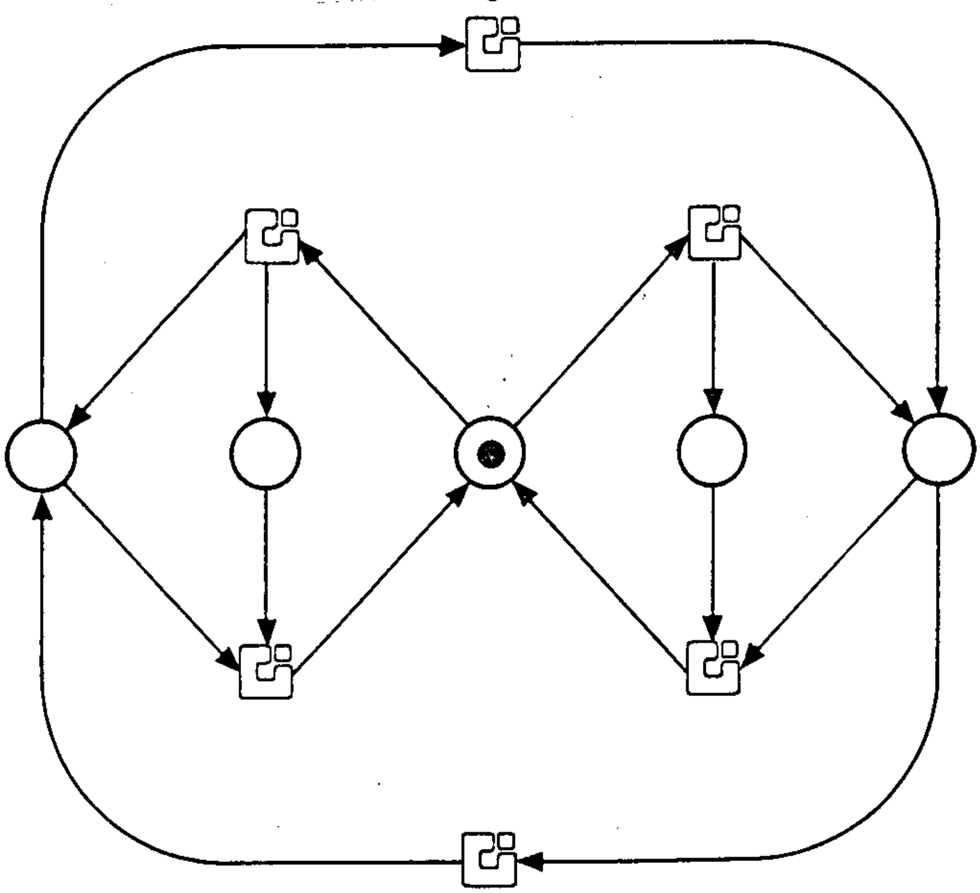
Petri Net Newsletter 33

August 1989

published by Gesellschaft für Informatik
Special Interest Group on Petri Nets and related System Models



Contents

Imprint: "Petri Net Newsletter" • 900 copies printed • Price per year: DM 20.–.
This newsletter is an information service for the members of the Special Interest Group 1.1.2 "Petri Nets and Related System Models" of Gesellschaft für Informatik, Godesberger Allee 99, D-5300 Bonn 2, West Germany. Three issues a year are being published.

Scope of concern: This newsletter serves as a medium for the rapid distribution of any information about Petri Nets and related system models all over the world. Topics include:

- technical contributions
- information on software packages
- correspondence
- problems and puzzles
- conference announcements
- conference reports
- reports on departments, institutes, companies, projects, research, education
- available reports
- abstracts of recent publications
- book reviews

Subscription: Application forms can be obtainend from Gesellschaft für Informatik and from the editors. As a simplified procedure, just send an informal letter and include cash.
For legal reasons, subscription of the Newsletter implies the membership in the Special Interest Group 1.1.2 of the GI. This causes no further obligations or charges.

Contributions: They should be sent to one of the editors. Any contributions to the field are welcome. They are not refereed. As they will be printed as submitted, make sure that no space is wasted.

Forms of payment

An excellent opportunity is the annual International Conference on Application and Theory of Petri Nets. Another simple form is to send cash to:

Gesellschaft für Informatik
Godesberger Allee 99
D–5300 Bonn 2

All kinds of payment should include the subscription number as given in the upmost line of your address label (GI–*nnn* or PN–*nnn*).

For transfers by account, the cheapest way is a transfer from a post–office transfer account (not from a banking account) to the post–office transfer account of the Gesellschaft für Informatik no. 1981 39–502 at Cologne (BLZ 370 100 50).

Another cheap way is a cheque in DM or, where the national regulations allow it, a cheque in some other convertible currency. Please use the most recent rate of exchange for buying foreign bills in DM (we get the cheque cashed free of charge).

A third cheap way is paying in cash at a post office for credit to our post–office transfer account ("Mandat de versement international", form MP 16).

Thus the deduction of banking charges from the invoice amount can be avoided.

If you charge a bank with the transfer, a notice such as "free of charge for the payee" does not always work; some intermediate banks deduct DM 7.00 or more. So, please transfer only to the account of the Gesellschaft für Informatik no. 46 581 at the Sparkasse Bonn (BLZ 380 500 00) and increase invoice amount by DM 7.00.

...from the editors

We are sorry for the delay of this newsletter, due to holidays, some deadlines have not been met. The 10th European Conference on Petri Nets has been a great success. We would have liked very much to publish a report on this event in this issue of the Newsletter. Who of the 200 particpants is willing to write a report for the next issue?

With best regards

O. Herzog
IBM Germany Scienctific Center
Institute for Knowledge Based Systems
Dept. 3527, 7000-75
P.O. Box 80 08 80
D–7000 Stuttgart 80
Phone: +49–711–6695–501

W. Reisig
Institut für Informatik
TU München
P.O. Box 20 24 20
D–8000 München 20
Phone: +49–89–2105–2405
Telefax: +49–89–2105–8207
e_mail:
reisig@lan.informatik.tu-muenchen.dbp.de

R.Valk
FB Informatik
Universität Hamburg
Rothenbaumchaussee 67–69
D–2000 Hamburg 13
Phone: +49–40–4123–5675

Cover Picture Story

AC/DC-Systems

Jörg Desel
Institut für Informatik der Technischen Universität München
Arcisstraße 21, D-8000 München 2

Eike Best
Institut für Informatik der Universität Hildesheim
Samelsonplatz 1, D-3600 Hildesheim

Free choice systems (FC-systems) are a prominent class of systems for which situations called confusions are excluded by the structure of the underlying nets. It is also known that FC-systems are not the biggest class with this property; they are a proper subclass of extended free choice systems (EFC-systems). However, in [Best, Shields 83], for instance, it is shown that EFC-systems do not have the capability of modelling really more than FC-systems. Each EFC-system can be simulated by an FC-system which is just a slight modification of the original system. In [Best, Desel 89] it is shown that most of the established free choice theory can be transferred to EFC-systems.

Now the question arises: are EFC-systems the biggest class excluding confusion? In the hierarchy of system classes proposed in [Best, Shields 83] the next class bigger than EFC-systems are asymmetric choice systems (AC-systems). These systems exclude one kind of confusion, called symmetric confusion while the other kind (asymmetric confusion) can occur.

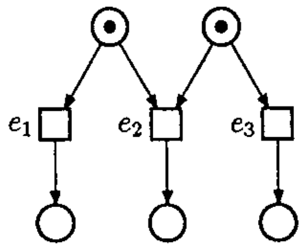
In this contribution we present another superclass of EFC-systems in which asymmetric confusion is excluded. For more or less obvious reasons we christen this class DC-systems (dissynchronous choice systems). Like the other classes it is defined by a property of the net structure of systems. Since every situation of confusion is either asymmetric or symmetric (or both), the intersection of AC-systems and DC-systems is a class of confusion-free systems. The cover picture system is such an AC/DC-system which is not, however, an EFC-system. And, as we shall argue at the end, it can not be easily simulated by an EFC-system.

Hence the class of AC/DC-systems is a structurally defined class of systems whithout confusion which properly includes the class of EFC-systems.

Before we formally define confusion and show the confusion-freeness of DC-systems we recall the informal meaning of confusion.
A situation of confusion is an interaction of conflicting and concurrently occurring events. In the presence of confusion a lot of propositions fail to hold, which otherwise could be proven. Moreover, even some fundamental concepts are harder to express if confusion is not excluded.

If two (or more) events, say e_1 and e_2, are in a non-confused conflict situation, they compete for a token and only one of them can occur. But they don't have to worry about the time they need for that competition. Instead, the occurrence of one of them defines – locally – the next point of time. In a confused situation things are different. We can roughly distinguish two phenomena:

- A third transition e_3, which is completely independent to at least one of the competing transitions – e_1 in the figure below – intervenes in the quarrel and disables e_2. Or, seen positively, it mediates or even solves the conflict in favour to e_1. Note that there must have been a conflict between e_2 and e_3 before as well since both have been enabled and the occurrence of e_3 disables e_2. However, we still concentrate on the conflict between e_1 and e_2.

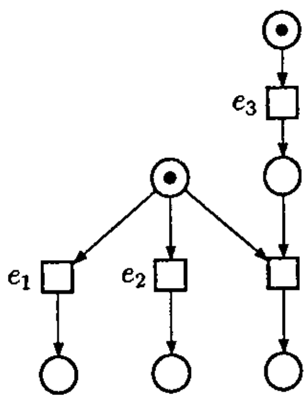


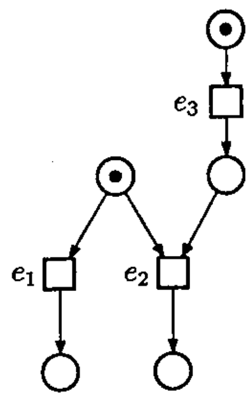
  The competing partners could have settled before with the same solution. If e_1 and e_3 occur concurrently there is no causal ordering defined between them. It is impossible to decide whether e_3 solved the conflict between e_1 and e_2 or e_1 solved the conflict between e_2 and e_3.
  This kind of confusion is called conflict decreasing in [Thiagarajan 87] and symmetric confusion below.

- The other phenomenon is, in a sense, similar to symmetric confusion. Considering again the competing events e_1 and e_2, it is possible that a third, independent, event e_3 occurs with the following consequence: more events get enabled which also need the token e_1 and e_2 fight for (see the left figure below). So if they do not agree in time, they all might lose or, at least, solving the conflict gets harder.

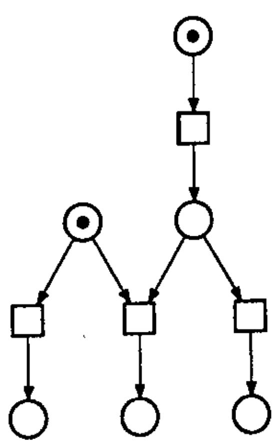
  Even worse, the same can happen if there was no conflict at all before. Consider an enabled event e_1, in conflict with no other event. Eventually it will occur. But if it is sure that nothing can happen and that it can wait arbitrarily long, a

conflict between e_1 and another event e_2 might occur and e_1 might even lose its concession without occurring (see the right figure below). Assume now that e_1 wins the conflict. Again, it could have occurred before e_2 was enabled and there is no hope to decide whether there ever was a conflict.
This kind of confusion is called conflict increasing confusion [Thiagarajan 87] or asymmetric confusion.





- Obviously both phenomena can occur together and in this case we speak about a confused situation which is both symmetrical and asymmetrical.



The formal definition of confusion highly depends on formal definitions of conflict and of concurrency. Both definitions turn out to be easy for elementary net systems (EN-systems, as defined in [Thiagarajan 87]) but for place transition systems some problems arise (see [Goltz 86]). Hence we give formal definitions of confusion for EN-systems here:

If c is a case of an EN-system and u is a step (i.e. a set of independent events) enabled at c we denote by cu the resulting case after the occurrence of u at c, i.e., $cu = (c - {}^\bullet u) \cup u^\bullet$.

Definition 1

An EN-system Σ exhibits symmetric (respectively, asymmetric) confusion if there is a reachable case c and events e_1, e_2, e_3 such that e_1 and e_3 are distinct events which are concurrently enabled at c and furthermore the following holds:

Symmetric confusion:
e_2 is enabled at c and e_2 is neither enabled at $c\{e_1\}$ nor at $c\{e_3\}$.

Asymmetric confusion:
e_2 is enabled at $c\{e_3\}$ and e_2 is neither enabled at c nor at $c\{e_1, e_3\}$.

Σ exhibits confusion if it exhibits asymmetric confusion *or* if it exhibits symmetric confusion.

Note that this definition implies $e_1 \neq e_2 \neq e_3$.
In case of symmetric confusion $e_1 \neq e_2$ since the occurrence of e_3 disables e_2 while e_1 and e_3 can occur concurrently. $e_2 \neq e_3$ follows by symmetry.
In case of asymmetric confusion $e_1 \neq e_2$ since e_1 but not e_2 is enabled at c and $e_2 \neq e_3$ since e_2 is enabled at $c\{e_3\}$.

A conflict-increasing confusion (as defined in [Thiagarajan 87]) corresponds to a situation in which there is asymmetric confusion but no symmetric confusion and a conflict-decreasing confusion corresponds to a situation in which there is symmetric confusion but no asymmetric confusion.

In the sequel we shall concentrate on *contact-free* elementary net systems. This has the advantage that for all pairs of events, which can ever get into a conflict situation, we can find a common condition in their presets. Furthermore, if the occurrence of an event e ever enables an event e', then there is a condition in the intersection of the postset of e and the preset of e'. We say e enables e' at a case c if e' is enabled at $c\{e\}$ but not at c.

Definition 2

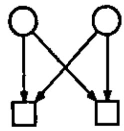
Let $\Sigma = (B, E, F, c_0)$ be an EN-system. Σ is called

(a) free choice system (FC-system) if
$\forall b_1, b_2 \in B: \; b_1^\bullet \cap b_2^\bullet \neq \emptyset \Longrightarrow b_1^\bullet = b_2^\bullet = \{e\}$ for some $e \in E$.

allowed:

(b) extended free choice system (EFC-system) if
$\forall b_1, b_2 \in B:\ b_1^\bullet \cap b_2^\bullet \neq \emptyset \Longrightarrow b_1^\bullet = b_2^\bullet$.

allowed:

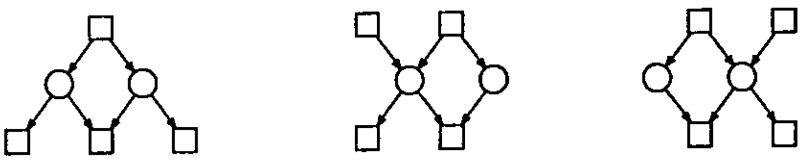


(c) asymmetric choice system (AC-system) if
$\forall b_1, b_2 \in B:\ b_1^\bullet \cap b_2^\bullet \neq \emptyset \Longrightarrow b_1^\bullet \subseteq b_2^\bullet \vee b_2^\bullet \subseteq b_1^\bullet$.

allowed:

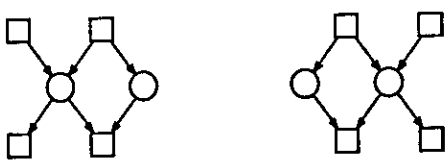
(d) dissymmetric choice system (DC-system) if
$\forall b_1, b_2 \in B:\ b_1^\bullet \cap b_2^\bullet \neq \emptyset \Longrightarrow b_1^\bullet \subseteq b_2^\bullet \vee {}^\bullet b_2 \subseteq {}^\bullet b_1$.

allowed:



(e) AC/DC-system if it is both an AC-system and a DC-system.

allowed:



Obviously each EFC-system is an AC/DC-system. The cover picture is an example of an AC/DC-system which is not an EFC-system.
Next we show that AC-systems (DC-systems) do not exhibit symmetric (asymmetric, respectively) confusion:

Proposition 3
Contact-free AC-systems do not exhibit symmetric confusion.

Proof: Assume that a contact-free AC-system $\Sigma = (B, E, F, c_0)$ exhibits symmetric confusion.
Then there exists a case c of Σ, events $e_1, e_2, e_3 \in E$ with ${}^\bullet e_1 \cap {}^\bullet e_3 = \emptyset$ such that the following holds: e_2 is enabled at c but neither enabled at $c\{e_1\}$ nor at $c\{e_3\}$.
Since e_2 is at c in a conflict situation with e_1 there is a condition $b \in {}^\bullet e_2 \cap {}^\bullet e_1$.
$b \notin {}^\bullet e_3$ since ${}^\bullet e_1 \cap {}^\bullet e_3 = \emptyset$.
For similar reasons we can find a condition $b' \in {}^\bullet e_2 \cap {}^\bullet e_3 - {}^\bullet e_1$.
In all, $e_2 \in b^\bullet \cap b'^\bullet$ but neither $b^\bullet \subseteq b'^\bullet$ nor $b'^\bullet \subseteq b^\bullet$.
Hence Σ is not an AC-system. ∎

Proposition 4
Contact-free DC-systems do not exhibit asymmetric confusion.

Proof: Assume that a contact-free DC-system $\Sigma = (B, E, F, c_0)$ exhibits asymmetric confusion.
Then there exists a case c of Σ, events $e_1, e_2, e_3 \in E$ with ${}^\bullet e_1 \cap {}^\bullet e_3 = \emptyset$ such that e_2 is enabled at $c\{e_3\}$ and e_2 is neither enabled at c nor at $c\{e_1, e_3\}$.
We can find a condition $b \in {}^\bullet e_1 \cap {}^\bullet e_2$ since e_1 disables e_2. Furthermore we can find a condition $b' \in {}^\bullet e_2 \cap e_3^\bullet$ since e_3 enables e_2.
$e_2 \in b^\bullet \cap b'^\bullet$ and so $b^\bullet \cap b'^\bullet \neq \emptyset$.
$e_1 \in b^\bullet - b'^\bullet$ since e_1 and e_3 can occur concurrently at c. So $b^\bullet \subseteq b'^\bullet$ does not hold. $e_3 \in {}^\bullet b' - {}^\bullet b$, again since e_1 and e_3 can occur concurrently at c. So also ${}^\bullet b' \subseteq {}^\bullet b$ does not hold.
Hence Σ is not a DC-system. ∎

As a corollary we get the result that contact-free AC/DC-systems exhibit no confusion at all.

Finally we want to reason about whether the cover picture could be simulated by an FC-system, i.e., whether it can be modified such that essentially the behaviour does not change and the modified system is an FC-system.
Consider the uppermost event and call it e. There are two different cases in which e is enabled. In one of these cases e is in a conflict situation with another event while e is the only event enabled by the other case. Such situations are typically non-FC like. Hence without splitting e in different events the system cannot be simulated by an FC-system.

