

Local Nets

Daniel Yankelevich

Univ. Pisa, Dip. Informatica and HP Labs, Pisa Science Center

Pisa - ITALIA

E-mail: dany@pisa3.italy.hp.com

Abstract

Petri nets have been widely used to model and specify concurrent, distributed systems. However, they give no mean to reflect spatial distribution of resources.

In this work, an extension of P/T Petri nets is introduced in order to capture the concept of spatial distribution, specifying not only places and events but also how places are distributed in sites.

These new nets, called Local nets, are a very simple extension which is shown to be semantics respecting, and to support a methodology for the definition of systems.

Some theoretical results are given in order to establish the basis for a practical use of these nets for specification and verification, and the relation with previous existing models is also discussed.

Keywords: Petri nets, concurrent systems, specifications.

1 Introduction

Petri nets [9] have been widely used to model concurrent systems. In particular, P/T Petri nets conform a very useful tool for specification and verification of systems composed of interacting entities, since they have a nice graphical representation and present algebraic properties which allow the use of algebraic techniques for verification [11].

Petri nets have also been used to specify and analyze complex processes, not only in computer science; for example they have been used to help in the definition of big projects. They express directly the concept of *causality* and reflect the causal relationships between events. However, they do not give an idea of *distribution in space* of entities. In particular, no information on *where* an event takes place can be specified with a Petri net. Causality and distribution are strongly related, but they are not the same concept.

For example, the net of Figure 1 could be given to represent the following process:

Process One

- Two sites of a company, one in Bristol and one in Pisa, start a common development. The one in Bristol specifies one module (event sB) and that in Pisa specifies a different one (event sP).
- When both sites have finished, they exchange the specifications, in order to have a view of the whole project (event ex).
- Each site proceeds to implement its own part (events iP and iB).

In the net of Figure 1, no information on where the events take place, or of how the places are grouped together (in Pisa or in Bristol) is given. The same net specifies also the (very different) process in which Pisa implements the module that Bristol has specified and viceversa.

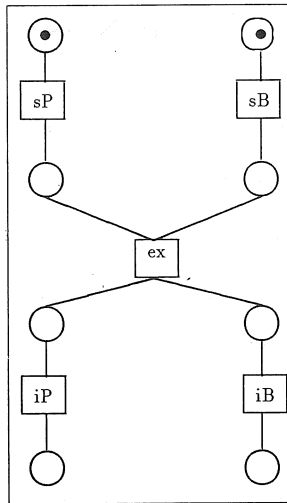


Figure 1: The net of Process One

When information about distribution is taken into account some processes that are identified using Petri nets have to be discriminated. Hence, some additional information has to be included in the specifications.

One solution could be the use of *high level* Petri nets [10]. However, for simple examples as the one given above, this seems to be an ad-hoc unsatisfactory solution: it requires to move from a propositional to a first order theory without a real need for first order features. Moreover, not all such nets have nice properties as a simple graphical representation.

On the other hand, the information about distribution should be orthogonal or independent of the underlying net, and it should not be codified inside the net in the form of colored tokens or predicates associated to events. In fact, there are three requirements for such an extension:

1. the extension should be semantics respecting, that is, if one associates some information about distribution and then (s)he forgets it, the resulting net should be equivalent to the one without extra information,
2. the extension should be orthogonal, since distribution is a different concept, and it has to allow one to specify or leave unspecified the spatial distribution, and
3. it has to be flexible enough to permit the definition of subsites of a given site (for example, as in a *fork* operation or when a team is divided to work in different subprojects), to have multiple inheritance of sites (for example, actions which involves more than one place) and sites to be destroyed or created.

Most of the existing models do not satisfy requirement (1) or (2). A simple idea would be “to group places together” in sets of places belonging to some common site, but it is easy to see that this idea do not satisfy the flexibility requirements of (3).

The problem of localities has been studied in the context of process description languages [7] by G. Boudol, I. Castellani, M. Hennesy and A. Kiehn [2]. In this work, the ideas of [2] are applied to the field of nets, defining Local nets as an extension of P/T Petri nets which satisfies the requirements stated above.

