Theoretical Computer Science 13 (1981) 109–136

SYSTEM MODELLING WITH HIGH-LEVEL PETRI NETS

H.J. GENRICH and K. LAUTENBACH

Institut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung, D-5205 St. Augustin 1, Fed. Rep. Germany

Abstract. The paper presents a high-level Petri net model of concurrent systems called *predicate/transition-nets* (PrT-nets). Its places represent variable properties of, or relations between, individuals; they are 'predicates' with variable extension. The transitions represent classes of elementary changes of those extensions. The model is introduced on the basis of a simple example from resource management. The central part of the paper is devoted to linear-algebraic techniques for verifying invariant assertions, yielding a calculus of *S*-invariants for PrT-nets. Finally, these modelling and analysis techniques are applied to a scheme for organizing a distributed data base taken from literature.

1. Introduction

When Petri first introduced 'his' nets of places ("Stellen") and transitions ("Transitionen") in [17], they served as a vehicle for developing a non-idealizing approach to concurrency and information flow in organizational systems. Later the possibility of using these nets, by then called *Petri nets*, in practical systems design was beautifully demonstrated by Holt et al. [8], Shapiro and Saint [27] and Patil [16].

Encouraged by this and the inspired writings of Holt and Commoner [9] a number of attempts were made to put Petri nets to the same kind of use, but in more ambitious settings. Here, the user of Petri nets was quickly and rudely brought up to face the fact that he was being forced to deal with rather large systems at an unacceptable level of detail. At this point, a number of people became disillusioned with Petri nets and promptly dropped the idea of considering them any further. Others persevered and developed some very useful extensions and derivations of the original model to fit their specific needs. (A typical example is the evaluation net model developed by Noe and Nutt [15].)

Recognizing well in time that a variety of net based models are needed in practice Petri proposed in [19] to interconnect the various models that may arise by means of meaning preserving transformations of 'inscribed nets'. The underlying idea of his proposal is rather simple: On any given conceptual level, the structure of a system (its decomposition into components) is represented by a simple formal object called a *(directed) net* – a natural generalization of the notion 'directed graph'. All other aspects of the modelled systems (function, purpose, behaviour, . . .) are expressed by assigning various kinds of 'inscriptions' to the elements of the net. Such inscriptions may be natural language texts, formal expressions, special symbols, or all kinds of graphical devices. Their semantics are deduced, by means of completion and

abstraction, from an axiomatically defined basic interpretation of nets, the *condition/event-system* model [20].

Petri [21, 22] called this 'programme' *General Net Theory* (G.N.T.) indicating its origin in the 'special' net theory of the token game played in place/transition-nets (c.f. [21]). Within this programme, for example, two classes of system invariants were disclosed through the vehicle of completion, the *enlogic structure* and the *synchronic structure* of systems [20]. By means of abstraction, e.g., the existence of two different forms of *information flow* in systems, called *flux* and *influence*, could be shown [18, 21]. The purely mathematical aspects of the programme are treated within the *category of net morphisms* [19].

In this paper we present as a new result of G.N.T. a transition net model with rather sophisticated inscriptions whose semantics are derived from 'ordinary' Petri nets, in a strictly formal way, through an equivalence transformation of inscribed nets. This model combines and completes a great deal of existing material: the net representation of first-order predicate logic derived from the enlogic structure by Genrich and Thieler-Mevissen [6, 28]; the transition nets with coloured tokens introduced by Lautenbach and investigated by Schiffers and Weide [24, 25]; and the transition nets with complex conditions and transmissions used by Shapiro [26]. A similar but independent attempt was made by Nieters in [14]; his TL-nets may now be viewed as a rather complicated special case of our model. At a very early stage, Jensen learned of our work and developed it further in a very interesting way [10]. It should be worthwhile to compare his approach with ours in a future paper.

Our model which we shall call *predicate/transition-nets* (PrT-nets) adds to the modelling power and complexity of Petri nets a new dimension, namely the formal treatment of *individuals* and their *changing properties and relations*. We shall see that this step is comparable – quantitatively and qualitatively – to that of going from propositional logic to first-order predicate logic.

Assuming some familiarity with Petri nets, we introduce in the next section the model on the basis of a simple example taken from the realm of resource management. The central part of the paper is devoted to the task of transferring the calculus of S-invariants, a powerful linear-algebraic technique for verifying invariant assertions known from [11, 12], to the new model. Finally, in Section 4, we apply the apparatus developed so far to the analysis and verification of a scheme for organizing a distributed database taken from literature [2, 13].

This paper is a greatly revised version of a paper [4] presented at the Evian Conference on Semantics of Concurrent Computation which was also included into the course material of the Advanced Course on General Net Theory of Processes and Systems held in Hamburg in October 1979 [5].

2. Predicate/transition-nets

Following the programme of G.N.T. sketched above we start the development of our high-level Petri net model with the following

Definition 2.1. A triple $N = (S, T; F)$ is called a (*directed*) *net* iff

(1) $S \cap T = \emptyset$,

(2) $S \cup T \neq \emptyset$,

(3) $F \subseteq S \times T \cup T \times S$,

(4) $\mathrm{dom}(F) \cup \mathrm{cod}(F) = S \cup T$.

For a given net $N = (S, T; F)$ we call

(5) $X := S \cup T$ the set of (S- or T-) elements of N, and

(6) F the *flow* relation containing the *arcs* of N;

For an element $x \in X$,

(7) $\bullet x := \{y \mid (y, x) \in F\}$ and $x \bullet := \{y \mid (x, y) \in F\}$ are called the *preset* and *postset* of x, respectively.

In Fig. 1, we see the graphical representation of a net. The S-elements are represented by circles ○, the T-elements by boxes □, and an arc $(x, y) \in F$ is represented by an arrow leading from the image of x to the image of y.

Additionally, the net shown in Fig. 1 is inscribed in two ways. First, the S- and T-elements are labelled by certain identifiers which will allow us to talk about the net and its properties. Second, some of the S-elements are marked by a 'token'. Thus the circles serve as *places* for tokens which allow us to play the 'token game' on the net and to simulate the behaviour of the simple system which we associate with Fig. 1.

To this end, we interpret the places as elementary *conditions* of the system, i.e. atomic propositions about the system with a changing truth value. In a given *case*, the presence or absence of a token on a place represent the holding or non-holding of the corresponding condition, respectively.

The boxes represent elementary changes, called *transitions*, in the holding and non-holding of conditions. For a given transition x, we call the elements of its preset $\bullet x$ and its postset $x \bullet$ the *preconditions* and *postconditions* of x, respectively. A transition has a *chance to occur* (to 'fire') in a given case if all its preconditions hold (carry a token) and all its postconditions do not. By an occurrence of a transition, all its preconditions cease to hold and all its postconditions begin to hold. Systems which are modelled in this way will be called CE-systems (systems of conditions and events).

In the case shown in Fig. 1, two transitions, $1l$ and $1r$, may occur, and they may occur *concurrently* since they are completely separated; they have no pre- or

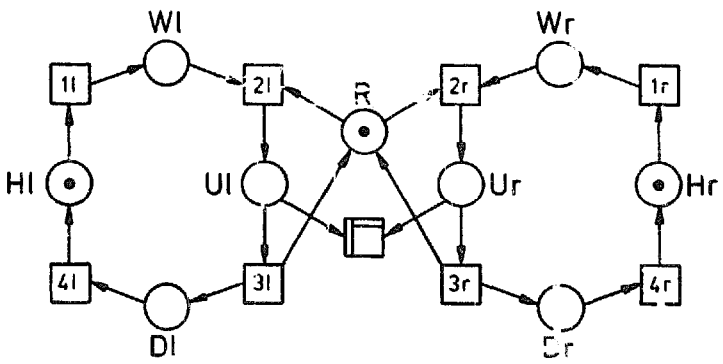


Fig. 1.

postconditions in common. If both $1l$ and $1r$ occur, the result is a new marking, in which Wl, Wr, and R carry a token. In this case, both $2l$ and $2r$ may singly occur; they are, however, in *conflict*: one transition looses the chance to occur by an occurrence of the respective other. Assuming that $2l$ occurs, $3l$ is the only transition which may occur next returning the token to the place R. Now $4l$ and $2r$ may occur concurrently.

Continuing this simulation of the processes which are supported by the system and exhausting all possibilities, we shall see that all transitions will get a chance to occur, with one exception: the transition ☐ is 'dead', i.e. it has no chance to occur at all.

We call those transitions which get a chance to occur *events*; the dead transitions we call *facts*. Facts are *conceivable* but factually impossible changes of the system. They play as an important role in the specification of systems as events: They represent *invariant assertions* about systems. In our example, ☐ represents 'the fact' that $\neg(Ul \wedge Ur)$ is an invariant assertion expressing the *mutual exclusion* of conditions Ul and Ur.

Events and facts are two classes of the *enlogic structure* of CE-systems which classifies *all* conceivable changes [20]. As in our example, we shall use the symbol ☐ from now on for facts in general. In order to emphasize the important role of facts in the theory of CE-systems (systems of conditions and events), we state without proof the following

Theorem 2.1. *Every dead transition of a CE-system represents an invariant (propositional) assertion built from conditions; and every such invariant assertion can be represented by a set of dead transitions* [20].

We have seen that we can interpret Fig. 1 as the net representation of a little system into which two sequential components are imbedded in such a way that they never can be in their respective 'critical' phases at the same time. The place R may be viewed as representing the availability of a resource which is needed by both components but can only be used exclusively.

In Fig. 2, we see the same net as in Fig. 1 but with slightly different inscriptions. First, the places can carry more than one token. Thus they can no longer be interpreted as conditions which either hold or don't. Rather they may be considered as non-negative integer quantities, the number of tokens expressing their current value. Second, arcs may be labelled by a positive integer expressing their *multiplicity*.