References

[Best, Desel 89] E.Best, J.Desel: Partial Order Behaviour and Structure of Petri Nets. Arbeitspapiere der GMD No.373 (1989).

[Best, Shields 83] E.Best, M.W.Shields: Some Equivalence Results on Free Choice Nets and Simple Nets, and on the Periodicity of Live Free Choice Nets. Springer LNCS Vol.159, pp.141-154 (1983).

[Goltz 86] U.Goltz: How many transitions may be in conflict? Petri Net Newsletter Vol.25 (1986).

[Thiagarajan 87] P.S.Thiagarajan: Elementary Net Systems. Springer LNCS Vol.254 pp.26-59 (1987).

Technical Contributions

Logics for Petri-nets: Partial order logics, Branching time logics and how to distinguish between them

A.Sinachopoulos
Laboratoire d'Informatique Théorique
Université Libre de Bruxelles
CP, Boul. de Triomphe
B-1050 Bruxelles
Belgium

Abstract: In the first section of this note we examine the differences between partial order and branching time logics. In the second section we discuss possible applications of these logics to Petri-net theory.

1 Partial order and branching time logics

In the last three years there has been a lot of discussion about linear order, partial order and branching time logics. It is not so clear however that the notions associated to those kind of logics are well understood, as sometimes very fine differences are involved.

An axiomatic system is intended to describe a model and vice versa. In the best case axiomatics and model are related by completeness, i.e. all formulas valid in the model are derivable from the axiomatics.

In generally we try to axiomatize properties of a concrete model, e.g. the standard axiomatisation of pre-orders is the axiomatic system $\mathcal{L}$ with the following rules: modus ponens and generalisation for $\Box$ and $\overleftarrow{\Box}$ and the following axioms :

$$
\begin{array}{ll}
\Box(p \to q) \to (\Box p \to \Box q), & \overleftarrow{\Box}\,(p \to q) \to (\overleftarrow{\Box}\, p \to \overleftarrow{\Box}\, q) \\
p \to \Box\, \overleftarrow{\Diamond}\, p, & p \to \overleftarrow{\Box}\, \Diamond p \\
\Box p \to \Box\Box p, & \overleftarrow{\Box}\, p \to \overleftarrow{\Box}\,\overleftarrow{\Box}\, p \\
\Diamond p \leftrightarrow \neg\Box\neg p, & \overleftarrow{\Diamond}\, p \leftrightarrow \neg\, \overleftarrow{\Box}\, \neg p
\end{array}
$$

as well as all axioms of the classical propositional logic.

$\mathcal{L}$ is an extension of the well-known S_4 axiomatic system by past operators and the corresponding axioms, [LT,Pe,Sin].

The models of such axiomatics and all models of temporal logic as well can be considered as Kripke structures (W, R, V) where W is the set of our possible "worlds", usually

a set of states, R is a relation in W, e.g. for the above mentioned axiomatic system $\mathcal{L}$ R is a pre-order, and V is the valuation function s.t. for all atomic propositions a and for all $S \in W$ we have $\models_{S} a$, a is valid at S, iff $a \in V(S)$. (Atomic propositions are the elementary propositions of the logic, all formulas of propositional logic are built up by atomic propositions.)
In such a model composed of states, $\Box$ is interpreted as "for all states greater than or equal to the present state" and $\overleftarrow{\Box}$, the mirror of $\Box$, as "for all states less than or equal to the present state".
In logics there is no specific axiom characterising antisymmetry, [Pe,Sin]. Hence the above mentioned axiomatics $\mathcal{L}$ characterizes both pre- and partial orders as well.

A tree is a partial order in which there is a root element with no predecessors and all other elements have exactly one predecessor,[RU]. That means that the route from a given element to the root element is ordered linearly. This property, the backward linearity, is expressed formally by "if T is a tree then: if for all $x, y, z \in T$ $x \leq z$ and $y \leq z$ hold, then $x \leq y$ or $y \leq x$" and is axiomatized by the axiom

$$(1) \quad (\overleftarrow{\Box}(p \vee q) \wedge \overleftarrow{\Box}(p \vee \overleftarrow{\Box} q) \wedge \overleftarrow{\Box}(\overleftarrow{\Box} p \vee q)) \to \overleftarrow{\Box} p \vee \overleftarrow{\Box} q\,.$$

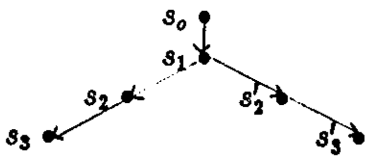
This axiom is valid only in trees. A tree being a partial order is a model for $\mathcal{L}$ but in general a pre- or a partial order cannot satisfy (1).

If we need axiomatics for linear orders we only have to extend the axioms of $\mathcal{L}$ by the backward linearity (1) and the forward linearity

$$(2) \quad \Box(p \vee q) \wedge \Box(p \vee \Box q) \wedge \Box(\Box p \vee q)) \to \Box p \vee \Box q\,.$$

The resulting axiomatic system has an important property characterising linear orders: If we have a next time operator we can derive a kind of induction, e.g. in [Kr] the rule (ind) $A \to B,\ A \to \otimes A \ \vdash\ A \to \Box B$, or in [Sti] $\Box(A \to \otimes A) \to (\otimes A \to \Box A)$. In both cases $\otimes$ is a strong next time operator, i.e. $\neg \otimes \neg A \leftrightarrow \otimes A$, as we expect in linear time logics where only one predecessor and only one successor is possible.

But when we deal with "branching time" logics in informatics we do not only mean partial orders provided with backward linearity, i.e. trees, because we do not necessarily consider the whole tree only: Our tree is the computation tree of a concurrent system and we want to state properties of single computations. This cannot be done by $\Box$ and $\Diamond$ which are more general. E.g. when we have $\models_{s_o} \Box A$ (3) and s_o is a state of the tree



Then (3) means that A is valid in all states greater or equal than s_o. This way we cannot study the computations $s_o s_1 s_2 s_3$ and $s_o s_1 s_2' s_3'$ separately because the logic we have cannot express the notion "single computation". Moreover in most branching time logics, [BPM,ES,PW,Sti], computations are paths, i.e. linearly ordered sets of states. In [Rei2] computations are runs, i.e. partially ordered sets of states. Hence if we want to talk about single computations we have to enrich our formal language with symbols denoting them and axiomatics expressing their properties. This can be done by two ways:

1. We extend our language by two operators $\forall$ and $\exists$, interpreted as "for all paths (runs)" and "for some paths (runs)" respectively. Then we extend our axiomatics by axioms expressing the properties of each path (run), where paths are linearly ordered and runs are partially ordered, using $\Box$, $\Diamond$, eventually a next time operator and the new operators $\forall$ and $\exists$. By this method [Sti] has given axiomatics for CTL^* of [ES].
   Our semantics is also modified by the existence of formulas being valid at states, state formulas, and formulas being valid at paths, path formulas, [ES].

2. We eliminate all modal operators $\Box$, $\Diamond$ and next time operator $\otimes$ and give new axioms using $\forall\Box$, $\exists\Diamond$ and $\forall\otimes$,[BPM]. If by our axioms we describe paths, we expect to have a kind of induction along paths. The new operators are interpreted semantically by state validity, e.g $\forall\Box p$ is valid at a state S iff for all paths t starting from S and for all states S' of t we have that p is valid at S'. The interpretation of $\forall\otimes$ is analogous depending on what properties we want the next state operator to have. Each new operator is not separable, it is considered as a single one.

By the latter method we only have formulas that are valid at states and we do not need to distinguish between state and path formulas. However our expressive power is restricted, e.g. the formula $\forall\exists\Box\Diamond p$, which is a well-formed formula by the former method, is not a well-formed formula in the latter formalism.

In this way, in branching time logics we have both notions of validity, namely validity at states, well known from linear and partial order logics, and validity at paths or runs. It is interesting that we define that a formula f is valid at a path or run π not iff f is valid at all states of π but iff f is valid at the first state of π. This is very well explained by the fact that computations have a given input (= first state). But there is another technical reason for this definition: Let f be the formula $\Diamond g$. $\models_\pi \Diamond g$ means intuitively that g is valid at some state S_k of π. We cannot demand that $\Diamond g$ is valid at all states of π, we do not know whether g is valid at the successor S_{k+1} of S_k. Hence it seems natural to say that $\Diamond g$ is valid at π iff $\Diamond g$ is valid at the first state of π.

A very interesting example of temporal axiomatics for concurrent systems is POTL. POTL contains both branching time operators, $\forall\Box$ and $\exists\Diamond$ and their mirrors for the past, and a partial order next time operator similar to that of [Reil,Sin] as well. This combination makes POTL a very rich and flexible axiomatic system.
The branching time operators of POTL, $\forall\Box$ and $\exists\Diamond$ and their mirrors cannot be separated to $\forall$ and $\Box$, $\exists$ and $\Diamond$ respectively, neither axiomatics nor semantics allows this. Moreover $\forall\Box$, $\exists\Diamond$ and their mirrors are tailored to describe paths, i.e. linear orders. Hence POTL describing paths cannot be syntactically and semantically akin to logics describing linear or partial orders.

2 Logics for Petri-nets

And what kind of logics shall we apply to net and Petri-net theory?

Let $N = (B, E; F)$ be a net. We know already that $(B, \leq_1)$, $(E, \leq_2)$ and $(B \cup E, \leq_3)$ are pre-orders, where $\leq_1$, $\leq_2$ are the orders introduced by F in B and E respectively and $\leq_3 = \leq_1 \cup \leq_2$. Moreover, if $\Sigma = (B, E, F; C)$ is a CE-net we can talk about the case graph of Σ, about the set of subsets of E which are enabled in the cases of Σ and about the slices of a process $\pi = (K, p)$ of Σ where K is an occurrence net. The first two sets have a pre-order structure and the slices of π this of a lattice. Thus all above mentioned structures can be considered as models of axiomatics describing pre-orders or lattices and be axiomatized by $\mathcal{L}$-axiomatics provided eventually with one more lattice axiom. If we want to have a next time operator $\otimes$ adequate for pre-orders we can extend the $\mathcal{L}$-axiomatics by axioms describing predecessors and successors, [Sin]. The semantical interpretation of $\otimes$ depends on the model, e.g. we have first to define which conditions $b' \in B$ or events $e' \in E$ of $(B, \leq_1)$ or $(E, \leq_2)$ are successors of $b \in B$ or $e \in E$ respectively and then to define semantics as $\models_{\{b\}} \otimes a$ or $\models_{\{e\}} \otimes a$ iff $\models_{\{b'\}} a$ or $\models_{\{e'\}} a$ respectively.

However from the above mentioned structures preferable models for $\mathcal{L}$-logics describing pre- and partial orders are the structure of events where the corresponding axiomatics is provided with a specific operator expressing conflicts, [LT,Pe], the structure of cases, [Sin], and the structure of slices of a process, [Rei1,Sin].

Moreover if we abstract all cases of all firing sequences to a case tree we have a model for the axiomatics UB, [BPM], CTL, [ES], CTL^*, [ES], and POTL, [PW]. (Finite firing sequences can be considered as infinite by means of the stuttering of [La].) Cases ordered by the ordering of their holding play the rôle of paths allowing us to have operators of the kind $\forall\Box$ or $\exists\Diamond$. Whether these operators can be separated to $\forall$, $\Box$ and $\exists$, $\Diamond$ depends on the given axiomatics and semantics.

If we want to reason about runs by means of branching time operators, [Rei2], we can use occurrence structures, [Str], or net unfolding, [W]. Our model can then be considered as a tree whose branches are runs. In this model, branching is the result of an existing conflict in the given system. However the corresponding axiomatics must express the partial order of slices of runs according to semantics of [Rei2].

But axiomatics for pre-orders or branching time are very general reflecting only the occurrence ordering of certain elements of a system, e.g. conditions, events, cases or slices of processes. Such axiomatics do not inform us about which elements are contained in a case or how cases are related. Moreover we want to reason about the elements of a concrete system Σ and not about all propositions of logics. Our atomic propositions are now the conditions of Σ, these are interpreted in appropriate models, i.e. we have a kind of "closed world assumption". According to [OL] every net has its own logic, an endogenous one.

What we can do is to extend standard axiomatics, e.g. $\mathcal{L}$-axiomatics, by axioms reflecting suitable properties of the system. For example the progress formula of an event

e of a system Σ, [Reil], $\wedge \bullet e \rightarrow \bigcirc \wedge e\bullet$ can be derived only by specific axiomatics characterizing Σ. Such specific axiomatics are extensions of standard ones by axioms describing the characteristic features of nets, [Sin].

If we work with event structures, [LT,Pe], we do not need specific axioms, because the structure of events, which is very abstract, is well described by $\mathcal{L}$-axiomatics extended by axioms expressing properties of conflicts. But all information carried out by conditions is lost.

I would say that we have a great variety of axiomatics to work with and to apply to net theory. I have only referred to the differences between branching time and partial order logics, however Propositional Dynamic Logic, [Tu], describing linear orders, and the proof-theoretic logics of [Av] are also applicable to net theory, [GG]. And we have not yet started to work on axiomatics for PrT nets! There is a lot to be done!

References

[Av] Avron A.: The Semantics and Proof Theory of Linear Logic. TCS 57, 1988.

[BPM] BenAri,Pnueli,Manna: The Temporal Logic of Branching Time. Acta Informatica 20, 1983.

[ES] Emerson,Srinivasan: Branching Time Temporal Logic. Proceedings of the REX School/Workshop 1988 on Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, LNCS 354, Springer-Verlag, 1989.

[GG] Gunther,Gehlot: Nets as Tensor Theories. Proceedings of the 10 International Conference on Application and Theory of Petri Nets, Bonn, FRG, June 1989.

[Kr] Kröger F.: Temporal Logic of Programs. EATS Monographs on Theoretical Computer Science, Vol. 8, Springer-Verlag, 1987.

[La] Lambort L.: What Good is Temporal Logic? Information Processing 83, R.E.A. Mason(Ed), North-Holland, IFIP 1983.

[LT] Lodaya,Thiagarajan: A Modal Logic for a Subclass of Event Structures, 14 ICALP, LNCS 267, Springer-Verlag, 1987.

[OL] Owicki,Lambort: Proving Liveness Properties of Concurrent Programs. ACM, Vol. 4, No 3, July 1982.

[Pe] Penczek W.: A Temporal Logic for Event Structures. Fundamenta Informaticae 11, 1988.

[PW] Pinter,Wolper: A Temporal Logic for Reasoning about Partially Ordered Computations. Proceedings of the third ACM Symposium on Principles of Distributed Computing, Vancouver, Canada, 1984.

[RU] Rescher,Urquhart: Temporal Logic. Springer Wien/New York, 1971.

[Rei1] Reisig W.: Towards a Temporal Logic for True Concurrency. Part I: Linear Time Propositional Logic. Arbeitspapiere der GMD 277, 1987.

[Rei2] Reisig W.: Towards a Temporal Logic for Causality and Choice in Distributed Systems. Proceedings of the REX School/Workshop on Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, June 1988, LNCS 354, Springer-Verlag 1989.

[Sin] Sinachopoulos A.: Temporal Logics for Elementary Net Systems. Arbeitspapiere der GMD 353, 1988.

[Sti] Stirling C.: Expressibility and Definability in Branching and Linear Time Temporal Logics. Department of Computer Science, University of Edinburgh, 1988.

[Str] Strack V.: Occurrence Structures: A Model of Concurrency and Choice in Systems Behaviour. Arbeitspapiere der GMD 328, 1988.

[Tu] Tuominen H.: EN-Systems and Dynamic Logic. 9th European Workshop on Application and Theory of Petri Nets, Venice, Italy, June 1988.

[W] Winskel G.: Event Structures. LNCS 255, 1987

AN ALGEBRAIC WAY OF DEFINING THE BEHAVIOURS OF PLACE/TRANSITION PETRI NETS

Józef Winkowski
Instytut Podstaw Informatyki PAN
00-901 Warszawa, PKiN, Skr.p.22, Poland

A b s t r a c t. The concept of a marked Place/Transition Petri net is generalized by introducing a concept of a seminet and some operations are defined which allow constructing seminets from seminets. A structure which represents a behaviour is assigned to each seminet such that the behaviour of a compound seminet can be obtained from the behaviours of components.

K e y w o r d s: Petri net, marking, seminet,configuration, behaviour, composition, compositionality.

1. INTRODUCTION

In this note we present a method of defining the behaviours of finite marked Place/Transition Petri nets with places of infinite capacity (P/T nets).

P/T nets can be defined by showing how they consist of simpler parts, called seminets. The latter differ from complete P/T nets in that they may have dangling links, that is edges without source or without target node. A P/T net can be obtained from seminets which represent its parts by combining each outgoing link of each part with the corresponding incoming link of the corresponding part.

The behaviours of P/T nets and their parts are represented by exhibiting how tokens constituting markings are absorbed and created. This is done with the aid of systems analogous to the families of finite configurations of labelled event structures (cf. [5-8]). Such systems contain information similar to that in representations of behaviours of P/T nets by suitable unfoldings (cf. [2] and [7]). In particular, they are related to the formal models of executions of P/T nets in [1] and [4].

The method of defining the behaviours of P/T nets and their parts is compositional in the sense that the behaviour of a complex seminet can be obtained by decomposing the seminet into parts with known behaviours and by composing the behaviours of parts with the aid of suitable operations. At this point it is similar to that in [3], [7], and [8]. However, our approach is essentially different from those in [3], [7], and [8]. Unlike in [7] and [8], where compound nets are constructed from simpler ones with the aid of concepts of category theory, we build nets in a manner which reflects and exploits their graphical representation. Unlike in [3], where compound nets are constructed from simpler ones by combining transitions, we build nets from seminets by combining dangling links.

Working with seminets allows us to deal not only with P/T nets, but also with incomplete parts of P/T nets. This may be important since such parts are often particularly well suited for modelling open systems which communicate with their environments.

2. SEMINETS

Seminets are structures similar to P/T nets but more general in the sense that they may have dangling links. We declare them to be of some sorts, where a sort specifies how the elements of a seminet are partitioned into subsets depending on their roles (dangling links to places, to transitions, from places, from transitions, and internal links and other elements).

For a formalization, let **Places**, **Transitions**, and **Links**, be fixed nonempty universes (of places, transitions, and links, respectively).