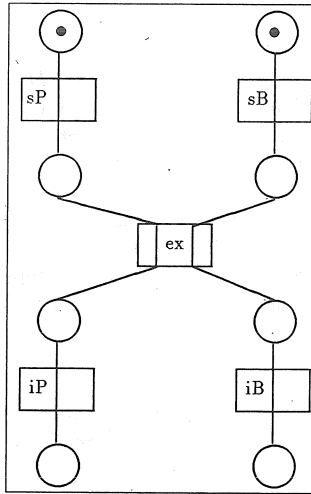


Figure 2: The local net of Process One

For example, the net of figure 1 can be extended with spatial information represented graphically as arcs inside the events to specify correctly the Process One. The extended net is shown in figure 2.

In this net, it is shown that the arc from Pisa causing the event *ex* continues with the line that causes the event *iP*. Notice that this information is not the same as having individual tokens. If the extra arcs of the net are deleted (turning transparent boxes into opaque ones), the net of figure 1 is obtained.

With this extension, the same properties of nets are still valid and the same verification mechanisms can be used. Moreover, extending existing procedures to take into account the spatial distribution is easy.

Section 2 recalls the main definitions of P/T Petri nets. Section 3 introduces local nets, and include some definitions and theorems to support the use of these nets for specification.

2 Background: P/T Petri Nets and Boolean Algebras

In this section, the main definitions of P/T nets and of lattices are recalled. For a complete presentation of Petri nets, the reader can refer to [9]. For the definitions about algebra, the reader can refer to [3].

Definition 2.1 P/T Net

A net $N = (S, T, W, M_0)$ consists of disjoint sets S and T of places and transitions, the weight function $W : S \times T \cup T \times S \rightarrow \mathbb{N}$ and the initial marking $M_0 : S \rightarrow \mathbb{N}$, where \mathbb{N} are the natural numbers.

Definition 2.2 Pre and post sets

Given a net $N = (S, T, W, M_0)$ and an element $x \in S \cup T$,

- the pre-set of x is ${}^*x = \{y \in S \cup T \mid W(y, x) \neq 0\}$

- the post-set of x is $x^\circ = \{y \in S \cup T \mid W(x, y) \neq 0\}$

Definition 2.3 Markings and enabling

Given a net $N = (S, T, W, M_0)$, a marking is a function $M : S \rightarrow \mathbb{N}$. A transition $t \in T$ is enabled under a marking M , denoted $M[t >$ if for all $s \in S$, we have $M(s) \geq W(s, t)$. An enabled transition may occur, producing a follower marking M' , written $M[t > M'$, if $M[t >$ and $M'(s) = M(s) - W(s, t) + W(t, s)$ for all $s \in S$.

The previous definition and notation is generalized for multisets of transitions G , and we write $M[G > M'$. In this case, $M[G > M'$ is called a step.

A marking M is called reachable if for some sequence of steps $G_1 \dots G_n$ we have that $M_0[G_1 > M_1 \dots [G_n > M$, and it is called safe if for all $s \in S$, it holds that $M(s) \leq 1$. A net is called safe if all its reachable markings are safe.

Definition 2.4 Processes

The tuple $\Pi = (B, E, F, p)$ is said to be a process of a given safe net N if $\Pi = (B, E, F)$ is a safe net with no initial marking and p is a function from $B \cup E$ to $S \cup T$, such that the following conditions hold:

- the graph defined by $(B \cup E, F)$ is acyclic.
- for all $b \in B$ we have $|^\circ b| \leq 1$ and $|b^\circ| \leq 1$
- p maps elements of B to elements of S and elements of E to elements of T and the initial places¹ of Π to the places e such that $M_0(e) = 1$
- for each $e \in E$ and for all $s \in S$, $W(s, p(e)) = |p^{-1}(s) \cap e|$ and $W(p(e), s) = |p^{-1}(s) \cap e^\circ|$

From the definition of a process of a safe net, it is possible to construct a partial order of events, representing the causal relations of occurrences of transitions in the process. Hence, to a computation of the net it is associated a partial order in the following way:

Definition 2.5 From processes to partial orders

Let $\Pi = (B, E, F, p)$ be a process of a safe net N . We associate to it the partial order $\langle E, \leq \rangle$, where $e \leq e'$ iff $(e, e') \in F^*$, where F^* denotes the reflexive and transitive closure of F .