This inscribed net is a simple example of a Petri net or, as we call it more precisely, a *place/transition-net* (PT-net). It differs from the CE-system model in that the places may carry more than one token, and that a transition may remove or add more than one token from or to the places, according to the multiplicity of the respective in- and outgoing arcs. One may or may not wish to assign *capacities* to the places, i.e. a maximal number of tokens which cannot be exceeded by transition occurrences. Accordingly, one distinguishes between the *strict* and the *weak* transition rule. (Thus we can say that CE-systems are modelled by PT-nets with capacity one.) If we assign capacities to the places of our model in Fig. 2 consistently with the initial marking,

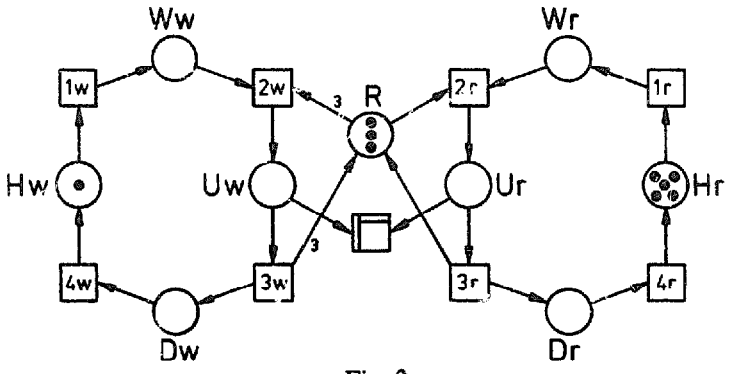


Fig. 2.

there is no danger of exceeding these capacities: The system is *safe* with respect to the capacities.

As one might have guessed already, Fig. 2 represents a simple version of the so-called reader/writer system. There are five components ('readers') which may share the resource, and one component ('writer') which can use the resource only exclusively. The three tokens on R indicate that up to three components may use the resource at the same time.

Again the main specification, the essential restriction to the unco-ordinated behaviour of the two components, is expressed by the single dead transition ⊟. If we do not trust in simulation for verifying that this specification is met, we can use the method of *S-invariants* in order to prove it.

This method profits from the linear-algebraic version of the token game based upon the *incidence matrix* of (loop free) Petri nets. Briefly, the incidence matrix of a Petri net is a matrix with rows for each place and columns for each transition. The entry for row s and column t is n if there is an arc from t to s with multiplicity n, and it is $-n$ if there is an arc from s to t; otherwise the entry is zero.

In Fig. 3 we see the incidence matrix of the net shown in Fig. 2 (zeroes being omitted) together with the vector representation of the initial marking (M_0), and a vector i which has the following property: The linear combination of the rows of the incidence matrix using the corresponding entries of i as coefficients is the zero row. It is now easy to prove that for an arbitrary marking M which can be derived from M_0 by means of occurrences of transitions, the inner product of M with i equals the inner

	1w	2w	3w	4w	1r	2r	3r	4r	M_0	i
Hw	−1			1				1		
Ww	1	−1								
Uw		1	−1							3
Dw			1	−1						
R		−3	3			−1	1		3	1
Hr					−1			1	3	
Wr					1	−1				
Ur						1	−1			1
Dr							1	−1		

Fig. 3.

product of M_0 with i:

$$i^T \bullet M = i^T \bullet M_0. \qquad (i)$$

For example, (1) becomes

$$3 \bullet M(Uw) + 1 \bullet M(R) + 1 \bullet M(Ur) = 1 \bullet M_0(R) = 3.$$

Taking (1) for granted (we shall return to it in the next section) it follows immediately that it is impossible to have a token both on Uw and Ur if we observe the rules of the game.

The way in which we have represented the reader/writer system by means of a place/transition-net does not allow to identify the individual readers. It is only possible to determine the number of readers which are in a specific phase in a given case. In order to retain the identity of the readers, we have to 'unfold' the reader part of Fig. 2 as shown, in a schematic form, in Fig. 4.

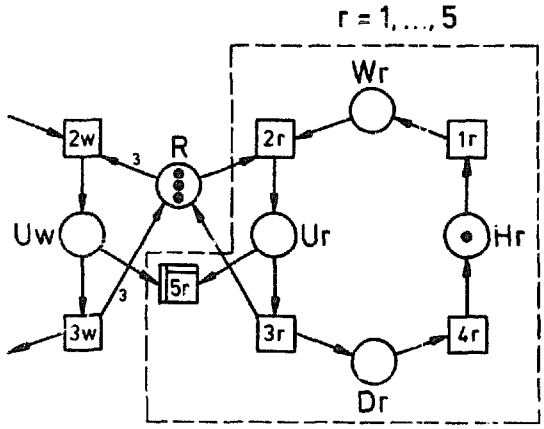


Fig. 4.

We are now going to show a technique for maintaining the identity of components even in a highly condensed representation. As an example, we take from [26] a slight generalization of our last example such that all components show exactly the same behaviour. We consider a community C of users of a single resource that may be used in either of two modes, 'exclusive' or 'shared'. Again there is, independently of the number of users, an upper limit N of concurrent shared usages.

In Fig. 5 we see a net representation of this system for a community $C = \{a, b, c\}$ and $N = 2$. The two different modes are denoted by the two identifiers s (shared) and e (exclusive) forming the set $M = \{s, e\}$. The system is modelled in terms of four predicates H, W, U, and D, and an integer quantity R:

$H\langle u\rangle \Leftrightarrow$ user u has nothing to do with the resource;
$W\langle u, m\rangle \Leftrightarrow$ user u wants to use the resource in mode m;
$U\langle u, m\rangle \Leftrightarrow$ user u is using the resource in mode m;
$D\langle u, m\rangle \Leftrightarrow$ user u has finished using the resource in mode m;
$R\langle\,\rangle =$ the number of times the resource is still available for shared usage.

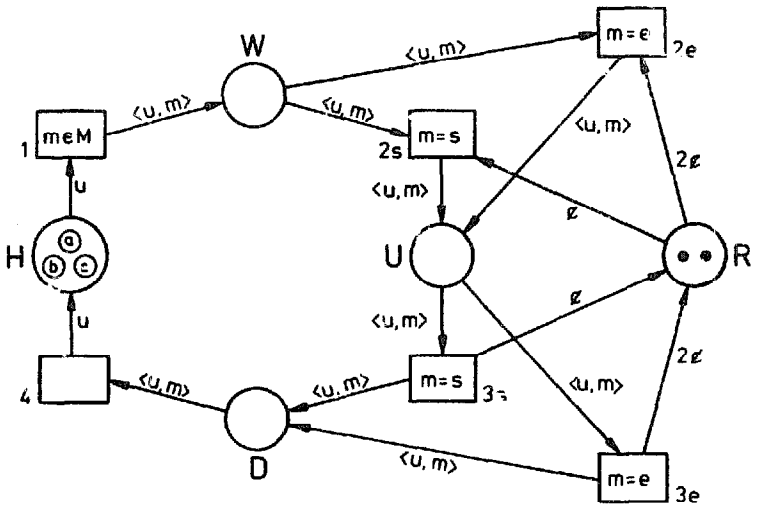


Fig. 5.

The predicates *H*, *W*, *U*, and *D* are schemes of conditions, i.e. of atomic propositions with changing truth values. Thus their extensions, the sets of (tuples of) individuals they map onto 'holding', may change. Instead of marking a place with simple tokens, we now mark the predicates with their current extension. In the generalization from conditions to integer quantities these extensions may contain the same tuple more than once; they are no longer sets but formal sums ('multisets') of tuples of individuals. In order to include ordinary places as a special case, we treat them as zero-place predicates and denote the 'zero-tuple' by ε.

Not only the 'places' of Fig. 5 represent schemes of places, the 'transitions' of Fig. 5 represent schemes of transitions, too. The arcs are labelled by (formal sums of) tuples of individual variables. An instance of a single transition in Fig. 5 is generated by consistent substitution: all variables at this transition are replaced by individual symbols, and all occurrences of the same variable are replaced by the same symbol. However, only those instances of the transition belong to the system which satisfy the logical formula inscribed to the transition (no inscription means no restriction). In Fig. 6 we show the result of applying this rule to transitions 1 and 2*s* of Fig. 5.

The complete expansion of Fig. 5 into an ordinary place/transition-net representing *the same system* is shown in Fig. 7. Its size demonstrates rather drastically the advantage of the representation used in Fig. 5 (try to imagine how Fig. 7 would look like for ten or a hundred users).

In order not to overburden Fig. 5, we have not represented the restriction meant by the terms 'shared use' and 'exclusive use'. This is shown in Fig. 8 separately.

By the graphical symbol ⊟ we denote again a dead transition or now, more precisely, a scheme of dead transitions. Then the diagram in Fig. 8 reads as follows: In no case of the system, place *U* carries two pairs one of which has an *e* at its second position. In other words, if one user is using the resource in mode 'exclusive', there is no other user using the resource at all.

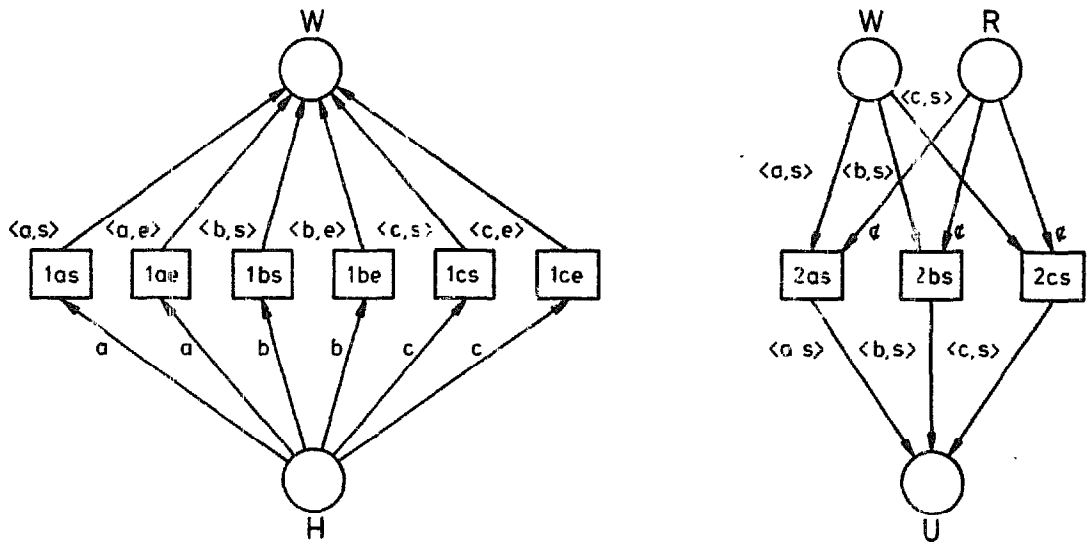


Fig. 6.

