Petri Inheritance: The Foundation of Nondeterministic, Concurrent Systems

Roberto Bruni and Ugo Montanari

Dipartimento di Informatica, Università di Pisa, Italy

1 A Personal Experience

In September 1973 I (the second author) attended one of the first (actually the second) conference on *Mathematical Foundations of Computer Science* at Štrbské Pleso, High Tatras. I was traveling by car and it was quite an adventure since there were some problems at the borders due to prevention measures for cholera. At Štrbské Pleso I met for the first time Carl Adam Petri. Actually, in this occasion I also heard about Petri nets for the first time. There was a lot of excitement about nets, which were anyway already more than ten years old. I was very impressed by Petri: his taste for mathematical foundations of concurrency, reminiscent of Dedekind, was quite appealing to me. Thus I decided to invite him for the first Italian *Convegno di Informatica Teorica* that I co-organized at Mantova, in November 1974. Later on, a thread of collaboration was developed: a Pisa former student, Pippo Torrigiani, worked for a few years at Schloss Birlinghoven, Sankt Augustin, where I also visited for two one-month periods at the end of the seventies.

At Schloss Birlinghoven I had several interesting discussions with Hartmann Genrich, Kurt Lautenbach and P. S. Thiagarajan. With Petri just one session, a full afternoon long, about general net theory. As a main goal, I was interested in possible semantic domains for concurrency consisting of unfolded occurrence nets representable in three dimensions: causality, nondeterminism and concurrency. Here *maximal* sections having as dimensions nondeterminism and concurrency would represent states of the nondeterministic computation, causality × nondeterminism would show the views of certain sequential observers, while causality × concurrency would model concurrent computations. Some results were presented in a joint paper with Andrea Simonelli. I learned quite a lot during my stay at Schloss Birlinghoven, in particular concepts and constructions very useful for my successive work on concurrency. Slightly later, the work by Nielsen, Plotkin and Winskel on event structures and domains appeared at a conference on *Semantics of Concurrent Computation* and Glynn Winskel discussed in Edinburgh his thesis on events in computation, establishing a strong basis for domain-based concurrency theory. In both cases the formal developments were explicitly motivated and made possible by Petri net models and results.

A renewed interest in Petri nets matured during my sabbatical at SRI in 1987. In collaboration with José Meseguer, we wrote a paper (published at LICS 1988 and in Information and Computation) about *Petri Nets are Monoids*, which gave for the first time a presentation of the semantics of Place Transition (PT) nets in terms of symmetric monoidal categories (SMC). Later on, together with Vladimiro Sassone, a full picture connecting SMC with event structures and prime algebraic domains was developed. Finally, further work, in collaboration with the first author, made the connection stronger, in fact functorial.

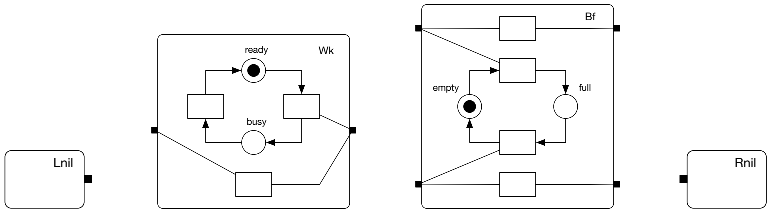
More recently, a variety of networks were represented as arrows of SMC enriched with signatures of operations equipped with axioms, concisely represented as string diagrams. This research suggested a line (by both authors, Hernán Melgratti and Pawel Sobocinski) where compositional versions of Petri nets with observations and connector algebras have been defined. Particularly interesting is the enabling mechanism, defining a distributed choice in the presence of distribution, concurrency and nondeterminism. An abstract definition of such mechanisms has shown that it is the most general possible. This universal result confirms the importance and generality of Petri nets, also in comparison with other formalisms (BIP, REO, etc.). This basic outcome is presented succinctly in the remainder of this paper.

