of $PTLAS$. □

We then show that (CONS-II) holds, i.e., that the successively constructed sets of events form a legal step sequence of $OLAS$ corresponding to σ.

Theorem 2. *Let* $D_1 = E_1, D_2 = E_2 \backslash E_1, \ldots, D_n = E_n \backslash E_{n-1}$. *Moreover, let*

$$C_j = \{b \in B_j \mid \textsc{out}(b) = \varnothing \text{ in } OLAS_j\} ,$$

for $0 \leq j \leq n$. *Then* $\zeta = C_0 D_1 C_1 \ldots C_{n-1} D_n C_n$ *is a mixed step sequence of OLAS.*

Proof Let $M_0 U_1 M_1 \ldots M_{n-1} U_n M_n$ be the mixed step sequence of *PTLAS* corresponding to the step sequence σ. We will prove, by induction on j, that $\zeta_j = C_0 D_1 C_1 \ldots C_{j-1} D_j C_j$ is a mixed step sequence of *OLAS* such that we have $\psi(C_0) = M_0, \ldots, \psi(C_j) = M_j$.

In the base case ($j = 0$), we have $C_0 = B_0 = M_0^{OLAS}$ (any (a/sync) condition added after the initial step has an incoming arc) and $\psi(C_0) = M_0$ directly by construction (note that $\textsc{in}(b) = \varnothing$, for all $b \in B_0$).

In the induction step, we start by showing that D_j is lmax-enabled at C_{j-1}. First, we observe that in *PTLAS* we have $M_{j-1} \oplus \widehat{\textsc{out}}(U_j) \geq \textsc{in}(U_j)$. Moreover, by the induction hypothesis, $\psi(C_{j-1}) = M_{j-1}$.

> *Hence it is possible to find the sets* $\widetilde{I}_e$, $\widehat{I}_e$, $\widetilde{O}_e$ *and* $\widehat{O}_e$, *as required by the process construction, which demonstrates its well-definedness.*

Furthermore, the choice of the sets must be such that $C_{j-1} \cup (\widehat{B}_j \setminus \widehat{B}_{j-1}) \geq \widetilde{\textsc{in}}(D_j)$. It therefore follows that D_j is token-enabled at C_{j-1}. If D_j is not lmax-enabled at C_{j-1}, then one of the following holds:

Case 1: There is a multiset $D \neq \varnothing$ over E such that $\mathit{supp}(\mathcal{L}(D)) \subseteq \mathit{supp}(\mathcal{L}(D_j))$ and $H = D_j \oplus D$ is token-enabled at C_{j-1}.
Then $M \oplus \widehat{\textsc{out}}(U) \geq \textsc{in}(U) \oplus \textsc{in}(\mathfrak{e})$. Thus $C_{j-1} \oplus \widehat{\textsc{out}}(H) \geq \textsc{in}(H)$. Hence, by (5), $\psi(C_{j-1}) \oplus \widehat{\textsc{out}}(\psi(H)) \geq \textsc{in}(\psi(H))$. Thus $M_{j-1} \oplus \widehat{\textsc{out}}(H) \geq \textsc{in}(H)$. Moreover, $\mathit{supp}(\ell(H)) = \mathit{supp}(\ell(D_j))$. Hence U_j is not lmax-enabled at M_{j-1} in *PTLAS*, a contradiction.

Case 2: There is a barb-event $\mathfrak{e} = \mathfrak{e}_C^L \in \mathfrak{E}$ such that $C_{j-1} \oplus \widehat{\textsc{out}}(D_j) \geq \textsc{in}(D_j) \oplus C$ and $L \subseteq \mathit{supp}(\mathcal{L}(U))$.
Let V be a multiset over T from which $\mathfrak{e}_C^L$ has been derived. By proceeding similarly as in Case 1, we can show that $U_j \oplus V$ is token-enabled at M_{j-1} and $\mathit{supp}(\ell(U_j)) = \mathit{supp}(\ell(U_j \oplus V))$, contradicting U_j being lmax-enabled at M_{j-1}. Hence D_j is lmax-enabled at C_{j-1}, and one can see that $C_{j-1}[D_j\rangle C_{j-1}$ as well as $\psi(C_j) = M_j$ which follows from $\psi(C_{j-1}) = M_{j-1}$ and (5). □

3.2 Removing redundant barb-events

It is relatively straightforward to simplify the constructed process without invalidating the results we have just presented. A barb-event $\mathfrak{e}_C^L$ is *redundant* if one of the following holds:

- there is a barb-event $\mathfrak{e}_{C'}^{L'} \neq \mathfrak{e}_C^L$ such that $C' \subseteq C$ and $L' \subseteq L$.
- there is a multiset H over E such that $C = \textsc{res}(H)$ and $L = \mathit{supp}(\mathcal{L}(H))$.
- there are $b_1, \ldots, b_m \in B$ ($m \geq 2$) such that $(b_i, b_{i+1}) \in F \circ F$ (for $i < m$) and $b_1, b_m \in C$ and $\{b_1, \ldots, b_m\} \cap \widetilde{B} \neq \varnothing$.