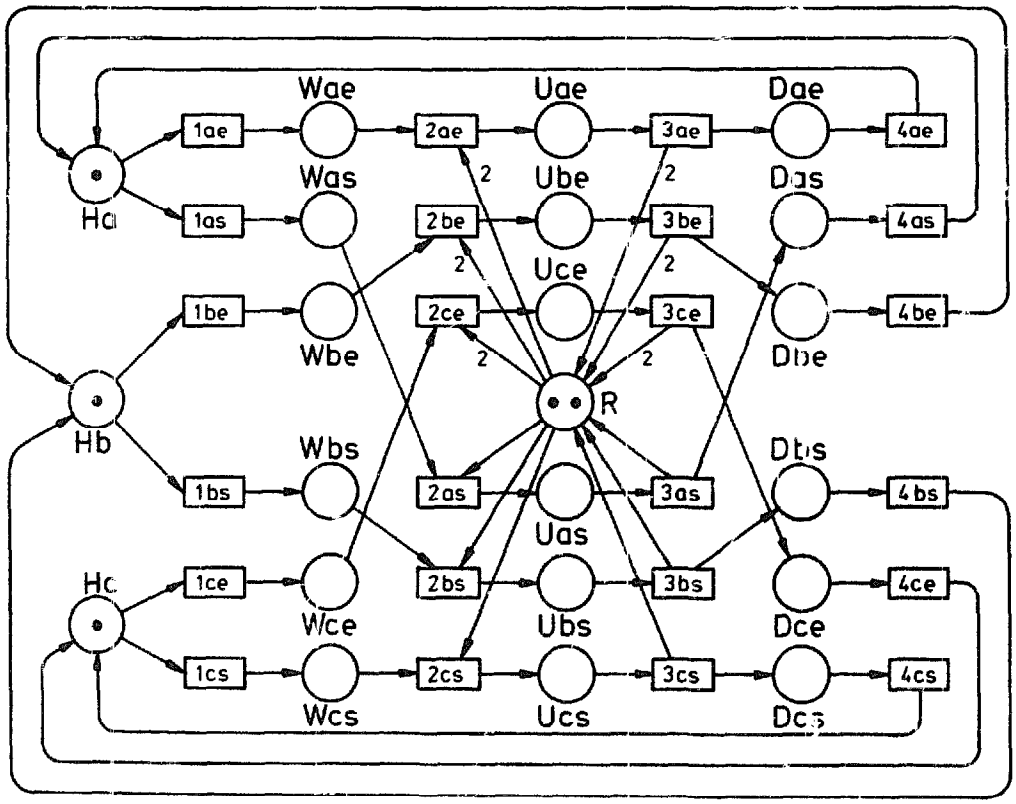


Fig. 7.

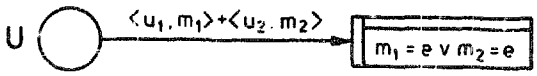


Fig. 8.

The inscribed nets shown in Fig. 5 and Fig. 8 are examples of the net model which we shall call, from now on, *predicate/transition-nets* (PrT-nets). We have seen how the size of a net model of a system can be considerably reduced by using these 'first-order Petri nets'. That the increase in modelling power is indeed comparable to that when going from propositional logic to first-order logic, has been shown in [6] from where we get the following

Theorem 2.2. *Every formula of first-order predicate logic can be equivalently represented by a set of dead transitions of a PrT-net with all places having the capacity one (no tuple can appear more than once on any given place).*

Before we enter into the more formal presentation of the PrT-net model, we will show one more possibility for compressing the net in Fig. 5, within our technique. The two transition $2s$ and $2e$ are connected to the same predicates in the same way. They only differ with respect to the inscriptions assigned to them and to the adjacent arcs. The same applies to transitions $3s$ and $3e$. We now allow to denote arc labels by expressions, like conditional expressions or functional terms. (It is this device Jensen [10] has based his technique upon.) Then, with

$$F(m) := \begin{cases} 1\ell, & m = s, \\ 2\ell, & m = e \end{cases}$$

the diagram of Fig. 5 can be equivalently transformed as shown in Fig. 9.

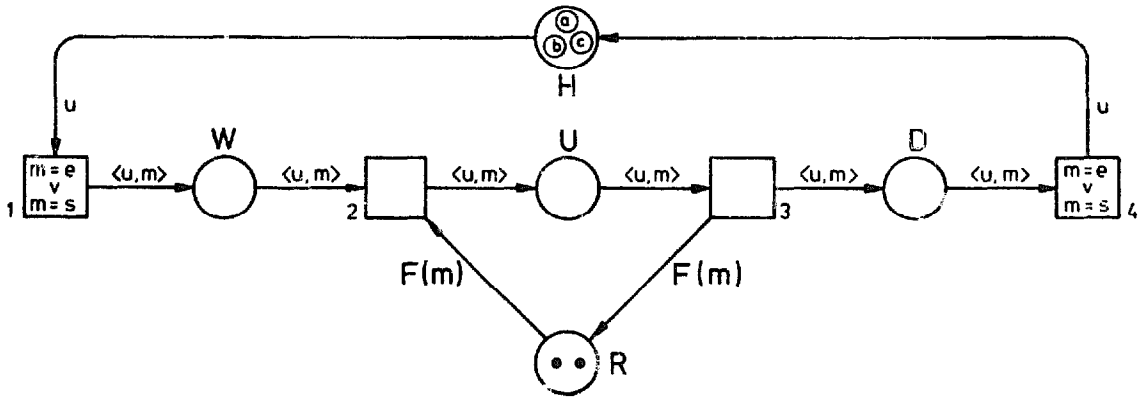


Fig. 9.

We will now summarize and formalize the result of our introduction of a new, higher-level Petri net model in the following

Definition 2.2. A *predicate/transition-net* (*PrT-net*) consists of the following constituents:

(1) A *directed net* $(S, T; F)$ where
- S is the set of predicates ('first-order' places) ○,
- T is the set of (schemes of) transitions □.

(2) A structure Σ consisting of some sorts of individuals together with some operations and relations.

(3) A labelling of all arcs with (an expression denoting) a formal sum of tuples of variables, whose length n is the 'arity' of the predicate connected to the arc. The zero-tuple indicating a no-argument predicate (an ordinary place) is denoted by the special symbol ϵ. For examples see Fig. 10(a).

Fig. 10(a).

(4) An inscription on some transitions being a logical formula built from the operations and relations of the structure Σ; variables occurring free in a formula have to occur at an adjacent arc.

If a formula at a box has the form $v = t \wedge \cdots$, where v is a variable and t a term, all occurrences of v at the transition may be replaced by copies of t. For examples see Fig. 10(b).

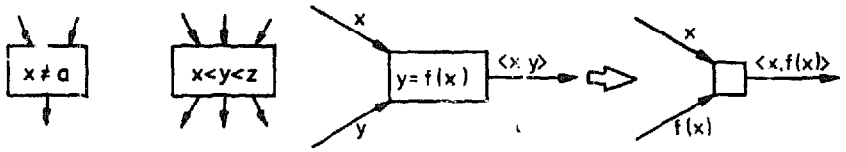


Fig. 10(b).

(5) A *marking* M_0 of predicates of S with formal sums of n-tuples of individual symbols. (We call these tuples just *items*). For examples see Fig. 10(c).

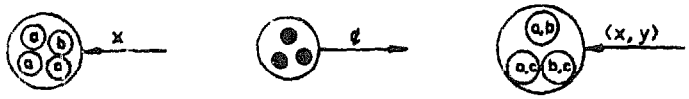


Fig. 10(c)

(6) A function K which assigns to the predicates an upper bound for the number of copies of the same item which it may carry. $K(s)$ may be called the *capacity* of s. (Ignoring capacities can be expressed by infinite capacities.)

(7) The *transition rule* '$\rightsquigarrow$' for predicate/transition-nets: Each element of T represents a class of possible indivisible changes of the markings of the adjacent predicates. Such a change consists of removing (○→□) and adding (□→○) copies of items from/to places according to the expressions labelling the arcs. It *may* occur whenever, for an assignment of individuals to the variables which satisfies the formula inscripted to the transition, all input predicates carry enough copies of proper items and for no output predicate the capacity K is exceeded by adding the respective copies of items. The set of all markings connected to M_0 through such occurrences of transitions is denoted by $[M_0]$.

For an example see Fig. 10(d). For a structure $(\{a, b, c\}; \langle :=$ alphabetical ordering) and $K = 3$, two of the nine instances of the transition are enabled under the marking

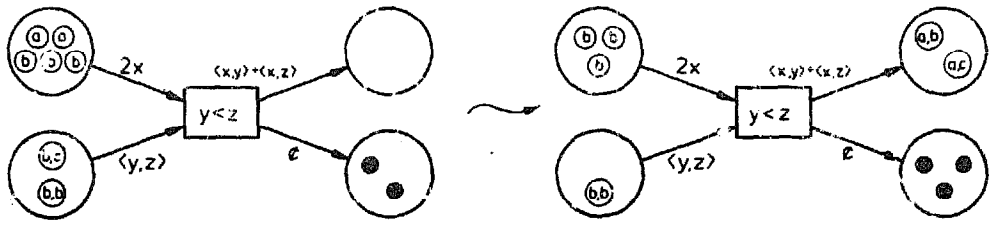


Fig. 10(d).

shown on the left side. Due to conflict, however, at most one will occur. For the assignment $(x, y, z) \leftarrow (a, b, c)$ the resulting marking is shown on the right side.