2.1. Definition. An S-sort (over **Places**, **Transitions**, and **Links**) is a quintuple $\alpha = (A_\alpha, B_\alpha, C_\alpha, D_\alpha, E_\alpha)$ of mutually disjoint subsets of **Places** $\cup$ **Transitions** $\cup$ **Links** such that $A_\alpha, B_\alpha, C_\alpha, D_\alpha$ are contained in **Links** and E_α contains at least one element from **Places** $\cup$ **Transitions**. For such an S-sort we define $L_\alpha = A_\alpha \cup B_\alpha \cup C_\alpha \cup D_\alpha \cup E_\alpha$. ∎

2.2. Definition. Given an S-sort α, a (finite) seminet of the sort α is $N = (P_N, T_N, Z_N, in_N, out_N, \mu_N)$, where:

(1) P_N, T_N, and Z_N, are mutually disjoint finite subsets of **Places**, **Transitions**, and **Links**, respectively, such that $P_N \cup T_N \cup Z_N \subseteq L_\alpha$ and $P_N \cup T_N \neq \emptyset$,

(2) in_N and out_N are functions assigning to each $x \in P_N \cup T_N$ a subset $in_N(x) \subseteq Z_N$ (of incoming links) and a subset $out_N(x) \subseteq Z_N$ (of outgoing links), respectively, such that:

(2.1) $Z_N = \bigcup \langle in_N(x) \cup out_N(x) \colon x \in P_N \cup T_N \rangle$,

(2.2) for each $z \in Z_N$ there exists at most one $x \in P_N \cup T_N$

with $z \in in_N(x)$ and at most one $y \in P_N \cup T_N$ with $z \in out_N(y)$,

(2.3) for each $z \in Z_N$, if $z \in in_N(x)$ and $z \in out_N(y)$ then either $x \in P_N$ and $y \in T_N$ or $x \in T_N$ and $y \in P_N$,

(2.4) $\bigcup \langle in_N(p): p \in P_N\rangle - \bigcup \langle out_N(t): t \in T_N\rangle = A_\alpha$,
$\bigcup \langle in_N(t): t \in T_N\rangle - \bigcup \langle out_N(p): p \in P_N\rangle = B_\alpha$,
$\bigcup \langle out_N(p): p \in P_N\rangle - \bigcup \langle in_N(t): t \in T_N\rangle = C_\alpha$,
$\bigcup \langle out_N(t): t \in T_N\rangle - \bigcup \langle in_N(p): p \in P_N\rangle = D_\alpha$,
$\bigcup \langle in_N(x) \cap out_N(y):\ x,y \in P_N \cup T_N\rangle \cup P_N \cup T_N = E_\alpha$,

(3) μ_N is a function (a marking) which assigns to each $p \in P_N$ a natural number (of tokens).

For such a seminet N we define F_N (the flow relation of N) by

$(x,y) \in F_N$ iff $x,y \in P_N \cup T_N$ and $out_N(x) \cap in_N(y) \neq \emptyset$.

The universe of seminets of a sort α is written as $\mathbf{Seminets}_\alpha$. ∎

Examples of seminets are shown in fig. 2.1.

Seminets of sorts satisfying suitable conditions can be composed into seminets of a resulting sort. The composition is an operation $(M,N) \longmapsto M|N$ which "glues" two seminets M and N at their common links provided that the respective pairs of such links are consistent in the sense that one element is an outgoing link from a place or a transition in one seminet and the other is an incoming link to a transition or a place, respectively, in the other seminet (cf. fig. 2.1). A formal definition can be given by the following obvious proposition.

2.3. Proposition. Let α and β be arbitrary S-sorts such that $L_\alpha \cap L_\beta \subseteq (A_\alpha \cap D_\beta) \cup (B_\alpha \cap C_\beta) \cup (C_\alpha \cap B_\beta) \cup (D_\alpha \cap A_\beta)$. Then
$\alpha \,\&\, \beta = ((A_\alpha - D_\beta) \cup (A_\beta - D_\alpha), (B_\alpha - C_\beta) \cup (B_\beta - C_\alpha), (C_\alpha - B_\beta) \cup (C_\beta - B_\alpha),$
$(D_\alpha - A_\beta) \cup (D_\beta - A_\alpha), E_\alpha \cup E_\beta \cup (A_\alpha \cap D_\beta) \cup (B_\alpha \cap C_\beta) \cup (C_\alpha \cap B_\beta) \cup (D_\alpha \cap A_\beta))$
is an S-sort and for any $M \in \mathbf{Seminets}_\alpha$ and $N \in \mathbf{Seminets}_\beta$ we have a unique $Q \in \mathbf{Seminets}_{\alpha \& \beta}$, called the composition of M and N and written as $M|N$, such that:

(1) $P_Q = P_M \cup P_N$, $T_Q = T_M \cup T_N$, $Z_Q = Z_M \cup Z_N$,

(2) $in_Q = in_M \cup in_N$, $out_Q = out_M \cup out_N$,

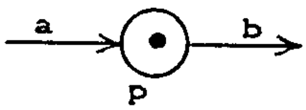
(3) $\mu_Q = \mu_M \cup \mu_N$. ∎

2.4. Definition. A one-place atomic seminet with a place p, incoming links $a_1, \ldots, a_m$, outgoing links $b_1, \ldots, b_n$, and k tokens residing in its place, is $N \in \mathbf{Seminets}_\alpha$, where

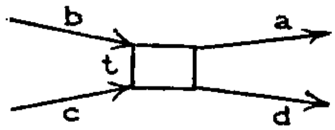
$\alpha = (\langle a_1,\ldots,a_m\rangle,\emptyset,\langle b_1,\ldots,b_n\rangle,\emptyset,\langle p\rangle)$, $P_N = \langle p\rangle$, $T_N = \emptyset$, $Z_N = \langle a_1,\ldots,a_m,b_1,\ldots,b_n\rangle$, $in_N(p) = \langle a_1,\ldots,a_m\rangle$, $out_N(p) = \langle b_1,\ldots,b_n\rangle$, and $\mu_N(p) = k$. A *one-transition atomic seminet* with a transition t, incoming links $a_1,\ldots,a_m$, and outgoing links $b_1,\ldots,b_n$, is $N \in Seminets_\alpha$, where $\alpha = (\emptyset,\langle a_1,\ldots,a_m\rangle,\emptyset,\langle b_1,\ldots,b_n\rangle,\langle t\rangle)$, $P_N = \emptyset$, $T_N = \langle t\rangle$, $Z_N = \langle a_1,\ldots,a_m,b_1,\ldots,b_n\rangle$, $in_N(t) = \langle a_1,\ldots,a_m\rangle$, $out_N(t) = \langle b_1,\ldots,b_n\rangle$, and $\mu_N = \emptyset$. An *atomic seminet* is a one-place or a one-transition atomic seminet. ▌

By induction on the size of a seminet one easily obtains the following result.

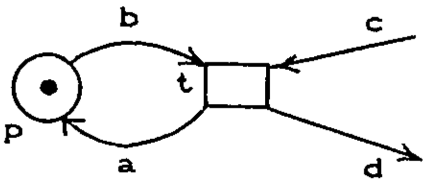
2.5. Theorem. Each seminet can be obtained by composing atomic seminets. ▌



A one-place atomic seminet M of the sort $(\langle a\rangle,\emptyset,\langle b\rangle,\emptyset,\langle p\rangle)$



A one-transition atomic seminet N of the sort $(\emptyset,\langle b,c\rangle,\emptyset,\langle a,d\rangle,\langle t\rangle)$



The composition $M|N$
(of the sort $(\emptyset,\langle c\rangle,\emptyset,\langle d\rangle,\langle p,t,a,b\rangle)$)

Fig. 2.1

3. BEHAVIOURS

The behaviours of seminets may be represented by the respective flows of tokens along links, both internal and dangling ones. They can be obtained by defining in a natural way the behaviours of one-place and one-transition atomic seminets and by combining such behaviours in a manner corresponding to that of constructing the respective seminets. To this end we introduce first a general concept of a behaviour and suitable operations on behaviours.

A behaviour is represented by the set of its possible states, each state represented by the set of events which have occurred, where an event is an execution of an action: either receiving a token from an incoming link, or sending a token to an outgoing link, or a transfer of a token along an internal link. Formally, each event may be regarded as a pair consisting of the respective execution symbol and action symbol, each state may be regarded as the respective configuration (set) of such pairs, and the entire behaviour may be regarded as the respective system of configurations, the latter satisfying suitable conditions. For action symbols one may choose names of the corresponding links. The respective sets of action symbols constitute what we call a B-sort.

3.1. Definition. A <u>B-sort</u> (over **Links**) is a triple $\xi = (I_\xi, J_\xi, K_\xi)$ of mutually disjoint subsets of **Links**. For such a B-sort we define $H_\xi = I_\xi \cup J_\xi \cup K_\xi$. ▌

3.2. Definition. Given a B-sort ξ, a <u>behaviour</u> of the sort ξ is a nonempty set S of finite functions such that:

(1) $\bigcup S$ is a function with values in H_ξ,

(2) $\bigcap T \in S$ for each nonempty $T \subseteq S$ which is <u>bounded</u> in S in the sense that there exists $s \in S$ such that $t \subseteq s$ for all $t \in T$,

(3) $\bigcup T \in S$ for each $T \subseteq S$ which is bounded in S.

We call all $s \in S$ <u>states</u> or <u>configurations</u> of S and, for each state $s \in S$ and each $(x,v) \in s$, we call (x,v) an <u>event</u> with the <u>execution symbol</u> x and <u>action symbol</u> v. The universe of behaviours of a sort ξ is written as $\mathbf{Behaviours}_\xi$. ▌

Examples of behaviours are given in 3.5 and 3.6.

In the sequel we do not want to distinguish between behaviours which are isomorphic in the following sense.

3.3. Definition. An <u>isomorphism</u> from $S \in \mathbf{Behaviours}_\xi$ to $T \in \mathbf{Behaviours}_\xi$ is a bijection $b: \bigcup S \longrightarrow \bigcup T$ such that, for all $(x,v),(y,w),s,t$:

(1) $(y,w) = b((x,v))$ implies $v = w$,

(2) $s \in S$ implies $b(s) \in T$,

(3) $t \in T$ implies $b^{-1}(t) \in S$.

If such an isomorphism exists then we say that the behaviours S and T are <u>isomorphic</u>. The class of behaviours which are isomorphic to a given behaviour S is written as $[S]$. ▌

Behaviours of sorts satisfying suitable conditions can be composed into behaviours of a resulting sort. The composition is an operation $(S,T) \longmapsto S \parallel T$ which combines certain pairs of configurations $s \in S$ and $t \in T$ into configurations of a resulting behaviour U. Each such a combination is obtained by taking a disjoint union of s and t and by coupling in this union in a consistent manner pairs of events: an event of sending a token to a common link, and a corresponding event of receiving a token from such a common link (cf. fig. 3.1). A formal definition can be given by the following proposition.

3.4. Proposition. Let ξ and η be B-sorts such that $H_\xi \cap H_\eta \subseteq (I_\xi \cap J_\eta) \cup (J_\xi \cap I_\eta)$. Then $\xi \,\$\, \eta$ defined by
$\xi \,\$\, \eta = ((I_\xi - J_\eta) \cup (I_\eta - J_\xi), (J_\xi - I_\eta) \cup (J_\eta - I_\xi), K_\xi \cup K_\eta \cup (I_\xi \cap J_\eta) \cup (J_\xi \cap I_\eta))$
is a B-sort and for any $S \in \mathbf{Behaviours}_\xi$ and $T \in \mathbf{Behaviours}_\eta$ we have a unique $U \in \mathbf{Behaviours}_{\xi \$ \eta}$, called the <u>composition</u> of S and T and written as $S \parallel T$, such that:
$u \in U$ iff u <u>consists</u> of some $s \in S$ and $t \in T$ in the sense that
$u = \{((0,x),h):\ (x,h) \in s - c^{-1}(t)\} \cup \{((1,y),h):\ (y,h) \in t - c(s)\}$
$\quad \cup \{(((0,x),(1,y)),h):\ ((x,h),(y,h)) \in c\}$
for a bijection $c: \{(x,h) \in s:\ h \in H_\xi \cap H_\eta\} \longrightarrow \{(y,h) \in t:\ h \in H_\xi \cap H_\eta\}$ which satisfies:

(1) for all (x,h) and (y,h'), $(y,h') = c((x,h))$ implies $h' = h$,

(2) if $e,f \in s$ are separated by some $s' \in S$ with $s' \subseteq s$ in the sense that either $e \in s'$ and $f \notin s'$ or $e \notin s'$ and $f \in s'$ then e,f can be separated by some $s'' \in S$ with $s'' \subseteq s$ such that there exists $t'' \in T$ with $t'' \subseteq t$ for which $c(s'') \subseteq t''$ and $c^{-1}(t'') \subseteq s''$; similarly for $e,f \in t$.

If S' and T' are behaviours which are isomorphic to S and T, respectively, then the behaviour $S' \parallel T'$ is isomorphic to $S \parallel T$. ▌

Proof outline:

For a nonempty $R \subseteq U$ with an upper bound $u \in U$ which consists of some $s \in S$ and $t \in T$ with a corresponding bijection c we have nonempty $R_S \subseteq S$ and $R_T \subseteq T$ such that each $r \in R$ consists of some

$r_S \in R_S$ and $r_T \in R_T$ with a bijection c_r. From the definition of U we obtain that each c_r is the restriction of c to r_S. From the fact that S and T are behaviours and that R_S and R_T are bounded we obtain that $\bigcap R_S \in S$ and $\bigcap R_T \in T$. On the other hand, $\bigcap R$ consists of $\bigcap R_S$ and $\bigcap R_T$ with $\bigcap \{c_r: r \in R\}$. Hence $\bigcap R \in U$, as required.

The property $\bigcup R \in U$ for bounded $R \subseteq U$ can be shown in a similar way. ▌

The behaviour of a one-place atomic seminet is defined as consisting of independent behaviours of tokens which can possibly appear in the respective place. A configuration of such a behaviour consists of mutually disjoint configurations of tokens belonging to three finite sets: a set X of tokens which resided at the beginning and have already been emitted, a set Y of tokens which have been received but not yet emitted, and a set W of tokens which have been both received and emitted. This is stated in the following obvious proposition.

3.5. Proposition. For each S-sort of the form $\alpha = (\{a_1,\ldots,a_m\},\emptyset,\{b_1,\ldots,b_n\},\emptyset,\{p\})$ and each one-place atomic seminet $N \in \mathbf{Seminets}_\alpha$ with k tokens in its place p we have a B-sort ξ with $I_\xi = \{a_1,\ldots,a_m\}$, $J_\xi = \{b_1,\ldots,b_n\}$, $K_\xi = \emptyset$, and a behaviour $S \in \mathbf{Behaviours}_\xi$, called the behaviour of N and written as beh_N, such that:

$s \in S$ iff $s: X \cup Y \cup W \cup (\{0\}\times W) \longrightarrow H_\xi$, where:

(1) $X \subseteq \{1,\ldots,k\}$,

(2) Y, W are finite disjoint subsets of the set $\{k+1,k+2,\ldots\}$,

(3) $s(X) \subseteq J_\xi$, $s(Y) \subseteq I_\xi$, $s(W) \subseteq I_\xi$, $s(\{0\}\times W) \subseteq J_\xi$. ▌

The behaviour of a one-transition atomic seminet is defined as consisting of independent executions of the respective transition, each execution consisting of coincident events of receiving tokens from incoming links and sending tokens to outgoing links.
A configuration of such a behaviour consists of mutually disjoint configurations of executions belonging to a finite set X. This is stated in the following obvious proposition.

3.6. Proposition. For each S-sort of the form $\alpha = (\emptyset, \langle a_1, \ldots, a_m \rangle, \emptyset, \langle b_1, \ldots, b_n \rangle, \langle t \rangle)$ and each one-transition atomic seminet $N \in \mathbf{Seminets}_\alpha$ we have a B-sort ξ with
$I_\xi = \langle a_1, \ldots, a_m \rangle$, $J_\xi = \langle b_1, \ldots, b_n \rangle$, $K_\xi = \emptyset$,
and a behaviour $S \in \mathbf{Behaviours}_\xi$, called the behaviour of N and written as $\mathbf{beh}_N$, such that:
$s \in S$ iff $s: \langle 1, \ldots, m, m+1, \ldots, m+n \rangle \times X \longrightarrow H_\xi$, where:
(1) X is a finite subset of the set $\langle 1, 2, \ldots \rangle$,
(2) $s(p,x) = a_p$ for $p \in \langle 1, \ldots, m \rangle$ and $x \in X$,
(3) $s(q,x) = b_{q-m}$ for $q \in \langle m+1, \ldots, m+n \rangle$. ▌

The behaviour of any seminet is defined by representing such a seminet as a result of composing atomic seminets and by composing correspondingly the behaviours of the respective atomic seminets (cf. fig. 3.1). In fact, this idea applies to seminets and isomorphism classes of behaviours rather than to seminets and concrete behaviours. The respective result is as follows.

3.7. Theorem. There exists a unique correspondence Γ between seminets and isomorphism classes of behaviours such that:
(1) for each seminet $N \in \mathbf{Seminets}_\alpha$, $\Gamma(N) \subseteq \mathbf{Behaviours}_{\mathbf{bs}(\alpha)}$, where $\mathbf{bs}(\alpha) = (A_\alpha \cup B_\alpha, C_\alpha \cup D_\alpha, E_\alpha)$,
(2) for each one-place or one-transition atomic seminet N, $\Gamma(N) = [\mathbf{beh}_N]$,
(3) for all α and β satisfying the requirement of 2.3, $\xi = \mathbf{bs}(\alpha)$ and $\eta = \mathbf{bs}(\beta)$ satisfy the requirement of 3.4 and, for all $M \in \mathbf{Seminets}_\alpha$ and $N \in \mathbf{Seminets}_\beta$, $\Gamma(M|N) = \Gamma(M) \parallel \Gamma(N)$. ▌

Proof outline:

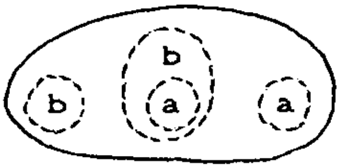
According to 2.5, each seminet N can be represented as a result of combining one-place and one-transition atomic seminets. Each such a representation is given by an expression e which is built in the standard way of symbols of one-place and one-transition atomic seminets and the symbol of composition. Thus we have for N a set $\varepsilon(N)$ of representing expressions.

In the set of expressions representing seminets we have a least equivalence $\equiv$ which satisfies the following two conditions:
(a) $e_0|e_1 \equiv e_1|e_0$,
(b) $(e_0|e_1)|e_2 \equiv e_0|(e_1|e_2)$.
This equivalence holds for e_0 and e_1 iff e_0 and e_1 belong to $\varepsilon(N)$

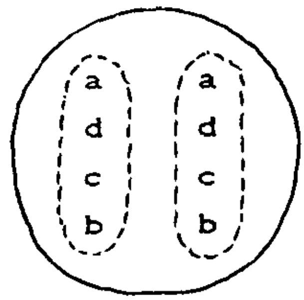
for a seminet N.

Assuming some order of expressions representing one-place and one-transition atomic seminets, we obtain a unique canonical representing expression $e(N) \in \varepsilon(N)$ for each seminet N.