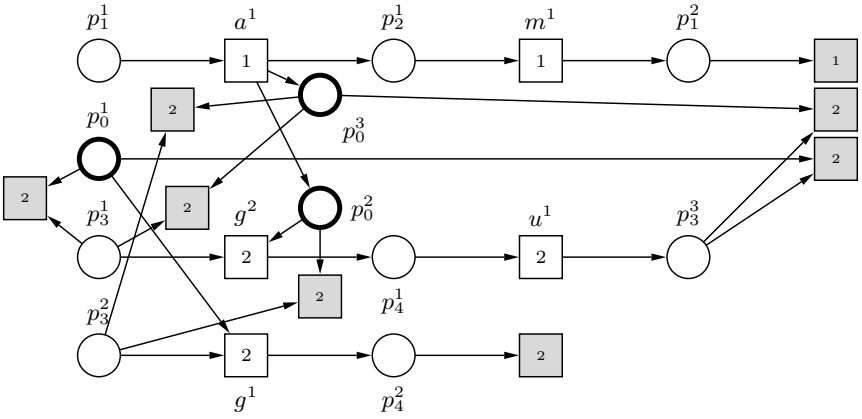

Fig. 3. Process *OLAS* generated for the PTLAS-net in Figure 1 and its step sequence $\sigma = \{a, g, g\}\{m, u\}$ after removing redundant barb-events.

Intuitively, the first two cases exclude barb-events whose 'enabledness' implies enabledness of a smaller barb-event or enabledness of a step of events which would require exactly the same tokens and localities as the redundant barb-event. The third case excludes barb-events which can never be enabled at a marking reachable from the default initial marking of *OLAS*. It therefore follows that Theorems 1 and 2 still hold if we remove all the redundant barb-events together with the adjacent arcs.

Figure 3 shows a process generated for the PTLAS-net in Figure 1 and its step sequence $\sigma = \{a, g, g\}\{m, u\}$ after removing all the redundant barb-events. In particular, the redundant barb-events in $\mathfrak{E}_0$ were: $\mathfrak{e}^{\{2\}}_{\{p_0^1, p_3^2\}}$ on account of g^1; $\mathfrak{e}^{\{1\}}_{\{p_1^1\}}$ on account of a^1; and $\mathfrak{e}^{\{1,2\}}_{\{p_1^1, p_3^1\}}$, $\mathfrak{e}^{\{1,2\}}_{\{p_1^1, p_3^2\}}$ and $\mathfrak{e}^{\{1,2\}}_{\{p_0^1, p_1^1, p_3^1, p_3^2\}}$ on account of $\mathfrak{e}^{\{1\}}_{\{p_1^1\}}$. Moreover, $\mathfrak{e}^{\{2\}}_{\{p_0^2, p_3^3\}}$ belonging to $\mathfrak{E}_2 \backslash \mathfrak{E}_1$ has been removed as $p_4^1 \in \widetilde{B}$ and $(p_0^2, g^2), (g^2, p_4^1), (p_4^1, u^1), (u^1, p_3^3) \in F$.

Omitting (all) barb-events from the process construction would invalidate their consistency. Consider, for example the PTLAS-net in Figure 2(b) and its step sequence $\sigma = \{t, u, v\}\{w, z\}$. Figure 2($c$) depicts its process *OLAS* after removing all redundant barb-events. We then note that deleting the only barb-event of *OLAS* leads to the net in Figure 2(d) which can execute a step sequence σ such that $\psi(\sigma) = \{u, v\}\{t, z\}\{w\}$. Thus Theorem 1 is violated since $\{u, v\}\{t, z\}\{w\}$ is not a legal step sequence of the PTLAS-net in Figure 2(b) (as $\{t, z, x, y\}$ is token-enabled at the marking reached after executing $\{u, v\}$).

In the process construction developed for PTL-nets in [9], each barb-event has a single locality associated with it. Such an approach would not work here. Consider, for example, the PTLAS-net in Figure 4(a) and its process corresponding to step sequence $\sigma = \{t, u\}$ shown in Figure 4(b). If, in the process construction, the two-localities barb-event was replaced by two 'equivalent' single-locality barb-events, then the resulting net shown in Figure 4(c) would fail to satisfy The-

orem 2. In fact, this net does not allow the event labelled by t to be executed in any reachable marking.