We shall see that formal sums in items play the same role in our model as integers play in ordinary Petri nets. In fact, the transfer of the linear-algebraic techniques for Petri nets to predicate/transition-nets is based exactly upon this generalization of the integers. Therefore we introduce here a minimum of notation needed in the next sections. The formal apparatus is that of polynomial rings over commutative rings; it can be found in any book on algebra, e.g. [7]. For the purpose of this paper, multilinear forms, i.e. formal sums of products of different items, will suffice. In the further development of the linear theory of PrT-nets, however, polynomials in general will be needed.

(1) An integer polynomial in n variables $p \equiv p(v_1, \ldots, v_n)$ is a sum $\sum p_{k_1 \cdots k_n} \bullet v_1^{k_1} \bullet \cdots \bullet v_n^{k_n} \mid k_1 \geq 0, \ldots, k_n \geq 0$, where each $p_{k_1 \cdots k_n}$ is an integer called the *coefficient* of the product $v_1^{k_1} \bullet \cdots \bullet v_n^{k_n}$.

(2) In our case, the variables are the items, i.e. tuples of individual names. The empty item ε is the unit element of the ring (the 0th power of any item). The integers are identified with polynomials of degree 0 (in ε only).

(3) For two polynomials $p = p(v_1, \ldots, v_n)$ and $q = q(v_1, \ldots, v_n)$ we write $p \leq q$ iff $p_{k_1 \cdots k_n} \leq q_{k_1 \cdots k_n}$ for all $k_1, \ldots, k_n \geq 0$.

(4) For a polynomial $p = p(v_1, \ldots, v_n)$ we denote by $|p|$ the (*unit*) *value* (sum of coefficients) $p(1, \ldots, 1)$.

(5) For a vector (matrix) of polynomials, its value is defined as the vector (matrix) of the values of its elements. If C and D are matrices of polynomials, then $|C \bullet D| = |C| \bullet |D|$. In the same way, if x and y are vectors in polynomials, then $|x * y| = |x| * |y|$ for the inner product.

(6) To a set of items we assign its *characteristic* polynomial by means of an operator π: $\pi(X) := \sum x \mid x \in X$.

(7) The *incidence matrix* of a pure ($F \cap F^{-1} = \emptyset$) predicate/transition-net is a mapping C from $S \times T$ into integer polynomials such that

$$C(s,t) := \begin{cases} -l, & l \text{ is the label of } (s,t) \in F, \\ l, & l \text{ is the label of } (t,s) \in F, \\ 0, & \text{otherwise.} \end{cases}$$

3. Invariant assertions and linear algebra

Let C be the incidence matrix of a predicate/transition-net PN; then a vector i of arc labels is called an *S-invariant* of the net PN if $C^T \bullet i = 0$ where C^T denotes the transpose of C (cf. [3, 11, 12]. If $i(p) \neq 0$ for some place p, we call $i(p)$ the *weight* of p in i, and i an S-invariant *through* p.

The unit value $|C|$ of C is the incidence matrix of an ordinary Petri net $|PN|$, the *(unit) value of PN*. Because of

$$|C^T \bullet i| = 0 \Rightarrow |C|^T \bullet |i| = 0$$

we see that the value of an S-invariant is an S-invariant of the value (of the net).

In place/transition-nets we take advantage of equations of the following kind:

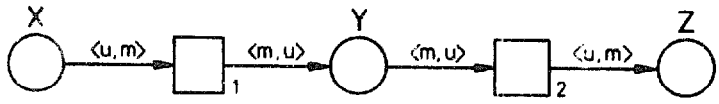
$$i^T \bullet M = i^T \bullet M_0 \tag{2}$$

for an S-invariant i and all $M \in [M_0]$ ($[M_0]$ denotes the set of all markings derivable from M_0) which states that the inner product of an S-invariant with the elements of one marking class is an invariant quantity. The unknowns of (2) are the elements of M because i and M_0 consist of integer constants. The normal application of (2) is to assume values for some elements of M and then to try to solve (2). If (2) is not solvable, then we know for sure that no marking $M \in [M_0]$ exists for which the assumption holds. On the other hand, every solution of (2) shows that there exist markings for which the assumption is satisfied. Moreover, our knowledge about such markings has grown.

The interpretation of (2) for PrT-nets is more complicated. In order to interpret (2) we should first show by means of a simple but characteristic example (Fig. 11) that like in place/transition-nets there exists a linear relationship between initial markings and their follower markings:

Let C be the incidence matrix of a PrT-net PN with the initial marking M_0, and let $M \in [M_0]$ be a follower marking of M_0; then there exists a linear representation

$$C \bullet f = M - M_0. \tag{3}$$



C	1	2	i	M_0
X	$-\langle u, m\rangle$		$\langle m, u\rangle$	$\langle a, e\rangle + \langle u, s\rangle + \langle b, e\rangle + \langle b, s\rangle$
Y	$\langle m, u\rangle$	$-\langle m, u\rangle$	$\langle u, m\rangle$	
Z		$\langle u, m\rangle$	$\langle m, u\rangle$	

Fig. 11.

In Fig. 11 we see a PrT-net and in a table the incidence matrix C, an S-invariant i, and the vector representation of the initial marking M_0. The S-invariant property of i can easily be checked by showing that the linear combination of the rows, using the corresponding entries of i as coefficients, is the zero row. It is obvious that the commutativity of the formal product is imperative.

There are four individual symbols a, b, e, s which are partitioned into two sorts, $\{a, b\}$ designated by the variable u, and $\{e, s\}$ designated by the variable m. X, Y, Z are binary predicates with variable extension. But note that the pairs belonging to these extensions are of different sorts: the extensions of X and Z are of sort $\langle u, m\rangle$, the extensions of Y are of sort $\langle m, u\rangle$. (In Fig. 11 and in all similar figures we have omitted zero-entries.)

First of all, we want to show (3) and (2) for the simplest case that exactly one transition fires exactly once: transition 1 fires once and takes $\langle a, e\rangle$ from X and puts $\langle e, a\rangle$ on Y. The resulting marking is

$$M_1 = \begin{pmatrix} \langle a, s\rangle + \langle b, e\rangle + \langle b, s\rangle \\ \langle e, a\rangle \\ 0 \end{pmatrix}.$$

The difference between the two markings is

$$\Delta_1 M := M_1 - M_0 = \begin{pmatrix} -\langle a, e\rangle \\ \langle e, a\rangle \\ 0 \end{pmatrix} := \begin{pmatrix} -\langle u, m\rangle \\ \langle m, u\rangle \\ 0 \end{pmatrix}_{\substack{(u=a)\\(m=e)}}$$

$$= \left[C \bullet \begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix}\right]_{\substack{(u=a)\\(m=e)}} =: C \bullet \left[\begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix}_{\substack{(u=a)\\(m=e)}}\right].$$

So we get (3) in the following form:

$$C \bullet f := C \bullet \left[\begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix}_{\substack{(u=a)\\(m=e)}}\right] = \Delta_1 M.$$

In this 'linear' representation, the multiplication of C by f means first to select column 1 of C and then to substitute u by symbol a and m by e which corresponds exactly to the single firing of transition 1 mentioned above.

Next, we want to verify and interprete (2). We start with

$$i^{\mathrm{T}} \bullet \Delta_1 M = -\langle m, u\rangle \bullet \langle a, e\rangle + \langle u, m\rangle \bullet \langle e, a\rangle.$$

Obviously, in order to get (2) we have in i to substitute $u = a$, $m = e$. This, however, is nothing else but accomplishing the following *consistent substitution*:

In elementary products we complete partial bindings.

For example, in $\langle m, u\rangle \bullet \langle a, e\rangle$ u and m are partially bound by a and e, resp. The completion of this bindings yields $\langle e, a\rangle \bullet \langle a, e\rangle$. So we finally have for (2)

$$\begin{aligned} i^{\mathrm{T}} \bullet \Delta_1 M &= -\langle m, u\rangle \bullet \langle a, e\rangle + \langle u, m\rangle \bullet \langle e, a\rangle \\ &= -\langle e, a\rangle \bullet \langle a, e\rangle + \langle a, e\rangle \bullet \langle e, a\rangle \\ &= 0, \quad \text{because the product } \bullet \text{ is commutative.} \end{aligned}$$

What we have found is, how to interprete (2) and (3) in the simplest case of one transition firing once, and that this interpretation is an obvious consequence of the transition rule. The extension of these observations for general firing sequences leads for (2) and (3) to *linear combinations* of the elementary case.

Let transition 1 fire for $\langle u, m\rangle = \langle a, e\rangle, \langle a, s\rangle, \langle b, e\rangle$ and transition 2 for $\langle m, u\rangle = \langle e, a\rangle, \langle s, a\rangle$ and let the resulting marking be M_2. The difference is

$$\Delta_2 M := M_2 - M_0 = \begin{pmatrix} -\langle a, e\rangle - \langle a, s\rangle - \langle b, e\rangle \\ \langle e, b\rangle \\ \langle a, e\rangle + \langle a, s\rangle \end{pmatrix}.$$

Now let be

$$\Delta_{21} M := \begin{pmatrix} -\langle a, e\rangle \\ \langle e, a\rangle \\ 0 \end{pmatrix}, \qquad \Delta_{22} M := \begin{pmatrix} -\langle a, s\rangle \\ \langle s, a\rangle \\ 0 \end{pmatrix}, \qquad \Delta_{23} M := \begin{pmatrix} -\langle b, e\rangle \\ \langle e, b\rangle \\ 0 \end{pmatrix},$$

$$\Delta_{24} M := \begin{pmatrix} 0 \\ -\langle e, a\rangle \\ \langle a, e\rangle \end{pmatrix}, \qquad \Delta_{25} M := \begin{pmatrix} 0 \\ -\langle s, a\rangle \\ \langle a, s\rangle \end{pmatrix}.$$