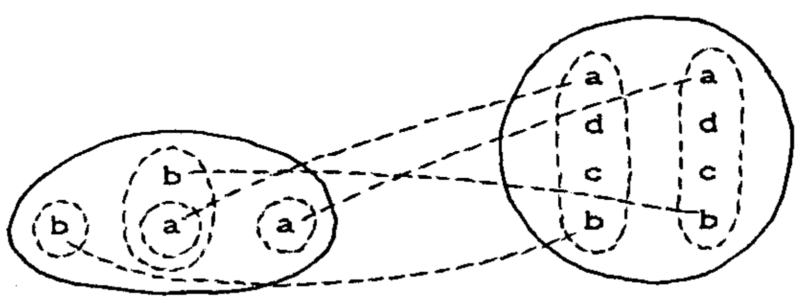
Now, given a seminet N, we consider $e(N)$ and, taking into account the expected properties of Γ, we define $\Gamma(N)$ by induction on the structure of $e(N)$. As the equivalence of behaviours up to isomorphism satisfies conditions similar to those defining the equivalence $\equiv$, the correspondence thus obtained is as required. ▌



A configuration s in $\mathbf{beh}_M$ for a one-place atomic seminet M with incoming link a, outgoing link b, and one token



A configuration t in $\mathbf{beh}_N$ for a one-transition atomic seminet N with incoming links b,c, and outgoing links a,d



Combining s and t into a configuration of $\mathbf{beh}_M \| \mathbf{beh}_N$

(attention: events are represented as occurrences of the respective action symbols in the figure)

Fig. 3.1

The semantics for seminets given by 3.7 agrees with the one given by a suitable extension of the firing rule of P/T nets. This can be formulated precisely as follows and proved in a standard way as a similar result in [5].

3.8. Definition. A <u>step</u> is a pair $M \Rightarrow N$ with $M,N \in \mathbf{Seminets}_\alpha$ such that $P_N = P_M$, $T_N = T_M$, $Z_N = Z_M$, $\mathbf{in}_N = \mathbf{in}_M$, $\mathbf{out}_N = \mathbf{out}_M$, and μ_M and μ_N are related in one of the following three ways:

(1) there exists $t \in T_M$ such that $\mu_M(p) > 0$ for all $p \in P_M$ with $(p,t) \in F_M$ and

$$\mu_N(x) = \begin{array}{ll} \mu_M(x)-1 & \text{for } x \in P_N \text{ with } (x,t) \in F_M \text{ and } (t,x) \notin F_M \\ \mu_M(x)+1 & \text{for } x \in P_N \text{ with } (t,x) \in F_M \text{ and } (x,t) \notin F_M \\ \mu_M(x) & \text{for } x \in P_N \text{ with } (x,t) \in F_M \text{ and } (t,x) \in F_M \\ & \quad\text{or with } (x,t) \notin F_M \text{ and } (t,x) \notin F_M, \end{array}$$

(2) there exists $p \in P_M$ such that $in_M(p) \neq \emptyset$, $\mu_N(p) = \mu_M(p)+1$, and $\mu_N(x) = \mu_M(x)$ for $x \in P_N-\{p\}$,

(3) there exists $p \in P_M$ such that $\mu_M(p) > 0$, $out_M(p) \neq \emptyset$, $\mu_N(p) = \mu_M(p)-1$, and $\mu_N(x) = \mu_M(x)$ for $x \in P_N-\{p\}$. ▌

3.9. Proposition. For each $S \in Behaviours_\xi$ and each $s \in S$ we have a unique $T \in Behaviours_\xi$, called a <u>continuation</u> of S from s and written as $S-s$, such that:

$t \in T$ iff $t = r-s$ for some $r \in S$ with $s \leq r$. ▌

3.10. Theorem. For each step $M \Rightarrow N$ and each behaviour $S \in \Gamma(M)$ there exists a minimal nonempty configuration $s \in S$ such that $S-s \in \Gamma(N)$. For each behaviour $S \in \Gamma(M)$ and each minimal and nonempty configuration $s \in S$ there exists a step $M \Rightarrow N$ such that $S-s \in \Gamma(N)$. ▌

REFERENCES

1. E. Best and R. Devillers: Sequential and Concurrent Behaviour in Petri Net Theory, TCS 55, 1987, 87-136

2. U. Goltz and W. Reisig: The Non-Sequential Behaviour of Petri Nets, Information and Control 57, 2-3, 1983, 125-147

3. A. Mazurkiewicz: Compositional Semantics of Pure Place/Transition Systems, Fundamenta Informaticae XI, 1988, 331-356

4. J. Meseguer and U. Montanari: Petri Nets Are Monoids: A New Algebraic Foundation for Net Theory, Proc. LICS, 1988

5. J. Winkowski: Event Structure Representation of the Behaviour of Place/Transition Systems, Fundamenta Informaticae XI, 1988, 405-432

6. J. Winkowski: An Equivalence of Communicating Processes in Distributed Environments, Fundamenta Informaticae XII, 1989, 97-128

7. G. Winskel: Event Structures, Springer LNCS 255, 1987, 325-392

8. G. Winskel: Petri Nets, Algebras, Morphisms and Compositionality, Information and Comput. 72, 1987, 197-238

Reprints

Reprinted From The Chinese Petri Nets Newsletter:

Petri网通讯

第 1 期

On Technical Safety and Security

C. A. Petri

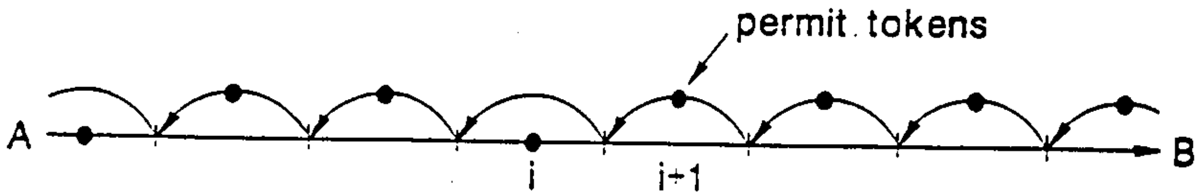
March 28, 1989

When the first railway trains were put into operation, a man (an quan yuan) had to walk ahead of each train, waving a small red flag and crying "attention! a train!". I laughed heartily when I heared of this for the first time – until it occurred to me that railway traffic was safer at that time than it is now.

Let us study the task of the "safety official" by means of the combinatorics of physical signals, by drawing graphs and nets on which trains and signals may move, represented by small pebbles or "tokens" which denote the presence of trains or signals along a track.

We start with a long one-way track for trains, from A-jing to B-jing:

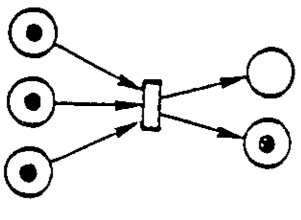
Train #n might have to stop because a cow is grazing between the rails, or because it has gone out of fuel, or for some other unforeseen reason. Train #n+1 has to be warned in time so that it does not crash into #n. Now a signal which carries the warning might be delayed, also because of an obstacle, or by losing some of its energy. Therefore, warning signals which say "stop as soon as you can" are not a reliable means to prevent accidents. Light signals can be absorbed by mist, etc. Rather, it is more appropriate to the spirit of safety to employ signals not for warning, but for permission. When a train leaves a segment of the track, a signal which permits entry into that segment is put on its way to the next train. Let us describe trains as dots (tokens) moving on a line, and let us segment the line into pieces which are longer than the longest train. No train may enter a segment without the presence of a token of permission:



If our graphical (mathematical) model is correct, the trains (and the signals!) are now *safe* in a very precise sense: if every train, in *transition* from segment i to segment i+1, shifts a permit token from the signalling arc i+1 into the arc belonging to segment i, then the number of tokens on each *basic circuit* remains unchanged by all occurrences of

transitions; if this number was one token per basic circuit at the time of construction, it remains 1 as long as the circuit exists, if the Rule of The Token Game is obeyed: A transition of tokens may occur if *all* arcs which point to the transition carry a token; and by the occurrence of a transition, precisely 1 token is removed from the input arcs of the transition, while precisely one token is added to each output arc of the transition.

Then, in our example, there can never be two tokens on the same arc. A marked graph or net with this property is called "safe": it can never happen that all the pre-conditions of a transition are fulfilled while a token is present on one of its output arcs. In a safe net, no situation of *contact* may occur: contact is, as in all traffic, the situation immediately before a crash. In the notation of net theory:



is a <u>contact</u> situation

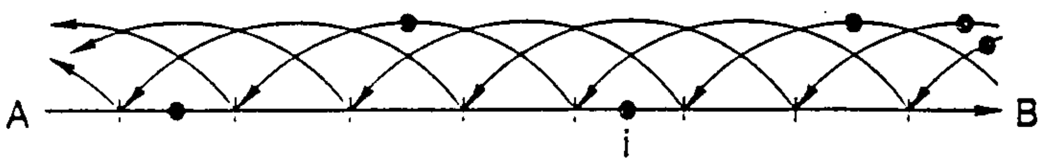
is a contact situation. So in our construction, we have made the net safe by adding a reverse signalling arc to each segment of the track, and putting a signal token (train or permit) onto each basic circuit. The net is safe, the tokens are safe; but are the trains safe?

By no means: our model is not correct in two respects:

1. Real trains have non-zero length. When a train has just entered a segment, it has not yet left the previous segment. Therefore we have to reserve at least two segments for each train, even if every segment is longer than each train.

2. Real trains have non-zero mass. When a moving train finds no permission to enter a segment, it will enter that segment nevertheless; therefore a certain number of segments ahead of the token which represents the signal-transmission-point of the train has to be reserved for the train.

It follows that we have to change the signalling structure in such a way that a (fixed) number of successive segments are reserved for each train: of course, in determining this number, the maximal distance needed for a train to come to a full stop is to be added to the maximal length of the previous train; it is necessary and sufficient to express the result in a number of successive segments to be reserved for each train. Only now it is correct to represent a train by a token. Example:



The token in segment i is now the only token on three (in this example) basic circuits, which overlap in segment i of the track. We had to draw three backward-signalling lines in order to respect length and mass of real trains. The marked graph is safe, and the tokens on the "track" A — B can now represent real trains.

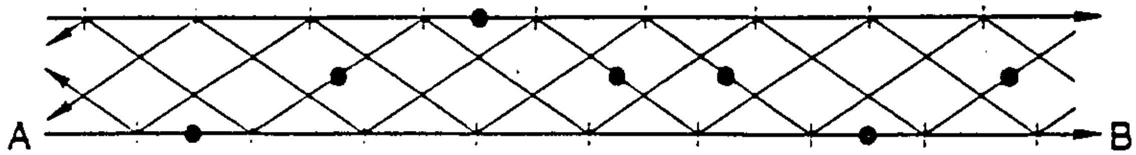
The question is now: can the tokens on the signalling lines represent real signals?

No! Our model is still incorrect in one respect:
Real signals have non-zero length, just like trains (they have to be identifiable as permits).

It follows that we have to reserve for each token which is to represent a real signal at least two consecutive segments of its signalling line. How can we achieve this? In the picture above, only by making sure that consecutive trains have never more than five empty segments between their tokens. That means that the train at i must not reach B before the next train (still near A) has moved.

This might be achieved in two equivalent ways: by coupling consecutive trains by a chain which is short enough, or by introducing additional forward signalling arcs which are long enough. Both ways, we would introduce the absurd constraint that a train would have to stop because the *next* train has to! When we construct a transmission line, we do not want to construct an obligation to transmit; economy requires that trains should be independent (not coupled) so that in times of low demand the schedule can be cut down freely.

In our last picture above, the trains can already move independently with the sole constraint of sufficient separation already satisfied. Only the signals are not sufficiently separated. Here is the simplest supplement to our last construction which fulfils all requirements for trains *and* signals:



By playing the token game, we can easily verify that:

1. Between train tokens, there are always at least two empty segments (one for length, one for mass, i.e. for halting when required.)
2. Between signal tokens, there is always at least one empty segment (for length).
3. The trains are independent (the track A → B may be made empty). The marked graph is safe, and no contact can occur, because every segment belongs to a circuit with one token only, i.e. to a basic circuit.
4. On the only forward signalling line (on top of the picture), signals move "in parallel" to the trains: they are not entry permits for trains. Between two of these signals there will always be a train, and between two trains there will always be a "security token" on the top line.

To understand this and to prove it, we must see that our new construction has an additional property which goes beyond safety, and which we call "*security*". The essence of security is, for synchronization graphs, that every *pair of consecutive arcs* belongs to a basic circuit.

This is a requirement which is widely unknown, which in fact looks strange when we see what it means for the relation between trains and permits: trains must be separated from all permits just like trains must be separated from trains and permits from permits!

Yet this is an indispensable requirement of security. Why?

The apparent paradox disappears if we consider transmission lines not for trains, but for electronic messages. These are not different in nature from their corresponding permits and security signals, and the requirement then reads in fact immediately: keep all signals sufficiently separated. We know that *this* is necessary; we are just not accustomed to treat trains like "signals with mass". Remember that we had to respect the mass of trains by *additional* separation between trains; after we have taken care of this, trains *must* be treated like signals in the question of minimal separation (one segment for "length").

Formal proof of security is certainly of high practical value. Two ingredients are necessary:

1. The mathematical model of the real world system must be realistic.
2. It must have the formal property called "security" here.

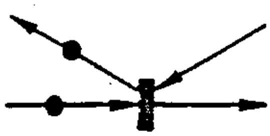
For the latter, we define the notion of "transjunction", short for "con*junction* of conditions across a *tran*sition":

Definition: A transition t is in transjunction in a case c if and only if some input condition and some output condition of t are holding in the case c.

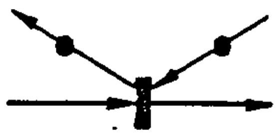
Formally:

$$\text{Transjunction } (t,c) :\Longleftrightarrow {}^{\bullet}t \cap c \neq \emptyset \ \underline{\text{and}}\ t^{\bullet} \cap c \neq \emptyset$$

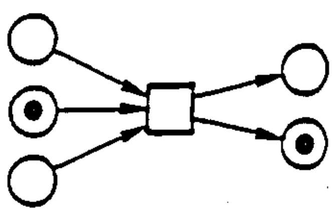
In our pictorial descriptions, examples of transjunction are, in synchronization graphs:



and



in nets more generally:



Finally, we give the

Definition:

An elementary net system or a synchronization graph is secure if and only if it has in no case contact or transjunction.

Formally:

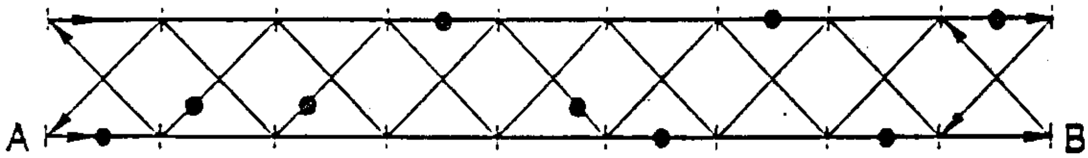
$$\text{Secure}(S,T;F,C) :\Longleftrightarrow$$

$$\bigwedge t \in T, \bigwedge c \in C:$$

$$\left.\begin{array}{l} {}^{\bullet}t \subseteq c \ \rightarrow\ t^{\bullet} \cap c = \emptyset \\ t^{\bullet} \subseteq c \ \rightarrow\ {}^{\bullet}t \cap c = \emptyset \end{array}\right\} \textit{no contact}$$

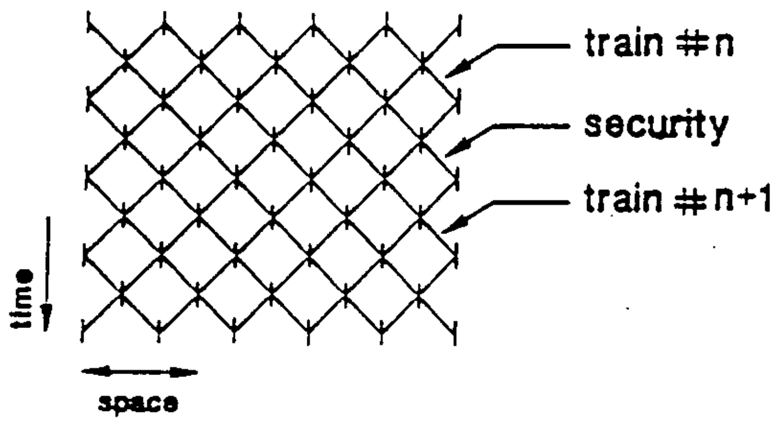
$$\left.\begin{array}{l} {}^{\bullet}t \cap c \neq \emptyset \ \rightarrow\ t^{\bullet} \cap c = \emptyset \\ t^{\bullet} \cap c \neq \emptyset \ \rightarrow\ {}^{\bullet}t \cap c = \emptyset \end{array}\right\} \textit{no transjunction}$$

We showed already the smallest secure transmission line for trains; for "massless trains" like electrical signals the smallest secure transmission line is simpler; it contains two backward-directed lines only.



As required, every transition belongs to *four* distinct basic circuits, because there are four pairings of input/output arcs that must be guarded against transjunction.

We should note that the occurrence graphs of all safe transmission lines shown above, and indeed of all regular secure transmission lines are *identical*, namely:



This insight helps greatly to construct formal proofs of security in general, and also to construct and understand regular secure signalling structures in more than one dimension; while a technical system may be more complex than the two we have considered here, the definition of security and the ingredients for obtaining it remain the same.

Yuan Chongyi helped me to prepare this paper.

Projects

ESPRIT Project No. 3148: DEMON
Design Methods Based on Nets

The ESPRIT Basic Research Project DEMON started in April 1989 and planned to end in September 1991, is part of the new EEC Programme for Basic Research in Information Technology. The objective of the project is to explore important theoretical issues involved in the formal reasoning about concurrent systems and to develop a formal framework supporting the design and verification of large concurrent and decentralized systems. Computer scientists from the institution listed below participate in this project.

The basis for this work is the Petri Net model which combines the benefits of a graphical system representation with a supporting formal theory which captures the essential locality of states, and action and is general in that it subsumes virtually all other formal models of concurrent systems.

The focus of the project is to enhance Petri net theory by a variety of concepts of modularity and composition required as means for designing concurrent systems: refinement and abstraction techniques, algebras and proof rules, appropriate notions of equivalence, Congruence and simulation, associated formal proof methods.

The work is organized in two strongly interrelated parts. The central part is concerned with the development of net classes exhibiting the above characteristics of modularity. The second part involves case studies, language studies and other activities supporting the development of suitable net classes; it also interfaces to related approaches and to other projects.