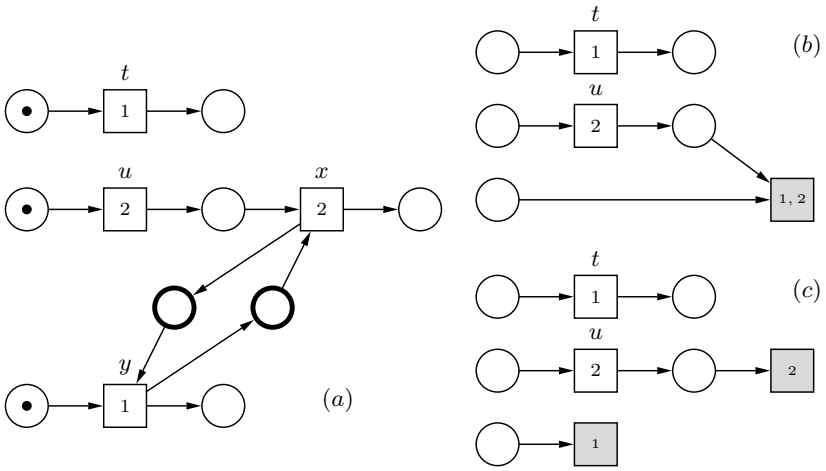

Fig. 4. A PTLAS-net (a); its (simplified) process constructed for $\sigma = \{t, u\}$; and an attempt to derive a process with barb-events based on single localities (c).

4 Conclusions

We have presented a process construction for PTLAS-nets which yields nets satisfying the consistency criteria of [6]. In our future work we plan to extend this construction to include also inhibitor and activator arcs which are of practical relevance when modelling, e.g., biological systems.

Acknowledgement We would like to thank Grzegorz Rozenberg for introducing us to the area of Natural Computing and his continuous encouragement and support of our research over the years.

References

1. E.Best and R.Devillers: Sequential and Concurrent Behaviour in Petri Net Theory. Theoretical Computer Science **55** (1988) 87–136
2. L.P.Carloni and A.L.Sangiovanni-Vincentelli: A Formal Modelling Framework for Deploying Synchronous Designs on Distributed Architectures. In Proc. of the First International Workshop on Formal Methods for Globally Asynchronous Locally Synchronous Architectures (2003)
3. J.Cortadella, M.Kishinevsky, A.Kondratyev, L.Lavagno and A.Yakovlev: *Logic Synthesis of Asynchronous Controllers and Interfaces*. Springer Series in Advanced Microelectronics, Springer-Verlag (2002)

4. J.Desel and W.Reisig: Place/Transition Petri Nets. Lecture Notes in Computer Science **1491** (1998) 122–173
5. J.Desel, W.Reisig and G.Rozenberg (eds.): Lectures on Concurrency and Petri Nets. Lecture Notes in Computer Science **3098** (2004)
6. H.C.M.Kleijn and M.Koutny: Process Semantics of General Inhibitor Nets. Information and Computation **190** (2004) 18-69
7. J.Kleijn and M.Koutny: Causality in Structured Occurrence Nets. Lecture Notes in Computer Science **6875** (2011) 283-297
8. J.Kleijn, M.Koutny and G.Rozenberg: Towards a Petri Net Semantics for Membrane Systems. Lecture Notes in Computer Science **3850** (2006) 292–309
9. J.Kleijn, M.Koutny and G.Rozenberg: Process Semantics for Membrane Systems. J. of Automata, Languages and Combinatorics **11** (2006) 321-340
10. M.Koutny and B.Randell: Structured Occurrence Nets: A Formalism for Aiding System Failure Prevention and Analysis Techniques. Fundamenta Informaticae **97** (2009) 41–91
11. Gh.Păun: *Membrane Computing. An Introduction*. Springer Verlag (2002)
12. Gh.Păun, G.Rozenberg and A.Salomaa (eds.): *The Oxford Handbook of Membrane Computing*. Oxford University Press (2009)
13. C.A. Petri: *Kommunikation mit Automaten*. Ph.D. Thesis (1962)
14. B.Randell: Occurrence Nets Then and Now: The Path to Structured Occurrence Nets. Lecture Notes in Computer Science **6709** (2011) 1–16
15. W.Reisig and G.Rozenberg (eds.): Lectures on Petri Nets I: Basic Models. Lecture Notes in Computer Science **1491** (1998)
16. W.Reisig and G.Rozenberg (eds.): Lectures on Petri Nets II: Applications. Lecture Notes in Computer Science **1492** (1998)
17. G.Rozenberg and J.Engelfriet: Elementary Net Systems. Lecture Notes in Computer Science **1491** (1998) 12–121