In this order, $e \leq e'$ means that e is a necessary condition for e' to occur.

Notation 2.1 Let $L = \langle D, \leq \rangle$ be a non-empty ordered set. We write $x \wedge y$ for the least upper bound of $\{x, y\}$ and $x \vee y$ for its greatest lower bound. Similarly, if $S \subseteq D$, we write $\vee S$ for its least upper bound and $\wedge S$ for its greatest lower bound.

Definition 2.6 Let $L = \langle D, \leq \rangle$ be a non-empty ordered set. The structure L is a complete lattice iff $\vee S$ and $\wedge S$ exist for all $S \subseteq D$.

Definition 2.7 A lattice L is said to be distributive if it satisfies:

$$(\forall a, b, c \in L) a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

Definition 2.8 A boolean algebra is a structure $(B, \vee, \wedge, ', 0, 1)$ such that (B, \vee, \wedge) is a distributive lattice, $a \vee 0 = a = a \wedge 1$, $a \in B$, $a \vee a' = 1$ and $a \wedge a' = 0$ for all $a \in B$

¹The initial places b of a process are the places such that $^\circ b = \emptyset$.

3 Local Nets

In this section the proposed extension of Petri nets, namely Local nets, are introduced, and some results about them are presented.

Definition 3.1 Local Net

A Local net $L = (S, T, W, M_0, R)$ is such that (S, T, W, M_0) is a net and $R \subseteq (S \times T \times S)$ is the local relation such that $(s, t, s') \in R$ implies $W(s, t) \geq 1$ and $W(t', s') \geq 1$.

The local relation R is the relation that tells us which places are in the same location. That is, $(s, t, s') \in R$ can be interpreted as "when event t occurs s' is at the same physical place of s ".

One interesting point of Local nets is that the graphical representation of Petri nets is not lost: the relation R can be drawn as arcs inside the boxes of the net.

Many examples of Petri nets specifications describe intuitively the system as composed of components: a process and a buffer, a producer and a consumer, readers, writers and an arbiter. However, from the description given in the net there is no way of recovering these agents, which play an important role in the intuitive understanding. Moreover, when the examples become more complex, it is difficult even to recognize in the graphical representation the different agents that interact in the system. From the Local net description, it is easy to individuate the different agents. In a previous work [4] the relation between automata describing individual behaviours and nets describing the whole system has been studied. The notion of local behaviours describing different subprocesses of a global system has been used previously without a formal basis. See for instance [1], where partial orders of events are classified in threads of control in order to introduce some concepts of parallel programming.

The second requirement discussed in the introduction was not to lose the properties we already have for P/T nets. In fact, it is possible to "forget" the additional structure without affecting the meaning of the net.

Definition 3.2 Given a Local net $L = (S, T, W, M_0, R)$, we associate to it a net $U(L) = (S, T, W, M_0)$.

Here two immediate facts relating the behaviour of L and that of $U(L)$ are stated:

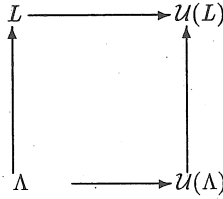
Fact 3.1 Let $L = (S, T, W, M_0, R)$ be a Local net. Then, $M_0[G_1 > M_1 \dots [G_n > M$ is a step sequence of $U(L)$ if and only if $M_0[G_1 > M_1 \dots [G_n > M$ is a step sequence of L .

Fact 3.2 Let L be a safe Local net. Then, $U(L)$ is also safe.

Processes of Petri nets are defined as a sub-class of nets without conflict and cycles. In the same way, processes of Local nets are just Local nets with the same conditions.

Definition 3.3 A Local process $\Lambda = (B, E, F, p, V)$ of a safe Local net $L = (S, T, W, M_0, R)$ is such that (B, E, F, V) is a safe Local net without initial marking, and (B, E, F, p) is a process of (S, T, W, M_0) , and V verifies that $p(b) = s$ and $p(b') = s'$ and $p(e) = t$ and $(s, t, s') \in R$ implies that $(b, e, b') \in V$.

Theorem 3.1 Let L be a safe Local net and let Λ be a process of L . Then, the following diagram commutes:



That is, if Λ is a process of L , then $\mathcal{U}(\Lambda)$ is a process of $\mathcal{U}(L)$.