Then we have $\Delta_2 M = \Delta_{21} M + \Delta_{22} M + \Delta_{23} M + \Delta_{24} M + \Delta_{25} M$ which represents a partition of our firing sequence into single firings. This yields according to the above for (3):

$$\begin{aligned} \Delta_2 M &= C \bullet f_1 + C \bullet f_2 + C \bullet f_3 + C \bullet f_4 + C \bullet f_5 \\ &= C \bullet \left[\underset{\substack{(u=a)\\(m=e)}}{\begin{pmatrix} 1 \\ 0 \end{pmatrix}} + \underset{\substack{(u=a)\\(m=s)}}{\begin{pmatrix} 1 \\ 0 \end{pmatrix}} + \underset{\substack{(u=b)\\(m=e)}}{\begin{pmatrix} 1 \\ 0 \end{pmatrix}} + \underset{\substack{(u=a)\\(m=e)}}{\begin{pmatrix} 0 \\ 1 \end{pmatrix}} + \underset{\substack{(u=a)\\(m=s)}}{\begin{pmatrix} 0 \\ 1 \end{pmatrix}} \right] \\ &=: C \bullet f. \end{aligned}$$

For (2) we find

$$\begin{aligned} i^{\mathrm{T}} \bullet \Delta_2 M = i^{\mathrm{T}} \bullet \Delta_{21} M + i^{\mathrm{T}} \bullet \Delta_{22} M + i^{\mathrm{T}} \bullet \Delta_{23} M + i^{\mathrm{T}} \bullet \Delta_{24} M + i^{\mathrm{T}} \bullet \Delta_{25} M \\ = -\langle m, u\rangle \bullet \langle a, e\rangle + \langle u, m\rangle \bullet \langle e, a\rangle - \langle m, u\rangle \bullet \langle a, s\rangle + \langle u, m\rangle \bullet \langle s, a\rangle \\ -\langle m, u\rangle \bullet \langle b, e\rangle + \langle u, m\rangle \bullet \langle e, b\rangle - \langle u, m\rangle \bullet \langle e, a\rangle \\ +\langle m, u\rangle \bullet \langle a, e\rangle - \langle u, m\rangle \bullet \langle s, a\rangle + \langle m, u\rangle \bullet \langle a, s\rangle. \end{aligned}$$

By applying the rule of consistent substitution to every elementary product and the commutative law we get

$$i^T \bullet \Delta_2 M = 0.$$

What we have achieved so far is an interpretation of (2) and (3) for PrT-nets. But, even more importantly, we have implicitely introduced a *calculus for S-invariants*. This calculus consists of the following rules:

commutative, associative, distributive law (abreviated C, A, D) for the formal product $\bullet$, and the rule of *consistent substitution* (*S*) for *elementary* products.

In addition, we need the *reverse of substitution* (R) and a *division rule* (V) which we will introduce by means of the following example: For the net of Fig. 11 we use again M_0 as initial marking. We want to completely know a follower marking M that is incompletely given by:

$$M(X) = 0, \qquad M(Y) = \langle s, b\rangle.$$

If M is indeed a follower marking of M_0, we may apply (2) (if not, applying (2) leads to a contradiction!).

$$i^T \bullet M_0 = \langle m, u\rangle \bullet (\langle a, e\rangle + \langle a, s\rangle + \langle b, e\rangle + \langle b, s\rangle)$$

$$= i^T \bullet M = \langle u, m\rangle \bullet \langle s, b\rangle + \langle m, u\rangle \bullet M(Z).$$

Substitution, reverse substitution, and commutativity lead to

$$\langle u, m\rangle \bullet \langle s, b\rangle \underset{(S)}{=} \langle b, s\rangle \bullet \langle s, b\rangle \underset{(R)}{=} \langle b, s\rangle \bullet \langle m, u\rangle \underset{(C)}{=} \langle m, u\rangle \bullet \langle b, s\rangle.$$

Next, we use the distributive law:

$$\langle m, u\rangle \bullet (\langle a, e\rangle + \langle a, s\rangle + \langle b, e\rangle + \langle b, s\rangle) \underset{(D)}{=} \langle m, u\rangle \bullet (\langle b, s\rangle + M(Z)).$$

We can divide the equation by the factor $\langle m, u\rangle$ according to the following *rule* (*V*):

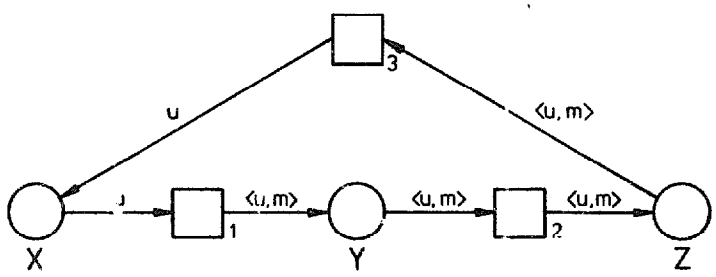
In an equation, all instances of a given sort can be replaced by their value. (Note: $\langle u, m\rangle$, $\langle a, m\rangle$, $\langle b, e\rangle$ are instances of sort $\langle u, m\rangle$, and $|\langle u, m\rangle| = |\langle a, m| = |\langle b, e\rangle| = 1$).

$$\langle a, e\rangle + \langle a, s\rangle + \langle b, e\rangle + \langle b, s\rangle \underset{(V)}{=} \langle b, s\rangle + M(Z).$$

The result then is

$$M(Z) = \langle a, e\rangle + \langle a, s\rangle + \langle b, e\rangle.$$

In order to show a further important aspect of the calculus we modifify the example of Fig. 11. Again, we regard two different sorts, $\{a, b\}$ and $\{e, s\}$ designated by u and m, resp. (Fig. 12).



	1	2	3	i	M_0
X	$-u$		u	$\langle u, m\rangle$	a
Y	$\langle u, m\rangle$	$-\langle u, m\rangle$		u	
Z		$\langle u, m\rangle$	$-\langle u, m\rangle$	u	$\langle b, e\rangle$
				$!\,m$	

Fig. 12.

By firing of transition 1 one item of sort u is taken from place X and one item of sort $\langle u, m\rangle$ is put on Y. For example, a is taken from X and either $\langle a, e\rangle$ or $\langle a, s\rangle$ is put on Y. The problem is that we cannot determine which one will be chosen by transition 1. Moreover, after this pair has left Z by firing transition 3 the system has 'forgotten' which choice had been made by transition 1. We are used to express phenomena like this in the following way. With respect of m (in $\langle u, m\rangle$) transition 1 is a *source* and transition 3 is a *sink of information*, and the information itself is observable on Y or Z and not on X. In contrast to m, u designates a sort the elements of which are permanently observable – either as items on X or as first entries in pairs $\langle u, m\rangle$ on Y and Z. This important difference between u and m is in the S-invariant i represented by the fact that u appears in all entries and m does not. m does not 'cover' i like u. The fourth entry of $i - m$ with a *warning flag* (!) – is to remind us that we have to treat m and the sort it designates very carefully when using i.

To demonstrate that, we use M_0 as initial marking and try to complete a follower marking M which is incompletely given by

$$M(X) = M(Y) = 0,$$

$$i^{\mathrm{T}} \bullet M_0 = \langle u, m\rangle \bullet a + u \bullet \langle b, !e\rangle \underset{(2)}{=} i^{\mathrm{T}} \bullet M = u \bullet M(Z).$$

Because sort m does not cover i, the binding of m to symbol e in item $\langle b, e\rangle$ is only 'temporary'. This is indicated by the warning flag in the pair $\langle b, !e\rangle$ which now will be replaced by $\langle b, m\rangle$:

$$u \bullet M(Z) \underset{(!)}{=} \langle u, m\rangle \bullet a + u \bullet \langle b, m\rangle \underset{(S)}{=} \langle a, m\rangle \bullet a + b \bullet \langle b, m\rangle$$

$$\underset{(C)}{=} a \bullet \langle a, m\rangle + b \bullet \langle b, m\rangle \underset{(R)}{=} u \bullet \langle a, m\rangle + u \bullet \langle b, m\rangle$$

$$\underset{(D)}{=} u \bullet (\langle a, m\rangle + \langle b, m\rangle),$$

$$M(Z) \underset{(V)}{=} \langle a, m\rangle + \langle b, m\rangle.$$

The interpretation of this equation is

$$\exists m_1, m_2 \in \{e, s\}: M(Z) = \langle a, m_1 \rangle + \langle b, m_2 \rangle.$$

This statement expresses the maximum of knowledge about M we can conclude from the above assumption. Note the different roles, the variables m, m_1, and m_2 play: m serves as a sort designator, it is a kind of 'reserved' variable used in the calculus; m_1, m_2 are just quantifiable variables.

There are more examples of S-invariants which are covered only by a subset of the sorts involved. It may even happen that all sorts have to be flagged (nevertheless such 'pseudo'-invariants may be useful since they express invariant relations between values). The need for warning flags always arises from some non-determinism in transitions: If there is a transition such that an elementary change of the marking of one adjacent predicate does not determine uniquely the changes in all adjacent predicates, we must use caution.

We now want to show how the calculus of S-invariants can be used to verify invariant assertions about systems. We do this for several different levels of the representation of the same system. Starting with the highest, i.e. most abstract, for m we reduce the 'arity' of the predicates until all predicates are 0-ary, i.e. ordinary places (integer quantities).

We again use the simple resource management example introduced in the previous section, and begin with its most condensed representation shown in Fig. 9. In Fig. 13 the corresponding incidence matrix C, two S-invariants (i and j), and the vector representation of the initial marking M_0 are shown.

	1	2	3	4	i	j	M_0
H	$-u$			u	$\langle u, m \rangle$		$a+b+c$
W	$\langle u, m \rangle$	$-\langle u, m \rangle$			u		
U		$\langle u, m \rangle$	$-\langle u, m \rangle$		u	$F(m)$	
D			$\langle u, m \rangle$	$-\langle u, m \rangle$	u		
R		$-F(m)$	$F(m)$			$\langle u, m \rangle$	$2e$
					$!\, m$	$!\, u$	

Fig. 13.

i is an S-invariant through H, W, U, D, j is an S-invariant through U and R. We may regard the circuits through these places as the graphical representations of i and j. Since m does not cover i, it has a warning flag. The reason why m has no warning flag in j is not because m appears in all non-zero-entries, but because F is bijective and so no information about m can get lost (the possible changes in M uniquely determine the value of m involved).