2 Petri Nets as a Connector Algebra

Nets composition is an useful tool for modelling systems and proving their properties as well as for comparing the expressive power of Petri nets with other formalisms. Several notions of composition have appeared in the literature, often based on interfaces made of places, of transitions or of both. Notable examples are the approaches by Nielsen, Priese and Sassone, based on combinators, by Best, Devillers and Koutny, based on the Petri box algebra, by Baldan, Corradini, Ehrig, and König, based on open nets, by Reisig, based on interface nets, by Katis, Sabadini, and Walters, based on the bicategory of Span(Graph), by van der Aalst and others, based on net modules or the very recent Springer book by Roberto Gorrieri that surveys suitable process algebras that can represent specific classes of Petri nets. Here we overview nets with boundaries, as introduced by Pawel Sobocinski and studied in our joint work *Connector algebras for C/E and P/T nets' interactions.*[1] Interestingly, they have been exploited to compare the expressiveness of Petri nets with other formalisms for modelling distributed decisions, like Reo and BIP.

Nets with boundaries extend ordinary Petri nets with left and right interfaces that can be used for composition. An interface is just a list of ports, to which transitions can be attached. Names of ports are not important, so they are named as natural numbers, according to the position they occupy in the interface. The distinction of left and right interfaces is convenient for defining sequential composition of nets with boundaries, but it would be misleading to think about input/output distinction. The general idea is that a transition attached to some ports is a sort of fragmented transition, that must be completed with other fragments via an interaction on shared ports. Graphically, this intuition is rendered by using undirected arcs to connect ports with transitions.

[1] Logical Methods in Computer Science 9(3) (2013).



(a) Lnil (b) Worker model (c) Buffer model (d) Rnil

Fig. 1: Some nets with boundaries

The operational semantics of nets with boundaries can be expressed as a labelled transition system, whose labels are pairs modelling the observable interactions on the ports of the left and right interfaces. Interestingly, the corresponding bisimilarity equivalence is a congruence w.r.t. parallel and sequential composition of nets with boundaries. A concise formal account follows.

For a natural number $n \in \mathbb{N}$, we let $\underline{n} = \{0, 1, ..., n-1\}$. Given $m, n \in \mathbb{N}$, a *net with boundaries* $N : m \to n$ is a tuple $N = (S, T, {}^{\circ}-, -^{\circ}, {}^{\bullet}-, -^{\bullet})$ where: S is the set of places, T is the set of transitions, the functions ${}^{\circ}-, -^{\circ} : T \to 2^S$ assign sets of places, called respectively pre-set and post-set, to each transition and the functions ${}^{\bullet}- : T \to 2^{\underline{m}}$ and $-^{\bullet} : T \to 2^{\underline{n}}$ map transitions to the left and right boundaries of N, respectively.

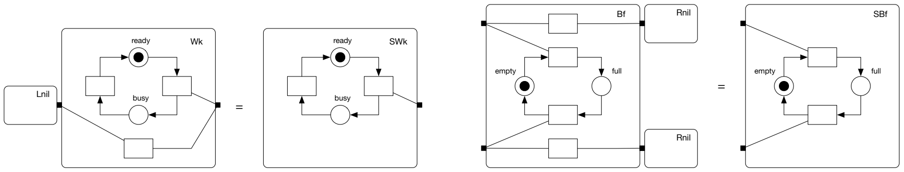
Figure 1 presents some nets with boundaries that we use as a running example. The net $\mathrm{Wk} : 1 \to 1$ in Fig. 1b models a worker (either a producer or a consumer) that is *ready* to perform some action to enter the *busy* state. Its left and right interfaces consist of just one port. The transition from ready to busy is attached to the unique port of the right interface. The bottom transition is attached to both ports: it will be used to compose the worker with other workers. The transitions that share a port compete for interacting on that port.

The net $\mathrm{Bf} : 2 \to 2$ in Fig. 1c models a buffer to store the item produced by some worker. The buffer has only one slot: it can be *empty* and ready to accept an item, or *full* and ready to dispense the stored item. In this case the left and right interfaces have two ports each: the topmost port in the left interface will be used to combine the buffer with some producers; the bottom port on the left interface will be used to combine the buffer with some consumers; the two ports in the right interface can be used to combine the buffer with other buffers.

The nets $\mathrm{Lnil} : 0 \to 1$ in Fig. 1a and $\mathrm{Rnil} : 1 \to 0$ in Fig. 1d are empty: they can be used to restrict interaction on some ports.

Given two nets $N_1 : m_1 \to n_1$ and $N_2 : m_2 \to n_2$, their parallel composition is the net $N_1 \otimes N_2 : m_1 + m_2 \to n_1 + n_2$ obtained by the disjoint union of the two nets, up to an obvious rearrangement of their left and right interfaces (the ports of N_1 precede those of N_2).