Please contact: Eike Best, ++49-2241-14-2786 or ++49-5121-883-741/0
email: eike@gmdzi.uucp or eike@infhil.uucp

Participating institutions and persons:

GMD	(E. Best, U. Goltz, A. Poigné, E. Smith)
LRI Univ. Paris-Sud	(E. Pelz, G. Vidal-Naquet, J. Vautherin, G. Berthelot, C. Froment, J. R. Fanchon, L. Petrucci)
DSI Univ. degli Studi di Milano	(F. de Cindio, C. Simone, L. Pomello, G. de Michelis, M. Ajmone-Massan, G. Mausi, N. Sabadini, L. Bernardinello)
TU München	(W. Brauer, W. Reisig, W. Vogler, V. Diekert, J. Desel, R. Gold)
Univ. Leiden	(G. Rozenberg, H. C. M. Kleijn, H. J. Hoogeboom, N. W. Keesmaat)
Univ. Passau	(M. Broy, T. Streicher)
Univ. of Newcastle upon Tyne	(R. P. Hopkins, M. Kountny, J. Hall)
Univ. Libre de Bruxelles	(R. Devillers, A. Sinachopoulos)
Univ. Zaragoza	(M. Silva, J. Esparza, J. Martinez, J. M. Colom)
Bull	(G. Memmi, Y. Souissi, J. F. Peyre)

Conference Announcements

ANNOUNCEMENT AND CALL FOR PAPERS

ELEVENTH INTERNATIONAL CONFERENCE ON APPLICATION AND THEORY OF

PETRI NETS

Wednesday 27 – Friday 29, June 1990

and

PETRI NETS TUTORIAL

Monday 25 – Tuesday 26, June 1990

PARIS, FRANCE

The Eleventh Annual International Petri Net Conference will be organized by the University of Paris VI and Bull, France. Papers presenting original contributions in any area of applications and theory of Petri nets are sought. The language of the conference is English.

TOPICS

System design and verification using nets
Causality/partial order theory of concurrency
Analysis and synthesis, structure and behaviour of nets
Net-based semantical, logical and algebraic calculi
Higher-level net models
Timed and stochastic nets
Relationships between net theory and other approaches
Symbolics net representations (graphical, textual, ...)
Computer tools
Experience with using nets, case studies
Educational issues

Applications of nets to:

Office automation
Flexible manufacturing systems
Programming languages
Protocols and interfaces
Hardware structures
Real-time systems
Performance evaluation
Operations research
Embedded systems

The conference takes place under the auspices of: AFCET SIG "Systèmes Parallèles et Distribués" and CNRS-C^3, AICA, BCS SIG "Formal Aspects of Computing Science", EATCS and GI SIG "Petri Nets and Related System Models".

PAPERS

Authors are invited to submit an *extended abstract* (approx. 10 pages) or a *full draft paper* (at most 30 pages) for the conference. The title page must contain a short abstract and a classification of the topics covered, preferrably using the list of topics above. The paper or extended abstract must clearly state the problem being addressed, the goal of the work, the results achieved and the relation to other work.

TOOLS, POSTERS AND PROJECTS

The conference will also comprise:

- An *exhibition of computer tools* for nets.
  Scheduled periods are set aside during the tutorial and conference for tool demonstrations.
- An *exhibition of posters* describing theoretical and practical results.
  Posters are displayed throughout the conference with a *scheduled period* for discussing them. Authors must submit a one page description of their poster.
- Short *presentations of projects* where nets are put into practice.
  This section of the conference allows the presentation of experiences of using nets in ongoing or completed projects. The presentation may be supplemented by a brief report in the proceedings. Authors must submit a 2 to 4 page outline of the project.

SUBMISSIONS

All four kinds of submissions (10 copies) must be received by the chairman of the program committee no later than January 15, 1990. Authors should clearly indicate the kind(s) of submission intended. Authors will be notified of acceptance/rejection by March 30, 1990. Final papers are due by May 14, 1990. The page limit will be 20 pages for papers and 10 pages for project presentations.

TUTORIAL

The tutorial will concentrate on the basic notions and fundamental concepts from the broad spectrum of Petri nets.

PROGRAM COMMITTEE CHAIRMAN

Prof. R. Valk
Fachbereich Informatik
Universität Hamburg
Rothenbaumchaussee 67/69
D-2000 Hamburg 13, FRG

ORGANIZING COMMITTEE CHAIRMEN

Prof. C. Girault
Laboratoire MASI
Université Pierre et Marie Curie
Tower 65
4, Place Jussieu
F-75252 Paris Cedex 05, France

PROGRAM COMMITTEE

G.F. Balbo, Italy
E. Best, FRG
J. Billington, Australia
G. Cutts, Great Britain
C. Girault, France
K. Jensen, Denmark
S. Kodama, Japan
V. Kotov, USSR
P. Lauer, Canada
G. Memmi, France
T. Murata, USA
L. Ojala, Finland
R. Shapiro, USA
M. Silva, Spain
C. Simone, Italy
P.S. Thiagarajan, India
R. Valk, FRG (Chairman)

INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING
IFIP-WG 10.3 CONCURRENT SYSTEMS

WORKING CONFERENCE on DECENTRALIZED SYSTEMS

Lyon (France), December 11-13, 1989

CALL FOR PAPERS

This conference will address recent issues in the implementation and evaluation of decentralized computer systems. It will be mainly focused on architectures and programming systems with no central point of control or communication.

List of topics (not limitative)

- Communication networks
- Non shared memory processors
- Distributed operating systems
- Distribution and communication concepts
- Dedicated programming languages
- Distributed data bases and expert systems
- Formal methods for design and verification
- Fault tolerance
- Performance evaluation of parallel systems
- Hardware and software tradeoffs
- Fully scalable systems
- Applications

Date and Place
The conference will be held in Ecole Normale Supérieure de Lyon, France, on 11-13 December, 1989

Deadlines
Prospective authors should submit five copies of their contribution in its final form (full paper), to the Chairman of the Scientific Programme Committee to arrive before **20 June, 1989** at the following address:

Professor Claude GIRAULT
Laboratoire MASI
Université Paris 6, Tour 65
4 Place JUSSIEU
F-75252 PARIS Cedex 05, FRANCE

telefax +(33)-1-43-29-41-96
telex UPMCSIX 200 145 F
telephone +(33)-1-43-36-25-25 ext 43-63
electronic mail mcvax ! inria ! litp ! cg

Paper submission
Papers should be no longer than 16 pages. The first page of the paper should indicate the author's name, affiliation, address, telephone, fax or telex numbers and electronic mail address. It should also contain a short abstract, some key words and the main topic (of the above list). A submission letter that contains a commitment to present the paper at the conference if accepted should accompany the paper. Authors will be notified of acceptance or rejection of their paper before the **end of September, 1989**.

Publication
Copies of the papers will be available at the conference. Camera ready copies are due at the conference. The proceedings will be published by North Holland after the conference, in the series of the WG 10.3 Conferences: Highly parallel computers, Nice, France, 1986 - Distributed processing, Amsterdam, The Netherlands, 1987 - Parallel processing, Pisa, Italy, 1988.

Provisional Scientific Committee

C.GIRAULT (Chairman) Paris, France

J.L.BAER, Seattle, USA
G.CARLSTEDT, Gotheburg, Sweden
V.CORDONNIER, Lille, France
E.DIRKX, Brussels, Belgium
E.FEUSTEL, Massachussets, USA
M.LEGENDI, Budapest, Hungary
K.MEHLHORN, Saarbrücken, RFG
M.RAYNAL, Rennes, France
B.SANDERS, Zürich, Switzerland
J.TIBERGHIEN, Bruxelles, Belgique
F. BAIARDI, Pisa, Italy
G.CHIOLA, Torino, Italy
M.COSNARD, Lyon, France
R.D.DOWSING, Norwich, UK
W.GILOI, Berlin, FRG
E.S.LEE, Toronto, Canada
H.PERROS, Raleigh, USA
G.L.REIJNS, Delft, Netherlands
N.SANTORO, Ottawa, Canada
M.VANNESCHI, Pisa, Italy
M.BARTON, Bristol, UK
Y.CHU, Maryland, USA
E.L.DAGLESS, Bristol, UK
K.BCIOGLU, New York, USA
C.HAMACHER, Toronto, Canada
R.W.MARCZYNSKI, Warsaw, Poland
R.PUIGJANER, Palma, Spain
B.SAMANI, New Jersey, USA
O.SPANNIOL, Aachen, FRG
C.WHITBY-STREVENS, Bristol, UK

CALL FOR PAPERS

Fifth Annual IEEE Symposium on

LOGIC IN COMPUTER SCIENCE

June 4–7, 1990, Philadelphia, PA

General Chair:
Prof. Albert R. Meyer
MIT Lab. for Computer Science,
NE43-315
545 Technology Square
Cambridge, MA 02139, USA
Internet:
meyer@theory.lcs.mit.edu

Conference Chair:
Prof. Jean Gallier
Dept. Computer and Info. Sciences
University of Pennsylvania
Philadelphia, PA 19104, USA
Internet: jean@cis.upenn.edu

Program Chair:
Prof. John C. Mitchell
Department of Computer Science
Stanford University
Stanford, CA 94305, USA
Internet: jcm@cs.stanford.edu

Program Committee:
K. Apt, *CWI Amsterdam and UT Austin*
J. Barwise, *Stanford*
E. Clarke, *CMU*
S. Cook, *Toronto*
S. Hayashi, *Ryukoku*
P. Kanellakis, *Brown*
J.-P. Jouannaud, *Paris-Sud*
D. Leivant, *CMU*
J. Mitchell (chair), *Stanford*
U. Montanari, *Pisa*
A. Pitts, *Cambridge*
E. Sandewall, *Linköping*
A. Scedrov, *Univ. of Pennsylvania*
M. Stickel, *SRI International*
G. Winskel, *Aarhus*

The LICS Symposium aims for wide coverage of theoretical and practical issues in computer science that relate to logic in a broad sense, including algebraic, categorical and topological approaches.

Suggested, but not exclusive, topics of interest include: *abstract data types, automated deduction, concurrency, constructive mathematics, data base theory, finite model theory, knowledge representation, lambda and combinatory calculi, logical aspects of computational complexity, logics in artificial intelligence, logic programming, modal and temporal logics, program logic and semantics, rewrite rules, software specification, type systems, verification.*

Paper submission: Fifteen (15) copies of a detailed abstract — not a full paper — should be *received* by **December 1, 1989** by the Program Chair (Attn: LICS). Authors from locations where access to reproduction facilities is severely limited may submit a single copy of their abstract. All authors will be notified of acceptance or rejection by **January 22, 1990.** Accepted papers typed on special forms for inclusion in the symposium proceedings will be due **March 1, 1990.**

Abstracts must be clearly written and provide sufficient detail to allow the program committee to assess the merits of the paper. References and comparisons with related work should be included. The entire extended abstract should not exceed ten (10) standard font double-spaced pages (2500 words). The title page of the submission should include a brief synopsis and author's name, address, phone number, and e-mail address if available. Papers must be unpublished and not submitted for publication elsewhere, including proceedings of other symposia or workshops. The December 1 deadline for receipt applies to overseas submissions as well. Late abstracts, or those departing significantly from these guidelines, run a high risk of rejection.

The symposium is sponsored by the IEEE Technical Committee on Mathematical Foundations of Computing in cooperation with the Association for Symbolic Logic and the European Association of Theoretical Computer Science, and with the anticipated cooperation of ACM SIGACT.

Organizing Committee: M. Abadi, J. Barwise, A. Chandra, E. Engeler, J. Gallier, J. Goguen, D. Gries, Y. Gurevich, D. Johnson, D. Kozen, Z. Manna, A. Meyer (chair), J. Mitchell, C. Papadimitriou, R. Parikh, G. Plotkin, D. Scott.

Publicity Chair: Prof. Daniel Leivant, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

Requests to join the LICS mailing list should be addressed to Internet: lics@cs.cmu.edu

Preliminary Announcement

International Workshop on Semantics for Concurrency

University of Leicester
23rd-25th July, 1990

The International FACS/BCS Workshop on Semantics for Concurrency will be held at the University of Leicester, Leicester, UK, in the week following ICALP 90. As you probably know, ICALP 90 will take place on July 16-20 at the University of Warwick, Coventry, UK. Leicester is conveniently positioned in central England, and is within easy reach of Coventry.

Semantics of concurrent systems is one of the most vigorous areas of theoretical computer science, but suffers from disagreement due to different, and often incompatible, attitudes towards abstracting non-sequential behaviour. When confronted with process algebras, which give rise to very elegant, highly abstract and compositional models, traditionally based on the interleaving abstraction, some argue that the wealth of contribution they have made is partially offset by the difficulty in dealing with topics such as fairness. On the other hand, the non-interleaving approaches, based on causality, although easing problems such as fairness and confusion, still lack structure, compositionality, and the elegance of their interleaving counterparts. Since both these approaches have undoubtedly provided important contributions towards understanding concurrent systems, the workshop will concentrate on what they have in common, rather than the way they differ.

The accommodation will be in student halls which lie in a residential area of Leicester close to the Botanical Gardens. The cost will be kept to a minimum. Those wishing to stay on after ICALP may obtain accommodation for the days preceding the workshop at very reasonable prices. During this time, there will be an opportunity to join in a programme of entertainment.

The workshop will consist of a tutorial day, devoted to invited talks concentrating on giving an overview of major approaches to concurrency, followed by two days of contributed talks, which will be sought in a Call for Abstracts to be sent out at a later date. The contributors will have an opportunity to submit a full paper to be refereed for inclusion in the proceedings.

Topics will include (the list is not exhaustive): mathematical models and notations for concurrency including categorical and topological methods, non-interleaving and partial-order semantics, distributed computation, process calculae, behavioural equivalences, behavioural properties of concurrent systems including fairness, logics for concurrency, real-time systems.

If you wish your name to be added to the mailing list, please contact:

Dr. Marta Kwiatkowska
Workshop on Semantics for Concurrency
Department of Computing Studies
University of Leicester
Leicester LE1 7RH, UK

E-mail JANET: mzk@uk.ac.le
Telephone: (0533)523603

Organising Committee: Marta Kwiatkowska, Mike Shields, Rick Thomas.

Recent Publications

This list is not claimed to be exhaustive. As we can refer only to papers we know about, please send appropriate hints to

Petri Net Newsletter
GMD-F1
Post Box 1240
D-5205 St. Augustin 1

AALST, W.V.D.; VOORHOEVE, M.; WALTMANS, A.: *The Taste Project.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 371-377 (1989)

The TASTE (The Advanced Studies of Transport in Europa) project aims at the description and simulation of logistic processes. The result of the project will be a toolkit of standard logistic components that can be combined graphically resulting in a prototype for decision support. The project is based upon a framework for modeling and specifying concurrent systems based on Petri nets.

ABELLARD, P.; BARBAGELATA, B.: *A Data Flow Numerical Processing Operator.* Microprocessing and Microprogramming, Vol. 25, pp. 133-138 (1989)

Efficiently carrying out a parallel caculation with a conventional multiprocessor machine poses difficulties: a limited concurrence, a complex control system, difficulties of programmation. In Data Flow architecture, these problems are non-existent; but a representation model of parallel calculation is necessary: Among the existing models, an extesnion of the basic Petri nets, dubbed "Data Flow Petri Nets", has been selected.

ACETO, L.; DE NICOLA, R.; FANTECHI, A.: *Testing Equivalences for Event Structures.* Lecture Notes in Computer Science, Vol. 280, Mathematical Models for the Semantics of Parallelism / Zilli, M.V. (ed.) — Springer-Verlag, pp. 1-20 (1987)

A flexible abstraction mechanism for models of concurrency is proposed. Using three classes of atomic observations (sequences of actions, sequences of multisets of actions and partial orderings of actions) different information on the causal and temporal structure of Event Structures is captured. As a reslult, three different semantic models for concurrent systems are obtained.

AHUJA, J.S.; VALAVANIS, K.P.: *Modified Petri Nets for Comprehensive Modeling of Flexible Manufacturing Systems.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 532-534 (1988)

AJMONE MARSAN, M.; CHIOLA, G.; FUMAGALLI, A.: *Timed Petri Net Models for the Accurate Performance Analysis of CSMA/CD Bus Lans.* Comput. Commun., Vol. 7, No. 6, pp. 304-312 (Dec., 1987)

This paper discusses timed Petri net models that have been developed to investigate the performance of a local area network with a linear topology. A detailed model is first constructed from which results can be obtained only through simulation. A compact, analytically tractable model is then developed, which is validated by comparing its results with those obtained by simulation of the detailed model. Performance curves show that the proposed compact model can be used to obtain accurate performance estimates.

AL-JAAR, R.Y.; DESROCHERS, A.A.: *A Survey of Petri Nets in Automated Manufacturing Systems.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 503-510 (1988)

ANISIMOV, N.A.: *Recursive Definition of Protocols Based on Petri Net Theory.* Avtomatika I Vychislitelnaya Tekhnika, No. 5, pp. 3-7 (1987), in Russian
also: Translated in: Autom. Control Comput. Sci., Vol. 21, No. 5, pp. 1-4 (1987)

The author introduces the algebra of regular macronets (RMN), which is an extension of the algebra of regular Petri nets and is intended for describing protocol structures. To describe protocols that employ similar procedures for executing dissimilar functions, the author proposes a formal model that is called a recursive RMN and that constitutes a compact notation of an RMN of a certain class.

ANISIMOV, N.A.: *Formal Models of Design and Description of Protocols on the Basis of Petri Nets.* Avtomatika I Vychislitelnaya Tekhnika, No. 6, pp. 3-10 (1988), in Russian

AUER, A.; KEMPPAINEN, P.; OKKONEN, A.; SEPPANEN, V.: *Automated Code Generation of Embedde Real-time Systems.* Microprocessing and Microprogramming, Vol. 24, No. 1-5, pp. 51–55 (1988)

A code generator is presented. It produces full tasks for real-time systems from a structured data/control flo description. The code generator is written in Prolog. It is now capable of producing programs in PL/M-86 languag for realistic applications and Petri nets for dynamic analysis.

BALAKRISHNAN, M.; MAJUMDAR, A.K.; BANERJI, D.K.; LINDERS, J.G.: *Synthesis of Decentralise Controllers from High Level Description.* Microprocessing and Microprogramming, Vol. 22, pp. 217–22 (1988)

The paper presents a technique for synthesizing decentralised controllers based on interconnections betwee control and data path. A modified Petri net called R-net is used as an intermediate representation. In an R-net firing of transitions and flow of tokens are controlled by two additional signals called select and enable conditions

BALDASSARI, M.; BERTI, V.; BRUNO, G.: *Object-oriented Conceptual Programming Based on PRO Nets.* Proc. of the Int. Conf. on Comput. Lang. 1988 — New York: IEEE, pp. 226–233 (1988)

The authors present PROTOB, an object-oriented computer-aided-software-engineering system based on high level Petri nets called PROT nets. It consists of several tools supporting specification, modeling, and prototypin activities.