A computation of a Local net does not generate a partial order, but a richer structure.

Definition 3.4 A colored partial order is a structure $\langle D, \leq, \sqsubseteq \rangle$ such that $\langle D, \leq \rangle$ and $\langle D, \sqsubseteq \rangle$ are partial orders and $d \sqsubseteq d'$ implies $d \leq d'$.

Colored partial orders have been introduced in [8] as structures for giving semantics to process description languages which take into account both causal dependencies and spatial distribution. They are called colored partial orders because, since $\sqsubseteq \subseteq \leq$, it is possible to represent a finite colored partial order $\langle D, \leq, \sqsubseteq \rangle$ with the partial order diagram of $\langle D, \leq \rangle$, where the arcs belonging also to \sqsubseteq are marked with a different color.

This definition permits to establish a connection between the theory of Local nets and the theory of location equivalence for process description languages. In fact, in [5] it is proved that *localities + causality = local/global causes*, which is an equivalence introduced for the CCS language [7] gluing together spatial and causal information, while in [8] it is shown that the semantics of local/global causes correspond to the semantics obtained using colored partial orders as observation domain. Hence, this characterization of computations of Local nets as colored partial orders offers a common language for computations (both of nets and of process description languages) reflecting causal and spatial information.

As a corollary of the previous lemma we can show how to obtain a colored partial order from a Local process and how to recover the causal relation from the colored partial order.

Corollary 3.1 Let $\Lambda = (B, E, F, p, V)$ be a Local process of a Local net L . Let $C = \langle E, \leq, \sqsubseteq \rangle$ be the structure where E is the set of transitions of the process, $e \leq e'$ iff $(e, e') \in F^*$, and $e \sqsubseteq e'$ iff $(e \leq e' \text{ and } \exists b, b', b'' \in B. (b, e, b'), (b', e', b'') \in V)^*$, where R^* denotes the reflexive and transitive closure of the relation R . Then,

1. C is a colored partial order, and
2. the partial order $\langle E, \leq \rangle$ is the partial order associated with $\mathcal{U}(\Lambda)$.

The following result characterizes the structure of the local nets corresponding to a given P/T net.

Definition 3.5 Let $N = (S, T, W, M_0)$ be a finite P/T net (i.e. a P/T net such that S and T are finite). Then, $\mathcal{L}_N = \{ \text{local nets } L \mid \mathcal{U}(L) = N \}$ forms an ordered set with the inclusion order $(S, T, W, M_0, R) \leq (S, T, W, M_0, R')$ iff $R \subseteq R'$.

All the local nets corresponding to a P/T net form an ordered set. In this set we have all the forms in which the spatial information can be incorporated in the net. This partial order turns out to be a complete lattice. However, lattices have few structure, which is not of much help when specifying systems. In fact, the set \mathcal{L}_N forms a boolean algebra, which is a richer structure with more operations.

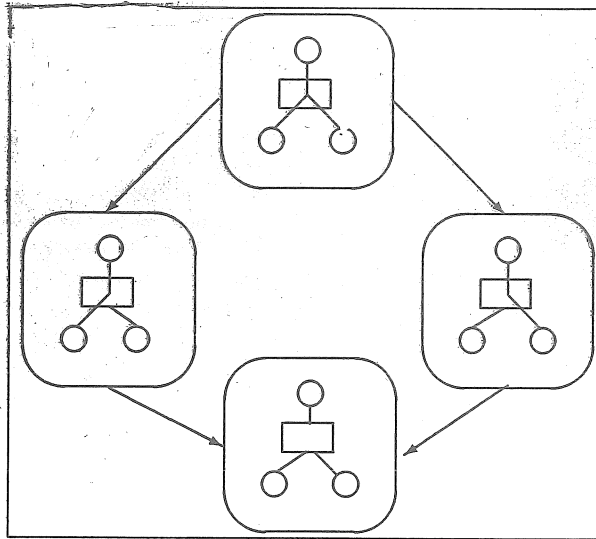


Figure 3: The boolean algebra of the fork operation

Lemma 3.1 Given a finite P/T net N , the set \mathcal{L}_N forms a boolean algebra.