Sequential composition is defined when the right interface of one net with boundaries matches with the left interface of another net. Given $N_1 : m \to k$ and $N_2 : k \to n$, their sequential composition is the net $N_1 ; N_2 : m \to n$ whose



(a) The net SWk = Lnil; Wk (b) The net SBf = Bf; (Rnil ⊗ Rnil)

Fig. 2: Some composed nets

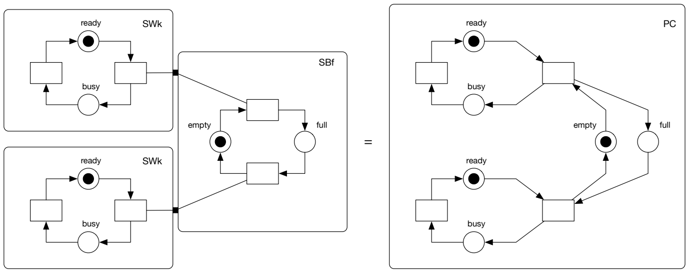


Fig. 3: The net PC = (SWk ⊗ SWk); SBf

places are the disjoint union of the places of N_1 and N_2 and whose transitions are all the possible (minimal) synchronisations of (mutually independent) sets of transitions from N_1 and N_2 that interact over the k shared ports.

Formally, two transitions t, u are said to be *independent* when their sources as well as their targets are separated, i.e. when

$$ {}^\circ t \cap {}^\circ u = \varnothing \ \wedge \ t^\circ \cap u^\circ = \varnothing \ \wedge \ {}^\bullet t \cap {}^\bullet u = \varnothing \ \wedge \ t^\bullet \cap u^\bullet = \varnothing $$

and a set U of transitions is mutually independent when, for all $t, u \in U$, if $t \neq u$ then t and u are independent. We say that a *synchronisation* is a pair (U_1, U_2) with U_1 and U_2 mutually independent (disjoint) sets of transitions of N_1 and N_2, respectively, such that: (1) $U_1 \cup U_2 \neq \varnothing$ and (2) $U_1^\bullet = {}^\bullet U_2$. The set of synchronisations inherits an ordering from the subset relation, i.e. $(U_1, U_2) \subseteq (U_1', U_2')$ when $U_1 \subseteq U_1'$ and $U_2 \subseteq U_2'$. A synchronisation is *minimal* when it is minimal with respect to this order. The intuition is that a minimal synchronisation cannot be broken in simpler synchronisations. As a special case, note that any transition t_1 in N_1 (respectively t_2 in N_2) not connected to the shared boundary k defines a minimal synchronisation $(\{t_1\}, \varnothing)$ (respectively $(\varnothing, \{t_2\})$).

Next, we discuss some compositions of the nets of our running example. First we can model a single worker by restricting the left interface of a worker: the corresponding net SWk = Lnil; Wk is shown in Fig. 2a. Similarly, we can model a single buffer by restricting the right interface of composable buffer: the corresponding net SBf = Bf; (Rnil ⊗ Rnil) is shown in Fig. 2b. Then, we can assemble two workers (a producer and a consumer) with a buffer: the corresponding net PC = (SWk ⊗ SWk); SBf is shown in Fig. 3.

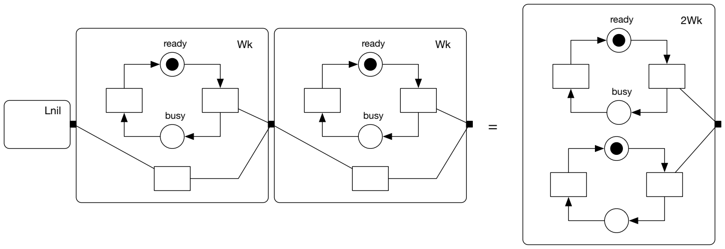


Fig. 4: The net 2Wk = Lnil; Wk; Wk

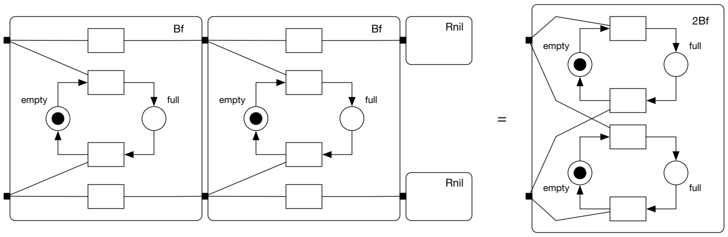


Fig. 5: The net 2Bf = Bf; Bf; (Rnil $\otimes$ Rnil)

It is interesting to observe what happens when we compose several workers or several buffers together. The net 2Wk = Lnil; Wk; Wk is given in Fig. 4, while the net 2Bf = Bf; Bf; (Rnil $\otimes$ Rnil) is given in Fig. 5. We can then compose a system with two producers, one consumer and two buffers as shown in Fig. 6. Of course, the composition immediately generalises to any number of workers and buffers, with the advantage that the notion of synchronisation automatically accounts for the combinatorial explosion of cases (any producer can post an item in any empty buffer and any consumer can retrieve an item from any full buffer).