BANDMAN, M.K.; BANDMAN, O.L.; ESIKOVA, T.N.: *Petri Nets – A Technique for Predicting th Process of the Formation of Territorial Systems.* Ekonomika I Matematicheskie Metody, Vol. 23, No. 6, pp. 1028–1040 (1987), in Russian

BARBEZ, J.M.; CRAYE, E.; GENTINA, J.C.; MAYET, J.: *Hierarchical Level and Implementation fo Analysis and Synthesis of Control and Reliability of Flexible Manufacturing Systems.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 552–558 (1988)

BARKAOUI, K.; LEMAIRE, B.: *An Effective Characterization of Minimal Deadlocks and Traps in Petri Nets Based on Graph Theory.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 1–21 (1989)

A new and simple characterization theorem of minimal deadlocks is given in terms of path properties based upon the original notions of "own transition" and "alternating circuit". Then, a polynomial algorithm is proposed that allows to decide whether a given place subset is a minimal deadlock. Last a straight forward extension of these results to traps is proposed.

BAUMGARTEN, B.; GIESSLER, A.; PLATTEN, R.: *The Derivation of Test Cases from Net Models of OSI Protocols.* Gesellschaft für Mathematik und Datenverarbeitung mbH, St. Augustin, Arbeitspapiere der GMD, Nr. 386 (1989)

In this paper the derivation of test cases from formal specifications of communication protocols and services is investigated in the framework of PROSIT methodology. The formal specification methods used are based on product nets, a special class of high level nets. Design and analysis tools are used in order to derive test cases from formal specifications, which greatly reduces sources of error.

BELIKOV, V.K.; RUTNER, Y.F.: *Matrix Specification and Analysis of Colored Petri Nets.* Soviet Journal of Computer and Systems Sciences, Vol. 26, No. 3, pp. 77–80 (1988)

A definition of a colored Petri net model on the basis of multi-dimensional matrices is introduced, and methods for analyzing it that are based on matrix equations are described. The proposed model represents more accurately the structure of the net, and this allows one to avoid the mutual destruction of information about the input and output arcs of a transition which is typical of other methods of specifying Petri nets.

BELIKOV, V.K.; RUTNER, J.F.: *Coloured Petri Nets and the Matrix Approach.* Program., (USSR), Vol. 12, No. 3, pp. 23–28 (May-June, 1988), in Russian
also: Translated in: Programming and Comput. Software, Vol. 14, No. 3 (March, 1989)

The authors define a model of a coloured Petri net using four-dimensional matrices. They describe methods of Petri net analysis based on matrix equations, which make it possible to overcome the difficulties of using coloured nets for firmware software.

BEST, E.; DESEL, J.: *Partial Order Behaviour and Structure of Petri Nets.* Gesellschaft für Mathematik und Datenverarbeitung mbH, St. Augustin, Arbeitspapiere der GMD, Nr. 373 (1989)

The paper argues that partial order semantics can be used profitably in the proofs of net theory. It is shown that most of Commoner's and Hack's structure theory of free choice nets can be phrased and proved in terms of partial order behaviour. The new poofs are considerably shorter.

BEST, E.: *Kausale Semantik nichtsequentieller Programme.* Berichte der GMD No. 174 — München: Oldenbourg Verlag (1989)

Es werden zwei parallele Programmiersprachen betrachtet, eine Pascal-ähnliche mit Zuweisungen, Variablen, atomaren Aktionen und Paralleloperationen und eine Occam-ähnliche mit Variablen und Kommunikation. Es wird eine semantiktreue Übersetzung von Programmen dieser Sprachen in elementare Petrinetze angegeben. Ein neuer, nicht-syntaktischer Abhängigkeitsbegriff wird untersucht, und es wird gezeigt, daß die Halbordnungssemantik von Petrinetzen eine Halbordnungssemantik von parallelen Programmen induziert, die solche Abhängigkeiten respektiert.

BHAT, U.N.; NARAYAN, K.; KRISHNA, M.: *Realiability Models for Computer Systems: An Overview Including Dataflow Graphs.* Sadhana, Vol. 11, No. 1-2, pp. 167–186 (1987)

This paper provides an overview of the approaches to reliability modeling. The models discussed include structure models, simple stochastic models and decomposable stochastic models. Structure models give reliability as a function of the topological structure of the system. Petri nets and dataflow graphs facilitate the analysis of complex systems by providing a convenient framework for reliability analysis.

BILLINGTON, J.: *A High-Level Petri Net Specification of the Cambridge Fast Ring M-Access Service.* University of Cambridge, Computer Laboratory, Technical Report No. 121 (Dec., 1987)

Numerical Petri Nets are used to characterise the Cambridge Fast Ring Hardware at a high level of abstraction. The NPN model describes the service provided to *users* of the hardware. The model has been developed to formalise the M-Access service definition in order to remove ambiguities and as a basis for the development and verification of protocols using the M-Access service.

BILLINGTON, J.; KINNY, D.: *Computer Aided Protocol Engineering.* Proc. of the Conf. on New Business Applications of Information Technology, Melbourne, pp. 69 ff (1989)

The main purpose of this paper is to present an overview of the field of protocol engingeering and to then briefly describe some of the tools being developed by Telecom to aid the protocol engineer. The authors introduce the concepts and scope of protocol engineering and stress the need for formal methods.

BOURBAKIS, N.G.; PAPAZOGLOU, M.; NGUYEN, S.H.: *Design and Simulation Using Petri Nets of Parallel WHT Multiprocessor System.* Microprocessing and Microprogramming, Vol. 25, pp. 341–346 (1989)

The paper discusses a parallel multiprocessor structure based on the C-MOVE architecture in order to design efficiently a system for the Walsh-Hadamard Transformation (WHT) of images. In additon, the functional simulation of tha parallel WHT multiprocessor system is presented for four levels of simulation, by using Petri nets.

BOUREY, J.P.; CASTELAIN, E.; GENTINA, J.C.; KAPUSTA, M.: *C.A.S.P.A.I.M.: A Computer Aided Disign of the Control System of F.M.S..* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 517–521 (1988)

BRODE, J.: *A Research Programming Support Environment for the Design and Study of Distributed Control Systems.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 543–545 (1988)

BÜTLER, B.; ESSER, R.; MATTMANN, R.: *A Distributed Simulator for High Order Petri Nets.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 22–34 (1989)

This paper describes a distributed simulator of high order Petri nets for parallel computer. It shows how the inherent parallelism of a Petri net can be used to obtain a fast simulator. The design decisions made in implementing a distributed simulator in hardware and software are discussed and a detailed description of both is given.

CAMPOS, J.; CHIOLA, G.; COLOM, J.M.; SILVA, M.: *Tight Polynomial Bounds for Steady-State Performance of Marked Graphs.* Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-07 (May, 1989)

The problem of computing both upper and lower bounds for the steady-state performance of timed and stochastic Marked Graphs is studied. In particular. Linear Programming Problems defined on the incidence matrix of the underlying Petri nets are used to compute tight (ie reachable) bounds for the throughput of transitions for live and bounded Marked Graphs with time associated with transitions.

CAMPOS, J.; CHIOLA, G.; SILVA, M.: *Properties and Steady-State Performance Bounds for Petri Nets with Unique Repetitive Firing Count Vector.* Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-08 (May, 1989)

The problem of computing both upper and lower bounds for the steady-state performance of timed and stochastic Petri Nets is studied. In particular, Linear Programming Problems defined on the incidence matrix of the underlying Petri net are used to compute bounds for the throughput of transitions for live and bounded nets with a unique possibility of steady-state behaviour.

CAMPOS, J.; SILVA, M.: *Steady-State Performance Evaluation of Totally Open Systems of Markovian Sequential Processes.* Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-10 (June, 1989)

Totally open systems of Markovian sequential processes are defined as a subclass of Markov stochastic Petri nets. They can be viewed a generalization of a subclass of monoclass queueing networks in which complex sequential servers can be synchronized according to some particular schemes. Structural analysis of these nets is considered for avoiding the state explosion problem of the embedded Markov chain.

CARDOSO, J.; VALETTE, R.; DUBOIS, D.: *Petri Nets with Uncertain Marking.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 35–51 (1989)

This paper shows the interest of introducing uncertainty and imprecision within Petri net based models. These two concepts are then introduced through a modification of the marking of a Petri net with objects, and of its interpretation. It is shown how, in some cases, uncertainty is propagated and how, sometimes, it is possible to go back to certainty.

CARPENTER, G.F.: *State Space Modelling in the Design of Robust Software for Distributed Systems: A Case Study.* Microprocessing and Microprogramming, Vol. 24, No. 1-5, pp. 793–799 (1988)

A distributed system comprising a set of discrete, loosely-coupled, processes will rely on the integrity of interprocess messages. This paper considers a particular design study in which state space modeling techniques (Petri nets and UCLA graphs) are used in the design of robust software which is free from dynamic faults.

CARPENTER, G.F.; HOLDING, D.J.; TYRRELL, A.M.: *Design and Simulation of Software Fault Tolerant Mechanisms for Application in Distributed Processing Systems.* Microprocessing and Microprogramming, Vol. 22, No. 3, pp. 175–185 (1988)

The paper considers techniques for the systematic and proper placement of software fault tolerant structures for distributed systems. It describes the design of such a system and shows how the error detection and recovery mechanisms can be included in the system model. The modelling and simulation are done using Petri nets.

CASTAGNOLO, B.; CORSI, F.; MARTINO, S.: *Penelope: A Graph Based Logic Simulator for MOS Circuits.* Proceedings of the IEEE International Symposium on Circuits and Systems — New York: IEEE, pp. 1365–1368 (1988)

A logic simulation tool called PENELOPE (Petri net logic performance evaluator) with precise delay estimation capabilities is presented. The simulator makes use of a description of the logic network in terms of a Petri-net-like graph which implements the truth table of each logic operator and also processes the property of describing the evolution of the signal transitions in the network.

CHANG, C.K.; AOYAMA, M.; JIANG, T.M.; SHI, W.; CHANG, Y.F.: *New Design Approach of Real-time Distributed Software Systems.* Proceedings of the IEEE Computer Society's International Computer Software & Applications Conference — New York: IEEE, pp. 474–479 (1987)

An approach to design specification based on the Petri net model is presented. In order to verify distributed software systems, an innovative system partitioning method is presented that is based on this specification technique. This method can partition a system represented by Petri nets into independent subsystems maintaining the communication behavior integrity.

CHANG, C.K.; JIANG, T.M.; AOYAMA, M.: *Specification Language for Real-time Distributed Systems.* Proceedings of the Int. Conf. on Comput. Lang.— New York: IEEE, pp. 258–265 (1988)

The authors introduce a design specification method, extended modified Petri nets (EMPN), and its description language (EMPNDL). The specification is based on a stochastic Petri-net model and is suitable for modeling real-time distributed systems. The syntax of the language is formally described in BNF grammar while the semantics is based on that for stochastic Petric nets.

CHAPPE, D.; BOURJAULT, A.: *Utilisation de Reseaux de Petri Temporises pour la Conception et l'Evaluation de Systemes d'Assemblage Automatises.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 538–542 (1988)

CHARRON-BOST, B.: *Combinatorics and Geometry of Consistent Cuts: Application to Concurrency Theory.* Laboratoire d'Informatique, URA 1327 du CNRS, Département de Mathématiques et d'Informatique, Ecole Normale Supérieure, LIENS–89-3 (Avril, 1989)

The authors define a concurrency measure of a distributed computation which is based an the number μ of its consistent cuts. The authors prove that counting consistent cuts take into account the non-transitivity of the concurrency relation, and they give a geometric interpretation of μ using the clock designed by Fidge and Mattern for characterizing concurrency between two events.

CHERKASOVA, L.A.; FILURIN, S.: *Concurrent Processes with Synchronization: Net and Algebraic Approach.* Lecture Notes in Computer Science, Vol. 363: Logic at Botik'89, Proc. of the Symposium on Logical Foundations of Computer Science — Springer-Verlag, pp. 63–80 (1989)

The paper consists of two parts: The first one deals with the net formalism (OS-nets) to specify concurrent processes with synchronization; the second part of the paper is devoted to the algebra of finite concurrent processes and its axiomatization.

CHISHOLM, G.H.; KLJAICH, J.; SMITH, B.T.; WOJCIK, A.S.: *Formal Modeling and Analysis of Fault-tolerance Properties for Software.* Argonne National Lab., IL., Report No. CONF-870399-1 (1987)

In an earlier paper the author has described the hierarchical modeling technique based upon Petri nets, and the formal analysis techniques based upon the automated reasoning software ITP/LMA. In this paper, the author demonstrates that the same modeling and analysis techniques apply to proving the fault-tolerance of the software. The approach that has been developed has provided insight into formal software specification as well as into the generation of test vectors for software.

CHIU, P.K.: *Interval Logic and Modified Labelled Net Model for System Specification and Verification.* Thesis, University of Hong Kong, Dept. of Electrical Engineering (1985)

COLOM, J.M.; SILVA, M.: *Convex Geometry and Semiflows in P/T Nets. A Comparative Study of Algorithms for Computation of Minimal P-Semiflows.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 74–95 (1989)
also: Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-01 (Jan., 1989)

P-semi-flows are non-negative left annullers of a net's flow matrix. The algorithms known in the domain of P/T nets for computing elementary semi-flows are basically improvements of the basic Fourier-Motzkin method. This paper presents two improved algorithms which are more efficient and robust when handling "real-life" nets.

COLOM, J.M.; SILVA, M.: *Improving the Linearly Based Characterization of P/T Nets.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 52–73 (1989)
also: Universidad de Zaragoza. departamento de ingenieria electrica e informatica, Research Report 89-03 (Jan., 1989)

The state equation is a linear description of the reachable markings and firing count vectores of a P/T net. It has the disadvantage that ists solution space, in general, includes additional integer unreachable or unfirable vectors. The existence of methodes which eliminate spurious solutions of the direct state equation would bring structural verification methods closer to behavioral methods. Two elimination methods are presented here. Both are based on adding to the state equation linear restictions which check the transition firing rule.

COUSIN, B.; ESTRAILLIER, P.: *PLASMA: A Multi-medium Protocol for the Link Layer.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 487–490 (1988)

COUVREUR, J.M.; MARTINEZ, J.: *Linear Invariants in Commutative High Level Nets.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 96–114 (1989)
also: Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-04 (Jan., 1989)

Commutative nets are a subclass of colored nets whose color functions belong to a ring of commutative diagonalizable endomorphisms. Commutative nets include net subclasses such as regular homogeneous nets and ordered nets. Mathematical properties of the color functions of commutative nets allow a a symbolic computation of a family of generators of flows.

DEGANO, P.; MESEGUER, J.; MONTANARI, U.: *Axiomatizing Net Computations and Processes.* Proc. of the 4th Annual Symposium on Logic in Computer Sciene, Asilomar, Ca., USA (1989)

The descriptions of concurrent behaviors in terms of partial orderings lack a suitable, general notion of *sequential composition.* In this paper, a new algebraic axiomatization is proposed, where, given a net N, a term algebra

$\mathcal{P}[N]$ with two operations of parallel and sequential composition is defined. The congruence classes generated by a few simple axioms are proved isomorphic to a slight refinement of classical processes.

DESEL, J.; MERCERON, A.: *Vicinity Respecting Net Morphisms.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 115–133 (1989)

Vincinity respecting morphisms are a restricted kind of net morphisms which respects pre- and post-sets of elements but allows to map S-elements to T-elements and vice versa. Besides some general properties of (vincinity respecting) morphisms it is shown that special vincinity respecting morphisms preserve S-component-coverings. A dual result holds for T-component-coverings.

)I LEVA, A.; GIOLITO, P.: *Information System Dynamics Representation in Production Environments.*)ata & Knowledge Engineering, Vol. 3, No. 3, pp. 149–161 (1988)

This paper discusses the representation of dynamic features of an organization. The approach adopted is based on the use of high level Petri nets and allows the description of the system at two different levels: (a) the organization level, and (b) the conceptual level.

)ILLON, L.K.; AVRUNIN, G.S.; WILEDON, J.C.: *Constrained Expressions: Toward Broad Applicability ›f Analysis Methods for Distributed Software Systems.* ACM Transactions on Programming Languages ›nd Systems, Vol. 10, No. 3, pp. 374–402 (1988)

In this paper the authors present a basis for broadly applicable analysis methods for distributed software systems. It is shown how constrained expression representations are obtained from descriptions of systems in three different notations: SDYMOL, CSP, and Petri nets. Features of these three notations span most of the significant alternatives for describing distributed software systems.

)UBOIS, P.: *A Flexible Workshop Design and Optimisation Using SEDRIC: A Petri Net Simulator.* 'roc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 378–393 (1989)

The goal of this paper is to insure the industrial credibility of a simulation study using Petri nets. This study represented a two month load for one person. Three weeks were necessary to analyse the problem, realise the model and run first simulations. Two weeks were in fact necessary to design and debug the model. The Petri net theory is the reason of this fast answer time.

'UGAN, J.B.; CIARDO, G.: *Stochastic Petri Net Analysis of a Replicated File System.* IEEE, 'ransactions on Software Engineering, Vol. 15, No. 4, pp. 394–401 (1989)

The authors present a stochastic Petri net model of a replicated file system in a distributed environment where replicated files reside on different hosts and a voting algorithm is used to mantain consistency. Witnesses, which simply record the status of the file but contain no data, may be used to reduce overhead. The model is sufficiently detailed to include file status as well as failure and repair of hosts where copies or witnesses reside.

UTHEILLET, C.; HADDAD, S.: *Regular Stochastic Petri Nets.* Supplement to the Proc. of the 10th Int. onf. on Application and Theory of Petri Nets, pp. 43–62 (1989)

An extension of regular nets, a class of colored nets, to a stochastic model is proposed. It is shown that the ;ymmetries in this class of nets make ist possible to develop a performance evaluation by constructing only a graph of symbolic markings, the vertices of which are classes of states, instead of the whole reachability graph. t is proven that all the states in a class have the same probability.

SPARZA, J.; SILVA, M.: *Circuits, Handles, Bridges and Nets.* Proc. of the 10th Int. Conf. on Application ıd Theory of Petri Nets, pp. 134–153 (1989)
so: Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-02 an., 1989)

This paper introduces two new structural objects for the study of nets: handles and bridges. They are shown o provide sufficient conditions of good behaviour for general ordinary nets as well as a new characterisation f structural liveness for the special subcalss of Free Choice nets. This characterisation is used to approach a nodular synthesis theory of Free Choice nets.

PARZA, J.; SILVA, M.: *A Polynomial-Time Algorithm to Decide Liveness of Bounded Free Choice Nets.* ıiversidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-06 (May, 89)

'he authors propose a polynomial time algorithm to decide liveness for Bounded Free Choice nets, thus proving n enlarged version of a conjecture raised by N. Jones et al. The combination of this algorithm with the well nown one that decides conservativity of a net yields a polynomial time algorithm to decide if any given Free 'hoice net is live&bounded.