We now solve some little problems in completing follower markings by applying the formalism w.r.t all the four nets representing the system.

$$i^{\mathrm{T}} \bullet M = \langle u, m \rangle \bullet M(H) + u \bullet M(W) + u \bullet M(U) + u \bullet M(D) \qquad \text{(A1)}$$

$$= \langle u, m \rangle \bullet (a+b+c) = i^{\mathrm{T}} \bullet M_0.$$

(1) *Assumption*:

$$M(H) = M(U) = M(D) = 0.$$

So:

$$u \bullet M(W) = \langle u, m\rangle \bullet (a+b+c)$$

$$\underset{(D)}{=} \langle u, m\rangle \bullet a + \langle u, m\rangle \bullet b + \langle u, m\rangle \bullet c$$

$$\underset{(S,R)}{=} \langle a, m\rangle \bullet u + \langle b, m\rangle \bullet u + \langle c, m\rangle \bullet u$$

$$\underset{D)}{=} u \bullet (\langle a, m\rangle + \langle b, m\rangle + \langle c, m\rangle),$$

$$M(W) \underset{(V)}{=} \langle a, m\rangle + \langle b, m\rangle + \langle c, m\rangle.$$

The interpretation of this equation is

$$\exists m_1, m_2, m_3 \in \{e, s\}: M(W) = \langle a, m_1\rangle + \langle b, m_2\rangle + \langle c, m_3\rangle$$

$$j^{\mathrm{T}} \bullet M = F(m) \bullet M(U) + \langle u, m\rangle \bullet M(R) \qquad \text{(A2)}$$

$$\underset{(2)}{=} \langle u, m\rangle \bullet 2\ell = j^{\mathrm{T}} \bullet M_0.$$

(2) *Assumption*:

$$M(R) = 2\ell.$$

So:

$$F(m) \bullet M(U) + \langle u, m\rangle \bullet 2\ell = \langle u, m\rangle \cdot 2\ell,$$

$$F(m) \bullet M(U) = (\langle u, m\rangle - \langle u, m\rangle) \bullet 2\ell.$$

The two pairs on the right-hand side cannot be different because $F(m) \bullet M(U) \geq 0$.

Consequently: $M(U) = 0$.

The next representation of our little system (Fig. 14, 15) is different w.r.t. representing modes. In the previous model the fact that, for example, user *a* has

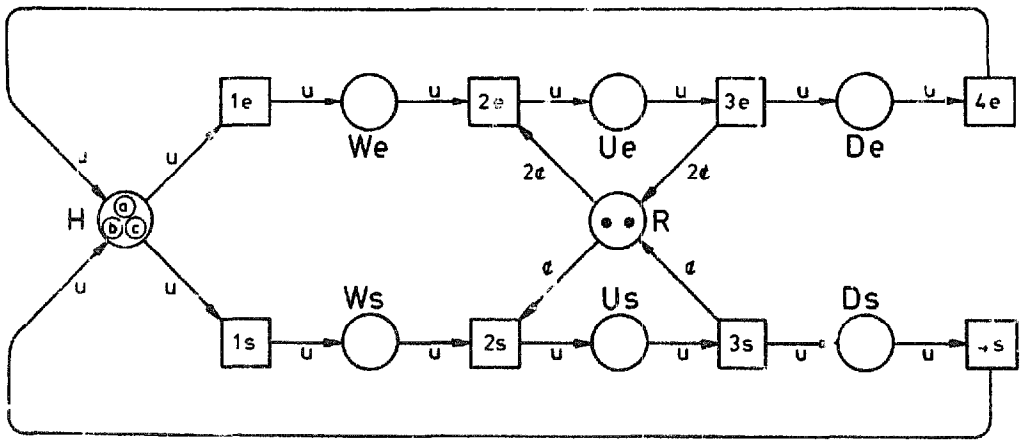


Fig. 14.

chosen mode e is modelled by item $\langle a, e\rangle$ being on W, U, or D. This is now modelled by item a being on We, Ue, or De. So, we have no longer binary predicates. All predicates are unary with exception of R which is 0-ary like in the previous model. The users, still designated by u, are now moving along i without being transformed into entries of pairs. Accordingly, the entries of i are 0 or 1.

	$1e$	$2e$	$3e$	$4e$	$1s$	$2s$	$3s$	$4s$	i	j	M_0
H	$-u$			u	$-u$			u	1		$a+b+c$
We	u	$-u$							1		
Ue		u	$-u$						1	2ℓ	
De			u	$-u$					1		
Ws					u	$-u$			1		
Us						u	$-u$		1	ℓ	
Ds							u	$-u$	1		
R		-2ℓ	2ℓ			$-\ell$	ℓ			u	2ℓ
										$!u$	

Fig. 15.

Comparing both representations we see that the price for reducing the 'arity' of predicates is a considerable increase of the size of the net. Let us now compare the formalisms:

$$i^T \bullet M = M(H)+M(We)+M(Ue)+M(De)+M(Ws)+M(Us)+M(Ds)$$
$$\underset{(2)}{=} a+b+c = i^T \bullet M_0. \qquad \text{(B1)}$$

(1) *Assumption*: $M(H)+M(Ue)+M(Us)+M(De)+M(Ds)=0$.
So: $M(We)+M(Ws)=a+b+c$.

$$j^T \bullet M = 2\ell \bullet M(Ue)+\ell \bullet M(Us)+u \bullet M(R) \underset{(2)}{=} u \bullet 2\ell = j^T \bullet M_0. \qquad \text{(B2)}$$

(2) *Assumption*: $M(R)=2\ell$.
So: $0 \leqslant 2\ell \bullet M(Ue)+\ell \bullet M(Us)=(u-u) \bullet 2\ell$.
Because the left-hand side is non-negative both copies of u must represent the same user. Consequently:

$$M(Ue)=M(Us)=0$$

This example demonstrates what must not surprise us: not only the size of the net, but also the complexity of the invariant assertions increases when the representation is refined. The calculations, however, become simpler. Thus, the costs for solving problems remain roughly the same.

We may regard the model of Fig. 14, 15 as a refinement of the previous model in which the information about modes is not represented by entries in pairs $\langle u, m\rangle$ but by the location of items of sort u (elimination of m).

The 'dual' refinement is shown in Fig. 16, 17 (elimination of u). Without further explanations we shall start to solve the two problems for this model:

$$\forall u' \in \{a, b, c\}: i^{\mathrm{T}} \bullet M = m \bullet M(Hu') + \varepsilon \bullet M(Wu') + \varepsilon \bullet M(Uu') + \varepsilon \bullet M(Du') \underset{(2)}{=} m \bullet \varepsilon = i^{\mathrm{T}} \bullet M_0.$$

(1) *Assumption*: $\forall u' \in \{a, b, c\}: M(Hu') = M(Uu') = M(Du') = 0$.
So: $\forall u' \in \{a, b, c\}: M(Wu') =_{(V)} m$.

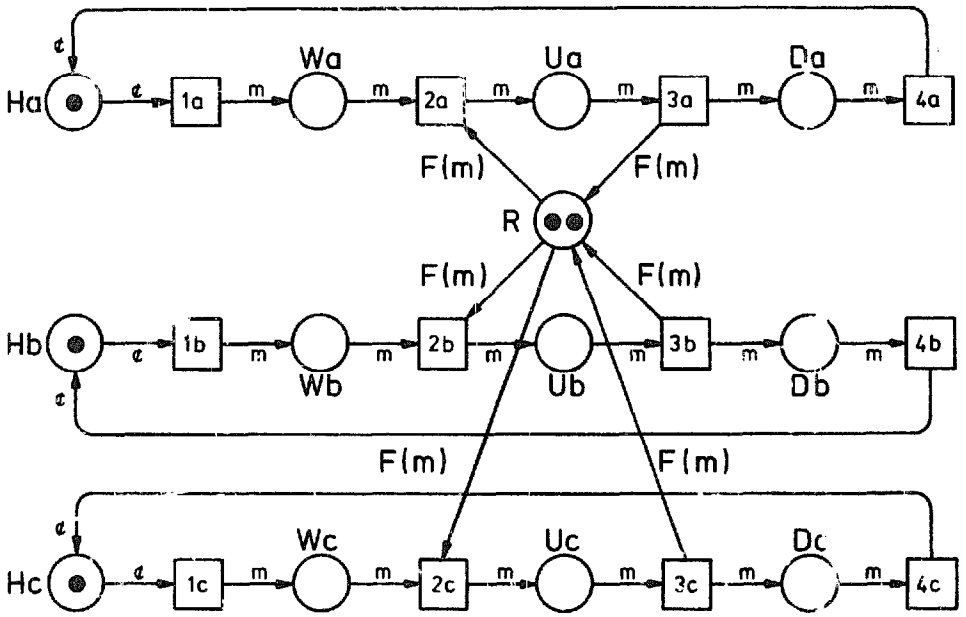


Fig. 16.

	1a	2a	3a	4a	1b	2b	3b	4b	1c	2c	3c	4c	in	ib	ic	j	M_0
Ha	$-\varepsilon$			ε									m				ε
Wa	m	$-m$											ε				
Ua		m	$-m$										ε			$F(m)$	
Da			m	$-m$									ε				
Hb					$-\varepsilon$			ε						m			ε
Wb					m	$-m$								ε			
Ub						m	$-m$							ε		$F(m)$	
Db							m	$-m$						ε			
Hc									$-\varepsilon$			ε			m		ε
Wc									m	$-m$					ε		
Uc										m	$-m$				ε	$F(m)$	
Dc											m	$-m$			ε		
R	$-F(m)$			$F(m)$	$-F(m)$			$F(m)$	$-F(m)$			$F(m)$				m	2ε
													!m	!m	!m		

Fig. 17.