Parallel and sequential compositions of nets with boundaries have been useful to show several results. First, it has been shown that Petri nets with boundaries are equivalent to an algebra of connectors that is freely generated by a small set of basic stateless components (identity, swap, synch, mutex, hiding and noact) together with a one-position buffer component. Exploiting this correspondence, Petri nets with boundaries have been shown equivalent to the exogenous coordination framework Reo proposed Arbab et al. Moreover, exploiting compositionality at the semantic level, Petri nets with boundaries have also been shown equivalent to the BIP framework (in the absence of priorities), even in the hierarchical case. An informal account of the above results can be found in our joint paper with Hernán Melgratti *A Survey on Basic Connectors and Buffers*,[2] where interesting connections with the tile model and the wire calculus are also drawn.

Such semantical equivalences provide further evidence that Petri nets (with boundaries) offer a core stateful model that accounts for distribution, concur-

[2] Proc. of FMCO 2011, LNCS vol.7542, pp. 49–68, 2013.

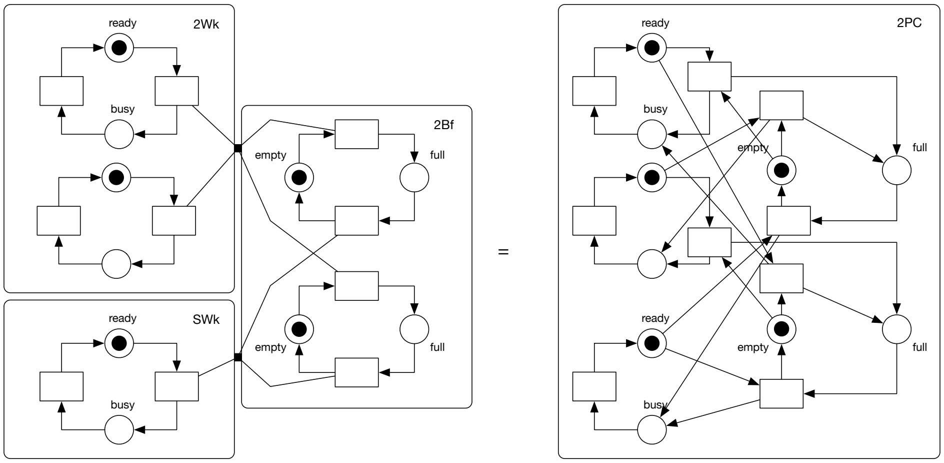


Fig. 6: The net $\mathsf{2PC} = (\mathsf{2Wk} \otimes \mathsf{Wk}); \mathsf{2Bf}$

rency and nondeterminism and that Carl Adam Petri vision was very insightful in this respect.

However, decades of studies have not been sufficient to solve in a fully satisfactory manner all the issues related to Petri nets. In fact they give ground to a still very active and lively research area, with many interesting open problems. Having the occasion to pick one of them, we point the attention to the replacement of nondeterminism with probability distributions, i.e., to the coexistence of concurrency and probability. A number of probabilistic versions of Petri nets have been proposed, but most of them introduce time dependent stochastic distributions, thus giving up the time and speed independence feature typical of proper truly concurrent models. In the absence of confusion, a solution has been proposed by Varacca and Winskel based on event structures. In the more general case, Abbes and Benveniste's branching cells provide a more complex solution but they miss compositionality and a description of the probabilistic execution in terms of some concurrent transition system.

We foresee that the connection between Petri nets (with boundaries) and algebra of connectors can be instrumental in devising a compositional approach to the definition of a general model where concurrency and probability are accounted for in a fully satisfactory manner. Also, remarkable analogies between probabilistic Petri processes without confusion and Bayesian networks may suggest how to extend well known analysis techniques from the latter to the former model.