ESPARZA, J.; BEST, E.; SILVA, M.: *Minimal Deadlocks in Free Choice Nets.* Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-09 (May, 1989)

The purpose of this note is to present a a simple characterisation of minimal deadlocks for Free Choice nets. This simplified characterisation is used to obtain two results: the first is an algorithm that constructs minimal deadlocks in (strongly connected) Free Choice nets; the second is the existence of a close relationship between minimal and strongly connected deadlocks in the same subclass.

FARAH, B.N.: *Decision Support System for the Analysis and Design of Information Systems for Flexible Manufacturing.* Advances in Modelling & Simulation, Vol. 12, No. 1, pp. 33–56 (1988)

To facilitate the analysis of information requirements and the design of information systems for flexible manufacturing, a decision support system is described. This system uses the information cell model. Petri net representations and their interactions are then constructed.

FLEISCHHACK, H.; WEBER, A.: *Rule Based Programming, Predicate Transition Nets and the Modeling of Office Procedures and Flexible Manufacturing Systems.* Supplement to the Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 23–42 (1989)

Production systems as known from rule based programming and Predicate/Transition nets are considered as models for describing office procedures and flexible manufacturing systems. It is shown that both concepts are equivalent in a very strong sense: each can be simulated stepwise by the other. Hence, production system nets are introduced as a formalism combining the advantages of the above models.

FLORIN, G.; KAISER, C.; NATKIN, ST.: *Petri Net Models of a Distributed Election Protocol on Unidirectional Ring.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 154–173 (1989)

This paper is devoted to the performance analysis of an election protocol for an undirectional ring. Two models are given of increasing complexity in order to compute several performance criteria. The two models are defined using coloured Petri nets.

FRACHET, J.P.; POUGET, J.P.; SAIDI, A.: *Object Oriented Model for the Specification of an Integrated Set of Tools for Aiding Design and Exploitation of Automated Production System.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 546–551 (1988)

FRIEDRICH, G.R.: *DIOGENES — An Interactive Tool for Modelling and Simulation by Modified Timed Petri Nets.* iir: informatik informationen. reporte, Vol. 17 : Modelling and Performance Evaluation of Distributed Computer Systems: Proceedings of the 2nd International Seminar, Nov. 1988, Wendisch Rietz, Part 3 — Akademie der Wissenschaften der DDR, Institut für Informatik (1988)

Modified, timed Petri nets (M-Nets) are used to model and to analyse the dynamical behaviour of complex concurrent systems. A brief introduction to the M-Net developoment system DIOGENES is given. As an example, special features for modelling interruptible processes are demonstrated.

FUKAMI, S.: *Performance Evaluation Method for Interprocessor Interrupts by Message Passing.* Microprocessing and Microprogramming, Vol. 23, No. 1-5, pp. 185–190 (1988)

A performance evaluation method for an inter-processor interrupt mechanism in a shared bus multi-microprocessor system is studied. In this mechanism, each processor has a message box to which other processors can write a message via a shared bus. Each message written into a processor's message box causes an interrupt. A generalized stochastic Petri net model is proposed for analyzing the mechanism's behavior.

GENTINA, J.C.; CORBEEL, D.: *Petri Nets in Manufactoring.* System & Control Encyclopedia: Theory, Technology, Applications; Vol. 6 / Singh. M. G. (ed) — Oxford: Pergamon Press, pp. 3670–3673 (1987)

The main purpose of this article is the description of a flexible manufactoring system by means of a top-down structured approach from the point of view of both control and errror processing. A three-level approach for the analysis and synthesis of such a system of production is proposed. Adaptive colored and structured Petri nets are used to model the control system and its various communications.

GIGLMAYR, J.: *On the Computation of Flow and Cycle Time Distributions of Marking States in Stochastic Petri Nets.* NTZ Archiv, Vol. 10, No. 12, pp. 339–349 (1988)

GOLOVIN, J.; ATVEJEVA, V.: *Use of Petri Net Expansions for Protocol Specification.* Avtomatika I Vychislitelnaya Tekhnika, No. 6, pp. 11–17 (1988), in Russian

GOMM, D.; WALTER, R.: *Das Problem der Verteilten Terminierung: Eine Lösung mit Petri Netzen.* Universität Bonn, Institut für Informatik, Diplomarbeit (1989)

Das Problem der verteilten Terminierung besteht in einem System kooperierender sequentialler Prozesse darin, durch einen Kontroll-Algorithmus die *einzelnen* Prozesse zu befähigen, den globalen Systemzustand, der ein Ende einer verteilten Bearbeitung anzeigt, feststellen zu können. Mit Hilfe von Pr/T-Netzen wird ein Lösungsalgorithmus hergeleitet.

GREVET, J.L.; JANDURA, L.; BRODE, J.; LEVIS, A.H.: *Execution Strategies for Petri Net Simulation.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 511–516 (1988)

GUNTER, C.; GEHLOT, V.: *Nets as Tensor Theories.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 174–191 (1989)

This report is intended to describe and motivate a relationship between a class of nets and the fragment of linear logic built from the tensor connective. In this fragment of linear logic a net may be represented as a *theory* and a computation on a net as a *proof.* A rigorous translation is described and a soundness and completeness theorem is stated.

HAAS, P.J.; SHEDLER, G.S.: *Stochastic Petri Net Representation of Discrete Event Simulations.* IEEE, Transaction on Software Engineering, Vol. 15, No. 4, pp. 381–393 (1989)

A generalized semi-Markov process (GSMP) is the usual model for the underlying stochastic process of a discrete event simulation. In this paper, the authors study the modeling power of stochastic Petri nets (SPN's) with timed and immediate transitions and show that such Petri nets provide a general framework for simulation. The principle result is that for any (countable) state GSMP there exists an SPN having a marking process that "mimics" the GSMP.

HAVERKORT, B.R.; NIEMEGEERS, I.G.: *Dynamic Queueing Networks.* Technische Univ. Twente, Enschede (Netherlands), Report No. MEMO-INF-88-31 (1988)

A class of combined queueing network (QN) and generalized stochastic Petri net (GSPN) models to analyze the performance of systems with a dynamic structure is proposed. A GSPN is used to describe the dynamic behavior of the system under study. The possible markings are used as actual parameters of a parameterized queueing network. Depending on the time-scales in the two model parts, either a direct overall solution method is proposed or a solution method based on behavioral decomposition.

HEINER, M.: *Petri Net Based Verification of Communication Protocols Specified by Language Means.* Informatik, Informationen, Reporte — Berlin: Akademie der Wissenschaften der DDR (1989)

The approach described in this paper should be comprehended as part of a general workstation of protocal engingeering. The poblems discussed are influenced by some experience collected by Petri net based general verification of any high level language for parallel or distributed processing. This general experience if transferred to the special field of language means for communication protocols.

HILLION, H.: *Modelisation et Analyse des Systems de Production Discrets par les Reseaux de Petri Temporises.* Paris, Univ., These (1989)

Nous présentons. dans ce travail, une approche basée sur les réseaux de Petri temporisés (déterministes) pour la modélisation et l'analyse des systèmes de production discrets. L'étude porte d'une part sur les job-shops (ateliers flexibles), d'autre part sur les systèmes d'assemblage multi-niveaux.

HILLION, H.P.: *Performance Evaluation of Decision Making Organizations Using Timed Petri Nets.* Massachusetts Inst. of Tech., Cambridge. Lab. for Information and Decision Systems, Master's Thesis, Report No. LIDS-TH-1590 (1986)

Decision making organizations are modeled as asynchronous concurrent systems, using Timed Petri Nets. The modeling allows for evaluating the time-related performances, with respect to the following measures: The maximum throughput rate, defined as the maximum processing rate achievable by the system, and the execution schedule, which determines the earliest instants at which the different operations can occur in the process.

HILLION, H.P.; LEVIS, A.H.: *Timed Event-graph and Performance Evaluation of Systems.* Massachusetts Inst. of Tech., Cambridge. Lab. for Information and Decision Systems, Report No. LIDS-P-1639 (1987)

The performance analysis of Timed Event-Graphs, including both deterministic and random models, is considered. First, a bound to the average firing rate in steady-state is given. The second result deals with an extended deterministic model, in which the transition processing times are a function of the number of firing repetitions.

HILLION, H.P.; PROTH, J.M.: *Performance Evaluation of Job-shop Systems Using Timed Event-graphs.* IEEE, Transactions on Automatic Control, Vol. 34, No. 1, pp. 3–9 (1989)

Timed event-graphs are used for modeling and analyzing job-shop systems. The modeling allows for evaluating the steady-state performance of the system under a deterministic and cyclic production process. Given any fixed procesing times, the productivity of the system can be determined from the initial state.

HIRSHFELD, J.; RABINOVICH, A.; TRAKHTENBROT, B.A.: *Discerning Causality in Interleaving Behavior.* Tel-Aviv University, Technical Report No. 133/89 (1989)
also: Lecture Notes in Computer Science, Vol. 363: Logic at Botik '89, Proc. of the Symposium on Logical Foundations of Computer Science — Springer-Verlag, pp. 146–162 (1989)

The paper examines situations where for a given system there is a strong intuition and a general consensus about its interleaving behavior but the inherently causal aspects of behavior are still to be discerned. As an application to the theory one aims at a better understanding of the semantics of Place-Transition systems.

HOWELL, R.R.; ROSIER, L.E.; YEN, H.C.: *Normal and Sinkless Petri Nets.* Lecture Notes in Computer Science, Vol. 380: Fundamentals of Computation Theory, Proc. of the Int. Conf. FCT'89 — Springer-Verlag, pp. 234–243 (1989)

The authors examine both the modeling power of normal and sinkless Petri nets and the computational complexities of various classical decision problems with respect to these two classes. They argue that although neither normal nor sinkless Petri nets are strictly more powerful than persistent Petri nets, the nonetheless are both capable of modeling a more interesting class of problems. The authors give strong evidence that normal and sinkless Petri nets are easier to analyze than persistent Petri nets.

HUBER, P.; JENSEN, K.; SHAPIRO R.M.: *Hierarchies in Coloured Petri Nets.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 192–209 (1989)

The paper shows how to extend Coloured Petri Nets with a hierarchy concept. The paper proposes five different hierarchy constructs, which allow to structure large CP-nets as a set of interrelated subnets. The paper discusses the properties of the proposed hierarchy constructs.

HUNG, D.V.; KNUTH, E.: *A Noninterleaving Semantic for Communicating Sequential Processes: A Fixed-point Approach.* Acta Cybern., Vol. 8, No. 3, pp. 293–311 (1987)

The paper introduces a noninterleaving type of semantics for communicating processes. In contrast to other approaches, concurrency is expressed here explicitly and is distinguished from nondeterminism. Semantics of programs are obtainable by equations in this model. A comparison is also given with respect to labelled event structures and Petri nets.

HURA, G.S.; COSTARELLA, M.A.; BUELL, C.G.; CVETANOVIC, M.M.: *PNSOFT: A Menu-driven Software Package for Petri Net Modeling and Analysis.* Proc. Int. Conf. on Comput. Lang. 1988 — New York: IEEE, pp. 41–47 (1988)

A discussion is presented of PNSOFT, a user-friendly, menu-driven software package for Petri-net modeling and analysis of real-time systems. It aids in the design, management and analysis of Petri-net models. PNSOFT runs in an automated programming environment that provides both a graphical and a matrix interface for the creation and modification of Petri-net models in different environments.

ITTER, F.: *Einsatz von Petri Netzen zur Beschreibung von Fertigungssystemen.* Zeitschrift f. Wirtschaftliche Fertigung u. Automatisierung, Vol. 84, pp. 206 ff (1989)

Im Beitrag wird der Einsatz von Petrinetzen am Beispiel der Konzeption einer Fertigungssteuerung erläutert. Es werden sowohl die Vorgehensweise bei der Modellbildung für die Anlage selbst und für die Steuerungssoftware als auch die Projekterfahrung geschildert.

JAQUEMIN, T.; DESCOTES-GENON, B.; LADET, P.: *Expert Systems for Grafcet Creation.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 525–527 (1988)

JOHNEN, C.: *Algorithmic Verification of Home Spaces in P/T-Systems.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 491–493 (1988)

KETTUNEN, E.; MONTONEN, E.; TUULINIEMI, T.: *A Comparison of Pr/T-Net Based FIFO Channel Models.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 479–482 (1988)

KINUYAMA, M.; MURATA, T.: *Generating Spihons and Traps by Petri-Net Representation of Logic Equations.* Proc. 2nd IECE (Japan) Conference on Net Theory, pp. 93–100 (Dec., 1986)

KIRILLOV, V.Y.: *Automation Interpretation of Petri Nets.* Soviet Journal of Computer and Systems Sciences, Vol. 26, No. 3, pp. 66–77 (1988)

The paper introduces an interpretation of Petri nets as collections of finite automata connected by certain constraints. A sufficient equivalence condition, which makes it possible to apply to Petri nets the apparatus

of finite-automation equivalent transformations, is proven on its basis. Conditions for Petri net liveliness are proven and an algorithm for testing for these conditions is defined.

KLJAICH, J.JR.; SMITH, B.T.; WOJCIK, A.S.: *Formal Verification of Fault-tolerance Using Theorem-proving Techniques.* Argonne National Lab. (1989)

This paper describes a formal verification system based on the use of automated reasoning techniques to validate fault-tolerance. A Petri net representation will be described together with the theorem-proving implementation of rule-based system for manipulating system descriptions.

KODOMA, S.; MURATA, T.: *On Necessary and Sufficient Reachability Conditions for some Subclasses of Petri Nets.* Chicago: Univ. of Illinois, Tech. Rep. #UIC-EECS 88-8 (June, 1988)

KORCZYNSKI, W.: *Algebraic Characterization of Concurrent Systems.* Fundamenta Informaticae, Vol. 11, No. 2, pp. 171–194 (1988)

In the paper a characterization of concurrent systems as algebras of a special kind is given. The algebras are defined by semi-equations in the first order language. — In the Petri net approach processes are some system morphisms; in the approach of Winkowski the notion of system is not needed. The approach presented in this papaper is a sense "between" the above ideas; it comprises a new concept of system morphism.

KOSYACHENKO, S.; KULBA, A.V.; MAMIKONOV, A.; SOKOLOVA, E.: *Using Petri Nets During Debugging to Localize Software Errors.* Automation and Remote Control USSR, Vol. 49, No. 5, pp. 680–689 (1988)

KYRATZOGLOU, I.M.; LEVIS, A.H.: *Computer Aided Petri Net Design for Decision-making Organizations.* Massachusetts Inst. of Tech., Report No.: LIDS-P-1806 (1988)

A Computer Aided Petri Net Design System is developed for creating decision-making organizational architectures of arbitrary complexity and for computing the structural attributes describing the organization. The Design System has four modes of operation: a Graphics Editor mode, a text Editor Mode, a Structural Analysis mode, and a Hardcopy mode.

LEU, D.; SILVA, M.; COLOM, J.M.; MURATA, T.: *Interrelationships among Various Concepts of Fairness for Petri Nets.* Proc. 31th Midwest Symposium Circuits and Systems (1988)

LI, H.F.; CHEUNG, Y.S.; CHIU, P.P.K.: *Microcomputer System Specification Using Interval Logic and a Modified Labelled Net Model.* IEEE Proceedings, Vol. 133, Pt. E. No. 4. pp. 223–234 (July, 1986)

The application of interval logic and modified labelled-net models for system specification and verification is introduced. By means of interval logic and modified labelled-net models, system properties involving time interval information can be specified and verified. The verification of system properties by derivation rules can be performed in the net space by net transformations which can be handled by an interactive computer.

LIN, C.; MARINESCU, D.C.: *Stochastic High-Level Petri Nets and Applications.* IEEE Transactions on Computers, Vol. 37, No. 7, pp. 815–825 (July, 1988)

A new class of stochstic Petri nets is proposed in this paper. The stochstic high-level Petri nets (SHLPN's) are high level Petri nets augumented with exponentially distributed firing times. The main advantage of modeling homogeneous systems using SHLPN's is that the resulting models are simpler, mor intuitive, and have a smaller number of states as the examples show.

LINDQVIST, M.: *Parameterized Reachability Trees for Predicate/Transition Nets.* Acta Polytechnica Scandinavica, Mathematics and Computer Science, No. 54 (1989)

An approach to elaborate the reachability analysis method in order to make it more applicable also to larger systems is discussed in detail. The difficulties encountered in this approach are taken as a motivation for developing an approach based on introducing parameters into the marking of a net. The formalism to deal with parameterized markings is devleloped; parameterized reachability trees are defined and an algorithm for generating them is presented. They are shown to be significantly smaller than ordinary reachability trees and to contain the same information.

MAKAROV, I.M.; NAZARETOV, V.M.; KULBA, A.V.; SHVETSOV, A.R.: *Petri Nets with Multicolor Markers.* Soviet Journal of Computer and Systems Sciences, Vol. 26, No. 4, pp. 34–40 (1988)

Problems of investigation of Petri nets with multicolor markers are considered. The notions of selective and regenerative Petri nets are introduced. Methods of analysis of color reachability are proposed. Properties of boundedness, safety, preservability and liveness of Petri nets with multicolor markers are investigated.

MARTINEZ, J.; MURO, P.R.; SILVA, M.; SMITH, F.S.: VILLARROEL, J.L.: *Merging Artificial Intelligence Techniques and Petri Nets Real Time for Scheduling and Control of Production System.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 528–531 (1988)

MERCIER DES ROCHETTE, R.; DESCOTES-GENON, B.; LADET, P.: *Modelling and F.M.S. Control Implementation.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 559–561 (1988)

MURATA, T.: *Petri Nets.* System & Control Encyclopedia: Theory, Technology, Applications; Vol. 6 / Singh, M. G. (ed) — Oxford: Pergamon Press, pp. 3665–3670 (1987)

The article gives a short introduction into the basic concepts of Petri net theory including timed and colored Petri nets.

MURATA, T.; KOMODA, N.: *Development of a Petri Net Based FA Controller and its Applications.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 394–403 (1989)

A Petri net based factory automation controller (SCR) for flexible and maintainable control specifications is developed. In SCR, system control specifications are represented by an enhanced Petri net based, problem oriented language. This paper describes some features of the SCR.

MURATA, T.: *Petri Nets: Properties, Analysis and Applications.* Proceedings of the IEEE, Vol. 77, No. 4, pp. 541–580 (April, 1989)

This is an tutorial-review paper on Petri nets. It starts with a brief review of the history and the application areas considered in the literature. It then proceeds with examples, behavioral and structural properties, methods of analysis, subclasses of Petri nets and their analysis. The paper contains a section devoted to marked graphs and presents introductory discussions on stochastic nets and on high-level nets. Also included are recent results on reachability criteria.