Interpretation: $\exists m_1, m_2, m_3 \in \{e, s\}$: $M(Wa) = m_1 \wedge M(Wb) = m_2 \wedge M(Wc) = m_3$

$$j^T \bullet M = F(m) \bullet M(Ua) + F(m) \bullet M(Ub) + F(m) \bullet M(Uc) + m \bullet M(R)$$
$$\underset{(2)}{=} m \bullet 2e = j^T \bullet M_0. \qquad \text{(C2)}$$

(2) *Assumption*: $M(R) = 2e$.
So:

$$0 \leqslant F(m) \bullet (M(Ua) + M(Ub) + M(Uc)) = (m - m) \bullet 2e,$$
$$M(Ua) = M(Ub) = M(Uc) = 0 \quad \text{because of } |F(m)| \geqslant 1.$$

Comparing this representation with the first one we observe again that the size of the net is larger and the formal costs are nearly the same. The common refinement of the models of Fig. 14, 15 and Fig. 16, 17 is the net of Fig. 7 shown in the previous section. This net is a place/transition-net the size of which is considerably larger than the size of the net in Fig. 9. Even without the incidence matrix of Fig. 7 the solution of the two problems should be easy to follow:

$$\forall u' \in \{a, b, c\}: i^T \bullet M = M(Hu') + M(Wu'e) + M(Wu's) + M(Uu'e)$$
$$+ M(Uu's) + M(Du'e) + M(Du's) \qquad \text{(D1)}$$
$$\underset{(2)}{=} 1 = i^T \bullet M_0.$$

(1) *Assumption*: $\forall u' \in \{a, b, c\} \forall m' \in \{e, s\}$: $M(Hu') = M(Uu'm') = M(Du'm') = 0$.
So: $\forall u' \in \{a, b, c\}$: $M(Wu'e) + M(Wu's) = 1$.

$$j^T \bullet M = 2[M(Uae) + M(Ube) + M(Uce)]$$
$$+ [M(Uas) + M(Ubs) + M(Ucs)] \qquad \text{(D2)}$$
$$\underset{(2)}{=} 2 = j^T \bullet M_0.$$

(2) *Assumption*: $M(R) = 2$.
So: $\forall u' \in \{a, b, c\} \forall m' \in \{e, s\}$: $M(Uu'm') = 0$.

Here we finish our play with the calculus of S-invariants on the several levels of detail of system representation. In the next section, we shall apply the modelling and analytical apparatus presented so far to a more interesting problem: The verification of a given scheme for organizing a distributed data base.

4. The analysis of a distributed data base

Fig. 18 shows the PrT-net model of the organization scheme of a duplicate database system. It is Milne's modification [13] of a model designed by Ellis [2].

In this example, each of n data base managers in responsible for one copy of the database. We assume that they are equally organized w.r.t. managing their copy (but nothing is assumed, for example, about their relative speeds). Furthermore, we assume that any two requests are in conflict with each other, i.e. only one data item or one resource is under consideration. This restriction focusses on the most difficult part of modelling an organization scheme for duplicate data base systems. Treating the general case of several data items would be beyond the scope of this paper.

In the PrT-net of Fig. 18 the dynamic behaviour of all the n database managers is represented. (For sake of comprehensibility, the diagram has been divided into several parts and several places appear more than once; 'sideconditions' are used to keep the net as small as possible.) The net is the result of folding together n isomorphic place/transition nets each representing one database manager. Consequently, in Fig. 18 we have to distinguish between the behaviour of different managers by means of the marking. The initial marking M_0 and its follower markings $M \in [M_0]$ ($M_0 \in [M_0]$ by convention) are defined by means of two finite sets, U and N, where the number of elements of U shall be n and $N = (U \times U) - id$.

U is a set of individual symbols, the identifiers of the database managers. Every $\langle s, r\rangle \in N$ is a request initiated by s (sender) for communication with r (receiver); for any given s, $N_s := N \cap (\{s\} \times U)$ contains all requests belonging to s. The initial marking M_0 is given by $M_0(\textbf{passive}) := \pi(U) = \sum u \,|\, u \in U$, $M_0(\text{HOME}) := \pi(N) = \sum \langle u, v\rangle \,|\, u, v \in U \wedge u \neq v$, all other places are unmarked. The transitions b1, b2, b3 serve as representations of the users. When firing, b1 puts $s \in U$ on place INTREQ. This describes that a user of database manager s wants to change (uniformly) all copies of the database. If this 'internal' request has been executed or rejected, the user receives a corresponding message, namely the same $s \in U$ via DONE or REJECT, by firing b2 or b3, respectively. It is reasonable to attach capacities to the places INTREQ, REJECT, and DONE whereby, for every $s \in U$, the number of copies of s on the respective place is limited. So, for $s \in U$, the capacities model the size of the user queues in database s.

We will explain now very briefly how the model works. First we show that always (under every marking) every manager is in some state and every request is at some location:

Proposition 4.1. *Let* $M \in [M_0]$; *then*

(a) $M(\textbf{pass.}) + M(\textbf{act.}) + M(\textbf{soak.}) + M(\textbf{updat.}) = M_0(\textbf{pass.}) = \pi(U)$.

(b) $M(\text{HOME}) + M(\text{EXTREQ}) + M(\text{ACK}+) + M(\text{ACK}-) + M(\text{ACKb}) + M(\text{UPD}) + M(\text{ACKd}) = M_0(\text{HOME}) = \pi(N)$

Proof. There exist two S-invariants I1 and I2 with

$$\text{I1}(\textbf{pass.}) = \text{I1}(\textbf{act.}) = \text{I1}(\textbf{soak.}) = \text{I1}(\textbf{updat.}) = 1,$$
$$\text{I1}(p) = 0 \quad \text{for all other places } p,$$
$$\text{I2}(\text{HOME}) = \text{I2}(\text{EXTREQ}) = \text{I2}(\text{ACK}+) = \text{I2}(\text{ACK}-) = \text{I2}(\text{ACKb}) = \text{I2}(\text{ACKd}) = \text{I2}(\text{UPD}) = 1,$$
$$\text{I2}(q) = 0 \quad \text{for all other places } q.$$

(a) and (b) are evaluations of (2) for I1 and I2.

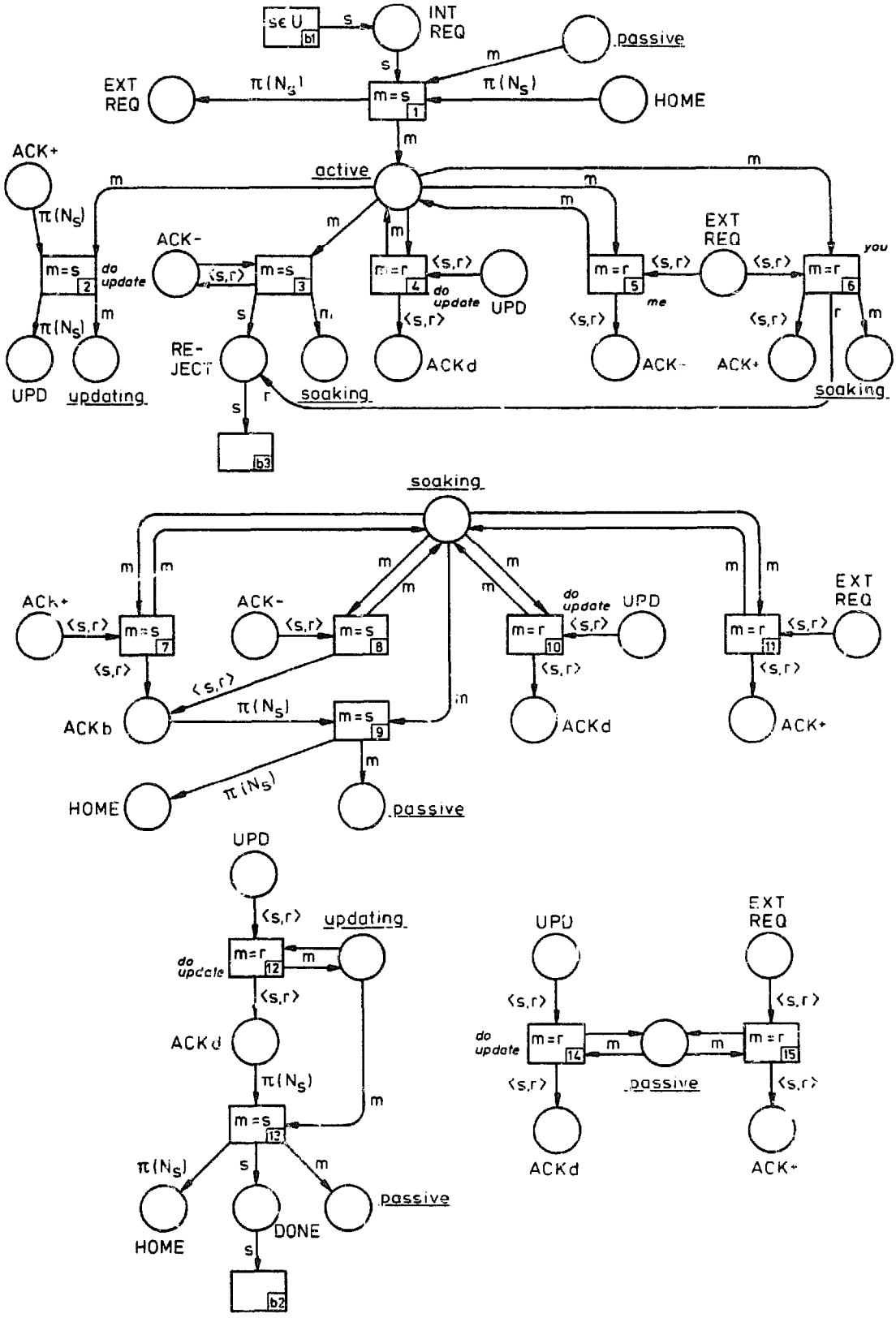

s∈U
b1
s
INT REQ
passive
m
s
EXT REQ
π(N_S)
m=s
1
π(N_S)
HOME
m
ACK+
active
m
π(N_S)
m
m
m
m
m
EXT REQ
ACK-
m=s
2
do update
⟨s,r⟩
m=s
3
m=r
4
⟨s,r⟩
do update
UPD
m=r
5
⟨s,r⟩
EXT REQ
⟨s,r⟩
m=r
6
you
me
π(N_S)
m
s
n
⟨s,r⟩
⟨s,r⟩
⟨s,r⟩
r
m
UPD
updating
RE-JECT
ACKd
ACK-
ACK+
soaking
soaking
s
r
b3
soaking
m
m
m
m
m
m
m
m
m
m
EXT REQ
ACK+
⟨s,r⟩
m=s
7
ACK-
⟨s,r⟩
m=s
8
do update
UPD
m=r
10
⟨s,r⟩
m=r
11
⟨s,r⟩
⟨s,r⟩
⟨s,r⟩
⟨s,r⟩
ACKb
π(N_S)
m=s
9
in
ACKd
ACK+
HOME
π(N_S)
m
passive
UPD
⟨s,r⟩
updating
do update
m=r
12
m
⟨s,r⟩
ACKd
π(N_S)
m=s
13
m
π(N_S)
s
m
passive
HOME
DONE
s
b2
UPD
EXT REQ
⟨s,r⟩
⟨s,r⟩
do update
m=r
14
m
m
m=r
15
⟨s,r⟩
passive
⟨s,r⟩
ACKd
ACK+