Proof: Let $N = (S, T, W, M_0)$. From the characterization theorem for finite boolean algebras [3, pages 164 - 165], we have that \mathcal{L}_N is a boolean algebra iff it is isomorphic to the powerset of the atoms of \mathcal{L}_N ordered by inclusion. It is easy to check that the set of atoms of \mathcal{L}_N is the set $A = \{((s, t, s') \in (S \times T \times S) \text{ such that } W(s, t) \geq 1 \text{ and } W(t, s') \geq 1\}$. It is immediate that \mathcal{L}_N is isomorphic to $\wp(A)$, and hence it is a boolean algebra. \square

Example 3.1 Let N be the Petri net which specifies a fork operation. The boolean algebra \mathcal{L}_N associated to N is shown in figure 3. It has four elements, one corresponding to each spatial distribution: the bottom element leaves the spatial distribution unspecified, there are two nets for the UNIX like fork operation (where the parent process continues and one child is created) and the top specifies the division of a team in two subteams. Notice that using local nets it is possible to specify different kinds of fork operations, while with standard nets this is not possible.

Hence, we have now some operations over local nets which suggest a natural two-step methodology in the definition of distributed systems:

1. Define a Petri net specifying the causal relationships in the system.
2. In the boolean algebra of local nets corresponding to the net of the previous step, choose one element.

With the help of lemma 3.1 the second step can be performed joining and meeting nets, taking complements, etc.

The structure proposed provides a basis for a language for spatial distribution. In fact, a very simple language (with the operations of the algebra as constructors) can be defined, and in this way a local net can be specified with a P/T Petri net and an expression of this language.

4 Conclusions and Further Work

The main result of this work is the definition of a simple extension of Petri nets (Local nets) able to specify spatial distribution without losing interesting properties of nets and incorporating as few information as possible.

Some theoretical results are given in order to provide a basis for the use of Local nets in specifications: theorem 3.1 shows that the extension is semantics-respecting, corollary 3.1 relates this approach with a previous one in the field of process algebras [2] and lemma 3.1 gives a characterization of the structure of the local nets corresponding to a given fixed net.

The work presented here can be extended in two ways. From the definition of local processes and using the same techniques applied in the definition of colored partial orders, to define *local event structures*, extending event structures [12] with spatial information. The second extension would be to define local nets inside the algebraic context of [6].

Acknowledgments: I wish to thank Juan Vicente Echagüe for many useful comments and Cosimo Laneve because of his careful reading.

References

- [1] P. Ancilotti and M. Boari. *Principi e Tecniche di Programmazione Concorrente*. Utet-Libreria, Torino, 1987.
- [2] G. Boudol, I. Castellani, M. Hennessy, and A. Kiehn. A theory of processes with localities. Tech. Report 13/91, University of Sussex, Brighton, 1991. To appear, Proc. CONCUR '92.
- [3] B. Davey and H. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.
- [4] F. De Cindio, G. De Michelis, L. Pomello, and C. Simone. Superposed automata nets. In *Application and Theory of Petri Nets*, number 52 in Informatik Fachberichte, pages 229–279. Springer-Verlag, 1982.
- [5] A. Kiehn. Local and global causes. Technical Report 342/23/91, Technische Universität München, Munich, 1991.
- [6] J. Meseguer and U. Montanari. Petri nets are monoids. *Information and Computation*, 88:105–155, 1990.
- [7] R. Milner. *Communication and Concurrency*. Prentice-Hall International, Englewood Cliffs, 1989.
- [8] U. Montanari and D. Yankelevich. A parametric approach to localities, 1991. To appear, Proc. ICALP '92.
- [9] W. Reisig. *Petri nets – an introduction*. EATCS Monographs on Theoretical Computer Science, Volume 4. Springer-Verlag, 1985.
- [10] W. Reisig. Petri nets and algebraic specifications. *Theoretical Computer Science*, 80:1–34, 1991.
- [11] W. Vogler. Deciding history preserving bisimilarity. In *Proc. ICALP '91*, volume 510 of *Lecture Notes in Computer Science*, pages 495–505. Springer-Verlag, 1991.
- [12] G. Winskel. Event structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, Part II*, volume 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer-Verlag, 1987.