MURATA, T.; SHENKER, B.; SHATZ, S.M.: *Detection of Ada Static Deadlocks Using Petri Net Invariants.* IEEE Transactions on Software Engineering, Vol. 15, No. 3, pp. 314–326 (March, 1989)

This paper presents a method for detecting deadlocks in Ada tasking programs using structural and dynamic analysis of Petri nets. Algorithmic translation of the Ada programs into Petri nets that preserve control flow and message flow properties is described. Properties of these Petri nets are discussed, and algorithms are given to analyze the nets to obtain information about static deadlocks that can occur in the original programs.

MURATA, T.; SILVA, M.: *Fair Control of Discrete Event Systems Via Petri Nets.* Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-11 (June, 1989)

This paper is concerned with the use of Petri net structures to control discrete event systems in a "fair" manner when some resource sharing is involved. There are many differnt ways of defining "fairness". In this paper, first the authors review the event synchronization distance and bounded-fairness concepts. The authors then discusss some relations between these and other concepts of fairness, such as unconditional, strong and weak fairness.

NUTT, G.J.: *A Flexible, Distributed Simulation System.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 210–226 (1989)

The paper describes a distributed modeling system based on a variant of timed Petyri nets. The system provides interactive editing and interpretation facilities. A user can define a model, then mark it with tokens and observe the operation of the net through real time animation. The system also supports simultaneous use among multiple users.

OREKHOV, A.A.: *A Polynomial Algorithm for Finding the T-invariants of a Petri Net.* Otdelenie Matematiki, Mekhaniki i Kibernetiki Akademii Nauk, Ukrainskoi SSR, Kibernetika, No. 3, pp. 102–103 (1988)

PAGNONI, A.: *Computer Management of Coordination in Flexible Manufacturing Systems.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 500–502 (1988)

PEREZ, T.A.: *A Quantitative Analysis of Bounded Ordered Producer-consumer Systems Operating in Real-time Modelled with Petri Nets.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 227–246 (1989)

The main aim of this paper is to study some aspects of quantitative nature in the behaviour of Producer-Consumer systems operating in real time, within the Petri net formalism. This is done for four different bounded ordered Producer-Consumer models by means of timed nets (with inhibitor arcs) and untimed nets; it is shown for each model that the various given representations are equivalent from the quantitative point of view.

PETERKA, G.; MURATA, T.: *Proof Procedure and Answer Extraction in Petri Net Model of Logic Programs.* IEEE Trans. Software Eng., Vol. 15, No. 2. pp. 209–217 (1989)

This paper discusses a proof procedure and answer extraction in a high-level Petri net model of logic programs. These programs are restricted to Horn clause subset of first order predicate logic and to finite problems.

PLESSMANN, K.W.; WYES, J.: *Veranschaulichung der Echtzeitkonzepte in PEARL anhand von Petri-Netzen.* Angewandte Informatik, No. 7, pp. 296–304 (1987)

This paper presents elements of a computer language and the resulting semantic conditions via nets. Especially in the field of real-time data processing it is of vast importance to take into account the condition of the general system environment. PEARL is a computer language which has been developed for real-time programming, whereas Petri nets are especially suited to discern system conditions in an overview.

POMELLO, L.; SIMONE, C.: *A State Transformation Preorder over a Class of EN-systems.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 247–271 (1989)

The paper introduces a notion of preorder between contact-free Elementary Net Systems, which is based on observability of places. This latter allows one to define the notion of Observable Local States, whose transformations are the main concern of the proposed preorder. This preorder relates systems with a different level of granularity in locals states transformation.

PRATT, T.W.: *Methods for Design and Evaluation of Integrated Hardware/Software Systems for Concurrent Computation.* Virginia Univ., Charlottesville, Report No. NAS 1.26:181522 (1987)

Two testbed programming environments to support the evaluation of a large range of parallel architectures have been implemented under the program Parallel Implementation of Scientific Computing Environments (PISCES). A new formal model of concurrent computation has been developed, based on the mathematical system known as H graph semantics together with a timed Petri net model of the parallel aspects of a system.

PULLI, P.J.: *Execution of Structured Analysis Specifications with an Object Oriented Petri Net Approach.* Proc. Conf. on Comput. Lang. 1988 — New York: IEEE, pp. 286–293 (1988)

The dynamic behavior of the basic modeling elements of the real-time structured-analysis specification method has been defined with an object-oriented approach as a hierarchy of classes of Smalltalk-inscribed Petri nets. As a result of this definition, a given instance of a structured-analysis basic modeling element can be derived as an instance of the corresponding net class.

REMY, P.A.; LEVIS, A.H.: *Algorithmic Generations of Feasible Systems Architectures.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 483–486 (1988)

RILLO, M.: *Using Petri Nets and Rule-based System in Manufacturing Systems.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 535–537 (1988)

SCHMIDT, H.W.: *Specification and Correct Implementation of Non-sequential Systems Combining Abstract Data Types and Petri Nets.* Gesellschaft für Mathematik und Datenverarbeitung mbH, St.Augustin, GMD-Bericht, Nr. 176 (1989)

This thesis defines and analyzes notions of specification. modularity and stepwise implementation for non-sequential systems. For this purpose the notion of Predicate-Event (PrE) nets is introduced and a theory of these nets is developed. Algebraic and net-theoretic properties of PrE nets are studied in the repsective framework.

SCHNIEDER, E.; GÜCKEL, H.: *Petri-Netze ind der Automatisierungstechnik.* Automatisierungstechnik (at), Vol. 37, No. 5, pp. 173 ff (1989)

Petri nets help to establish system requirements, to specify system models, and to define them more and more concretely. The theory of Petri nets permits in addition the static and dynamic testing of system models even before implementation. In the near future the automatic transformation of nets into programs and circuits is to be expected as the end of the product development.

SCHÜBEL, T.: *A Method for the Verification of Predicate/Transition Nets and its Comparison with Hoare's Calculus of Program Correctness.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 272–291 (1989)

The notion of a specification for predicate/transition nets is defined. An easily provable local criterion is stated, which is equivalent to global correctness. This equivalence yields a proof method for concurrent systems.

SONNENSCHEIN, M.: *On Petri Nets and Data Flow Graphs.* Technische Hochschule Aachen, Fachgruppe Informatik, Aachener Informatik-Berichte, No. 88-18 (1988)

This paper offers an attempt to unify the notations of Petri nets and data flow graphs by introducing a new concept called 'generalized computation nets' (GCN's). All common types of Petri nets and static data flow

graphs can be defined as special GCN's. The definitions of the various types of Petri nets are condensed to the definition of 'additive' GCN's. Using this definition it is also possible to unify the concepts of P-invariants for Petri nets.

SOUISSI, Y.; MEMMI, G.: *Compositions of Nets via a Communication Medium.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 292–311 (1989)

The general idea of this paper is to build a system in a modular way and to deduce its properties only by analysing its smaller components. Since, in general, composing subnets does not preserve properties at the level of the global net, the problem is to find constraints on the subnets for establishing such results. The authors discovered that in some cases it is sufficient to put structural constraints only on the subnet generated by the elements shared by the two nets to be composed.

STOICA, N.; ROTH, G.: *Neuronal Networks Modelled by Petri Type Nets with Controllers.* Proc. of the 12th IFIP Conf., Budapest: Lect. Notes Control Inf. Sci., pp. 923–932 (1986)

The network model is conceived to discretly simulate the analogue functioning of the neurons. The neuronal network is modelled by a Petri type net where the places represent the neurons, the transitions represent the synapses, and the marking is interpreted as potential of the neuron's membrane. The impulses are transmitted according to the flow relation of the net. For the purpose of simulations, a set of FORTRAN programs called NETCON was created.

STOTTS, P.D.; CAI, Z.N.: *Modelling Temporal Behaviour of Robot Lattices with Binary Timed Petri Nets.* Proc. of the 12th IMACS World Conf., Vol. 3, pp. 494–499 (1988)

STOTTS, P.D.; FURUTA, R.: *TRELLIS: A System for Writing and Browsing Petri Net Based Hypertext.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 312–328 (1989)

The authors developed a new model of hypertext in pilot studies. The model, based on Petri nets, also represents the hyertext's browsing semantics. The model permits development of browsing and authoring systems that can incorparate the analytical techniques of Petri net theory and can also incorporate the user interface designs that have been developed for hypertext systems.

SUGASAWA, Y.; SEYA, K.: *Fault and Robust Detection of Error of a System with Intermittent Faults.* Electronics & Communications in Japan, Part 1: Communications, Vol. 71, No. 8, pp. 46–51 (1988)

This paper discusses the intermittent fault of a system and the robust detection of fault, and introduces a model interpretation by Petri net. An intermittent fault is produced in an intermittent period, and the detection of such a fault is considered very complex. In this paper, the intermittent faults are divided into benign faults and active faults.

TANKOANO, J.; DERNIAME, J.C.: *Structure Design of Distributed Systems Using Interpreted Petri Nets.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 329–347 (1989)

This paper presents an approach to internal structure design of distributed systems in the field of process control. Starting from an interpreted Petri net specifying the external behavior of a system, this approach (based on a set of decomposition rules) allows a systematic construction of structure specification, thus making unnecessary consistency verification.

TAUBNER, D.: *The Representation of CCS Programs by Finite Predicate/Transition Nets.* Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 348–370 (1989)

A construction is given which for all CCS programs (in which every choice and recursion starts sequentially) yields a finite and strict predicate/transition net. Consistency of this construction is proved not only with respect to the interleaving semantics but with respect to the distributed operational semantics.

TAUBNER, D.: *Finite Representations of CCS and TCSP Programs by Automata and Petri Nets.* Berlin: Springer Verlag, Lecture Notes in Computer Science, Vol. 369 (1989)
also: Dissertation, Techn. Universität München, Report TUM-I8817 (1988)

This work relates differnt approaches for the modelling of parallel processes. Within a uniform framework the syntax and the operational semantics of CCS and TCSP are explained. The authors consider both, Milner's interleaving semantics wich is based on infinite transition systems, as well as the new distributed semantics which is based on infinite safe nets. The main part of their work contains three syntax-driven constructions of transition systems, safe nets, and predicate/transition nets.

TIMPTE, B.S.: *Control of Search Techniques and Search Strategies Using Petri Nets in the National Bureau of Standard's Real-time Control System.* Lehigh Univ., Bethlehem, Master's Thesis (1987)

For over a decade, the National Bureau of Standards has been developing a control system to help facilitate the flexibility of machines and robots on the factory floor and to develop a true 'system'. The theory of control of this

system is called the Real-Time Control System (RCS). This thesis covers the application of the RCS to the search of an area by underwater autonomous vehicles (AUV) and addresses issues concerned with the valid techniques to be used in a two AUV search.

TRIVEDI, K.S.: *Effectiveness Evaluation of Fault-Tolerant Multiprocessor Systems.* Duke Univ., Durham, NC. Dept. of Computer Sciences (1988)

An important area of research is in the analysis of the coverage of a fault tolerant-system, that is, the probability that the system can recover from a fault. The author has studied a variety of models, from simple phase-type models to very complex stochastic Petri net models, and has investigated solution techniques for each model type.

TRÜMMER, H.: *Funktionsnetze: Ein auf Bausteinen basierender Ansatz zur Software-Konstruktion.* Dissertation, Techn. Universität Berlin, Fachbereich 20 (Informatik) (1988)

The Function-Net approach makes use of the channel/activity interpretation of Petri-Nets, and extends it through associating the nodes with abstract components of a software toolbox. Depending on the semantics object each node may represent an object such as a module, a program, a procedure, a file or a parameter. Through this assignment the theory of Petri-Nets is linked to the concepts of sofware engineering and to the ideas of method- and modelbased systems.

TYRRELL, A.M.: *Design of Fault-tolerant Software for Loosely Coupled Distributed Systems.* Aston Univ., Birmingham, Thesis (Ph. D.) (1987)

This thesis addresses the problem of error detection and recovery in distributed systems. Petri nets are used to represent the state and to solve state-reachability problems for concurrent systems. The dynamic behavior of the system can be characterized by a state-change table derived from the state reachability tree.

UNGERER, T.: *ASTOR - An Architecture of Special Purpose Processing Units with Distributed Control and Message Passing.* Microprocessing and Microprogramming, Vol. 23, No. 1-5, pp. 227–232 (1988)

The *A*ugsburg *ST*ructure *OR*iented computer architecture ASTOR describes the architecure of an asynchronous multiprocessor system at an abstract level. It consists of various special purpose processing units, working in parallel with decentralized control and communicating via message passing. Different special purpose processing units are used for the execution of program flow control constructs among which there is a new one, named dependency, which allows specifying parallel execution of machine instructions by some sort of Petri net.

USHIO, T.; MATSUMOTO, R.: *State Feedback and Modular Control Synthesis in Controlling Petri Nets.* Proc. 27th IEEE Conf. Decision Contr., Austin, TX, pp. 1502–1507 (Dec., 1988)

VAGIN, V.N.; ZAKHAROV, V.N.: ROZENBLYUM, L.Y.: *Logical Inference on Interpreted Petri Nets.* Soviet Journal of Computer and Systems Sciences, Vol. 26, No. 3. pp. 98–105 (1988)

The necessity of developing methods of parallel deduction is emphasized. The possibility of applying the methodology of Petri nets to the problem of deparallelization of logical (deductive) inference procedures is illustrated. Examples of the representation of deductive inference procedures in first-order logic by interpreted Petri nets are presented.

VALETTE, R.; CARDOSO, J.: ATABAKHCHE, H.; COURVOISIER, M.; LEMAIRE, T.: *Petri Nets and Production Rules for Decision Levels in F.M.S. Control.* Proc. of the 12th IMACS World Conf., Vol. 3. pp. 522–524 (1988)

VALMARI, A.: *State Space Generation: Efficiency and Practicality.* Tampere University of Technology, Publications 55, Thesis (1988)

The problem of making state space generation based analysis and verification of concurrent systems more efficient are considered. Two techniques are applied: Firstly, state spaces are not generated directly from system models. Insted, the models are compiled to another formalism called *base formalism*, state spaces are generated using the base formalism, and the results are interpreted in terms of the original formalism. Secondly, a new method for reducing the number of states that are generated is developed; it reduces state spaces by ignoring redundant execution order.

VALMARI, A.: *Stubborn Sets for Reduced State Space Generation.* Supplement to the Proc. of the 10th Int. Conf. on Application and Theory of Petri Nets, pp. 1–22 (1989)

The "stubborn set" theory and method for generating reduced state spaces is presented. The theory takes advantage of concurrency, or more generally, of lack of interaction between transitions, captured by the notion of stubborn sets. The method preserves all terminal states and the existence of nontermination; it can be used to detect violations of invariant properties, and to check the liveness of transitions.

VAN BILJON, W.R.: *Extending Petri Nets for Specifying Man-Machine Dialogues.* National Research Inst. for Mathematical Sciences, Pretoria, Report No. CSIR-TWISK-506 (1987)
also: Int. J. Man-Machine Studies, Vol. 28, pp. 437–455 (1988)

The requirements of man-machine dialogue specification techniques are examined. Petri Nets are identified as possible candidates for a modelling technique for dialogues on the basis of their applicability to concurrent, asynchonous systems. Labelled Petri Nets are extended to Nested Petri Nets, allowing transitions to invoke sub-Nets. It is shown that this extension allows Nested Petri Nets to generate at least the set of context-free languages.

VILLARROEL, J.L.; MARTINEZ, J.; SILVA, M.: *Graphic Design of Coordination Level in Production Systems.* Universidad de Zaragoza, departamento de ingenieria electrica e informatica, Research Report 89-05 (Jan., 1989)

This paper introduces GRAMAN, a graphical/textual design tool for describing coordination functions in manufactoring systems. The underlying methodology allows to design separately the manufactoring plant and the process plans. The latter are described by Petri nets. Their transitions model operations to be performed on the system resources. It is also allowed to describe them hierarchically.

VISWANADHAM, N.; NARAHARI, Y.: *Stochastic Petri Net Models for Performance Evaluation of Automated Manufacturing Systems.* Information and Decision Technologies, Vol. 14, No. 2, pp. 125–142 (1988)

In this paper, we show that generalized stochastic Petri nets (GSPNs) provide an effective modeling framework for performance evaluation of automated manufacturing systems. Using GSPNs, we study the performance of two representative systems: a manufacturing cell with multiple material handling robots and a simple flexible manufacturing system with three machines and two part types.

VOGLER, W.: *Failures Semantics and Deadlocking of Modular Petri Nets.* Acta Informatica, Vol. 26 — Springer-Verlag, pp. 333–348 (1989)

One can construct labelled P/T-nets in a modular fashion by exchanging subnets such that the behaviour of the whole net remains the same. The author investigates which subnets can be exchanged such that deadlockfreeness is preserved and shows that some variations of failures semantics are useful in this context.

WEINGAERTNER, S.T.: *Model of Submarine Emergency Decisionmaking and Decision Aiding.* Massachusetts Inst. of Tech., Cambridge Lab., Master's thesis (1986)

The effect of decision aid upon the workload and performance of a five member decisonmaking organization is investigated by way of information theoretic modeling and analysis. A generalized submarine ship control party performing the emergency control task is modeled using the Petri net formalism. The organization is then modified to incorporate a decision aid that provides a situation assessment.

WOHL, J.G.; TENNEY, R.R.: *Integrated Analysis Techniques for Command, Control, and Communications Systems. Volume 1. Methodology.* Alphatech. Inc., Burlington, MA., Report No. TR-293-1-VOL-1 (1987)

The Integrated Analysis Technique for Command, Control and Communications (C3) systems is described, and the results achieved to date are summarized. This includes the description of a symbolic language involving a major extension of Petri net theory, for modeling and evaluating the performance of manned C3 systems at any level of description or decomposition, and a convenient means for aggregating and modularizing system details without masking their impact on system performance.

WONG, C.Y.; DILLON, T.S.; FORWARD, K.E.: *Analysis of Dataflow Program Graphs.* Proceedings - IEEE International Symposium on Circuits and Systems — New York: IEEE, pp. 1045-1048 (1988)

An integrated approach to the modeling and performance evaluation of data-flow program graphs is developed by exploiting the similarities of such graphs to Petri nets (PNs). The basis of the methodology is a time-extended PN model, enhanced with data handling capabilities. Taking such an approach means that both data and control flow are combined in one model.

ZHANG, W.: *Automatic Robot Planning by Marked Petri Net.* Proc. of the Int. Workshop on Artif. Intell. for Ind. Appl. — New York: IEEE, pp. 358–361 (1988)

An algorithm of automatic robot planning according to a marked Petri net is presented. This net is configured by a goal structure and initial constraints, both in Petri net formalism. Based on a matrix representation, an assembly planning algorithm is developed.