Fig. 18

To trace an internal request for a manager k we start with the firing of transition 1. By doing so k goes from state *passive* to *active* and its requests $\langle k, i\rangle$, $(i \in U, i \neq k)$, are put on EXTREQ, which means that they are sent to all other managers i, $i \neq k$, as external requests. Then two possibilities are conceivable:

(1) k gets a positive acknowledgement from all the other managers. Then the corresponding marking M' enables transition 2: $k \leq M'(\textbf{active}) \wedge \pi(N_k) \leq M'(\text{ACK}+)$. By firing of transition 2 k goes from *active* to *updating* and for every $i \in (U - \{k\})$ the request $\langle k, i\rangle$ is again sent to i, but now as an update request; furthermore we assume that k performs the update in database k. In database $i \neq k$ the corresponding update is performed by firing of transition 14, 4, 10 or 12, depending on manager i's current state. After all managers have performed this update as requested by k, the requests $\langle k, i\rangle$, $(i \in U, i \neq k)$, are collected on place ACKd. So transition 13 is enabled and by its firing k changes back to *passive* and the requests are put back to HOME. Moreover, one copy of k is put on DONE as an acknowledgement for the user that 'its' update is performed in all copies of the database.

(2) In case one manager, say $j \neq k$, is unable or unwilling to perform k's request as soon as possible, he sends a negative acknowledgement back to k; i.e. it fires transition 5 for $m = r = j$ putting $\langle s, r\rangle = \langle k, j\rangle$ from EXTREQ to ACK−. Now for k on *active* transition 3 is enabled. By firing it k goes from *active* to *soaking* and its user gets a negative acknowledgement in form of a copy of k on REJECT. In state *soaking* k collects all requests on ACKb by firing transition 8 and/or transition 7 (repeatedly). Then, by firing of transition 9, it goes back to *passive* and the requests $\langle k, i\rangle$, $(i \in U, i \neq k)$, are put back to their homeposition HOME.

The rest of the model shall be described from the receivers point of view. In case a manager j is in state *passive* or *soaking* and receives an external request $\langle k, j\rangle$ on EXTREQ, it grants by firing transition 15 or 11. In case j is in state *active* there is a conflict between j and k. Firing transition 5 means not granting k's request by putting $\langle k, j\rangle$ on ACK− as a negative acknowledgement; firing transition 6 means for j abandoning its request in favour of k by changing to *soaking*, putting $\langle k, j\rangle$ on ACK+, and a copy of j on REJECT to inform the user. If j is in state *updating*, it does not take notice of external request $\langle k, j\rangle$ on EXTREQ until being back in state *passive*. In any state, however, j has to notice an update request $\langle k, j\rangle$ from k on UPD, to perform the update requested by k, and to put $\langle k, j\rangle$ on ACKd as an update acknowledgement for k.

We are now prepared to formulate some results about the model. To start with, we state a result about a synchronization of a manager k and the requests $\langle k, i\rangle$, $(i \in U, i \neq k)$, initiated by k:

Proposition 4.2. *Let* $M \in [M_0]$; *then*

$$k \leq M(\textbf{passive}) \Leftrightarrow \pi(N_k) \leq M(\text{HOME}),$$

$$k \leq M(\textbf{active}) + M(\textbf{soaking}) \Leftrightarrow \pi(N_k)$$
$$\leq M(\text{EXTREQ}) + M(\text{ACK}+) + M(\text{ACK}-) + M(\text{ACKb}),$$

$$k \leq M(\textbf{updating}) \Leftrightarrow \pi(N_k) \leq M(\text{UPD}) + M(\text{ACKd}.$$

Before we prove this, we interpret it by dividing the places of I1 ∪ I2 into three 'request regions': no request region (NR), external request region (ER), update request region (UP):

$$NR \cap I1 := \{\textbf{passive}\}, \quad NR \cap I2 := \{HOME\},$$
$$ER \cap I1 := \{\textbf{active}, \textbf{soaking}\},$$
$$ER \cap I2 := \{EXTREQ, ACK+, ACK-, ACKb\},$$
$$UR \cap I1 := \{\textbf{updating}\}, \quad UR \cap I2 := \{UPD, ACKd\}.$$

(Here we have identified the S-invariants I1, I2 with the sets of places they pass through.)

Proposition 4.2 then states that a manager k is in one of these request regions if, and only if, all its requests $\langle k, i\rangle$, $(i \in U, i \neq k)$, are in the same region.

Proof. The stated property holds for M_0 (trivial). It is preserved by transitions 1, 2, 9, and 13, which are the only changes from one region into another.

For applying the organizational scheme it is important to know whether it is *deadlockfree* and *consistent.*

Theorem 4.3 (liveness). *Under any marking $M \in [M_0]$ there exists an enabled transition.*

Proof. First let us mention that this statement is non-trivial in the case of finite capacities for INTREQ (for every $k \in U$). Because of Proposition 4.1 every manager is always in one of four states. Let k be a given manager, and $M \in [M_0]$:

(1) $k \leq M(\textbf{passive}) \Rightarrow \pi(N_k) \leq M(HOME)$ because of Proposition 4.2.

Notice now that there is for every $s \in U$ a positive capacity for INTREQ:

$k \leq M(INTREQ) \Rightarrow$ transition 1 is enabled,

$k \not\leq M(INTREQ) \Rightarrow$ transition b1 is enabled.

(2) $k \leq M(\textbf{active}) \Rightarrow \pi(N_k) \leq M(EXTREQ) + M(ACK+) + M(ACK-) + M(ACKb)$ because of Proposition 4.2.

(2.1) $\neg\exists j: \langle k, j\rangle \leq M(ACKb)$ because putting $\langle k, j\rangle$ on ACKb is only possible for $k \leq M(\textbf{soaking})$.

(2.2) $\pi(N_k) \leq M(ACK+) + M(ACK-)$;

$\pi(N_k) \leq M(ACK+) \Rightarrow$ transition 2 is enabled,

$\exists j: \langle k, j\rangle \leq M(ACK-) \Rightarrow$ transition 3 is enabled.

(2.3) $\exists j: \langle k, j\rangle \leq M(EXTREQ)$;

$j \leq M(\textbf{pass.}) + M(\textbf{act.}) + M(\textbf{soak.}) \Rightarrow$ one of transitions 15, 5, 6, 11 is enabled;

$j \leq M(\textbf{updat.})$ see (4) below.

(3) $k \leq M(\textbf{soaking}) \Rightarrow \pi(N_k) \leq M(ACK+) + M(ACK-) + M(ACKb) + M(EXTREQ)$

(3.1) $\pi(N_k) \leq M(\text{ACKb}) \Rightarrow$ transition 9 is enabled,

(3.2) $\exists j: \langle k, j \rangle \leq M(\text{ACK+}) + M(\text{ACK-}) \Rightarrow$ transition 7 or 8 is enabled.

(3.3) $\exists j: \langle k, j \rangle \leq M(\text{EXTREQ})$ see (2.3) above.

(4) $k \leq M(\textbf{updating}) \Rightarrow \pi(N_k) \leq M(\text{UPD}) + M(\text{ACKd})$.

(4.1) $\pi(N_k) \leq M(\text{ACK1}) \Rightarrow$ transition 13 is enabled,

(4.2) $\exists j: \langle k, j \rangle \leq M(\text{UPD}) \Rightarrow$ one of the transitions 14, 4, 10, 12 is enabled for j (because of Proposition 4.1 applied to j).

Consistency means for the model under consideration that after every complete update the n copies of the database are identical. Under the assumption that the model is consistent for the initial marking M_0, the next theorem guarantees consistency:

Theorem 4.4 (consistency). *For any* $M \in [M_0]$ *and* $K \in U$:

$$k \leq M(\textbf{updating}) \Rightarrow i \not\leq M(\textbf{updating}), \quad (i \in U, i \neq k)$$

Proof. Let $k \leq M(\textbf{updating})$ and $\langle i, k \rangle \leq M(\text{EXTREQ})$; then $\langle i, k \rangle$ cannot leave EXTREQ for ACK+ because transition 6 is not enabled. So it is impossible to bring both k and i to place **updating**.

As a consequence of Theorem 4.4 transition 12 turns out to be useless for the model in its present form. Transition 12 would, however, be necessary if the model would be refined by adding further resources, thus granting concurrent updating.

We will finish our analysis of the scheme with some critical remarks using a catalogue of properties of a 'good' solution given by Ellis [2]. The model is *homogeneous* (all managers have essentially identical control programs), *speed independent*, *deadlockfree*, *consistent*, *functional* (in applications there are no restrictions concerning data and functions).

The model is, however, not free from *critical blocking*. Even for two managers this can be shown easily. Let $U := \{a, b\}$, $N = \{\langle a, b \rangle, \langle b, a \rangle\}$. In case both have sent an external request to each other the current marking is M where $M(\textbf{active}) = a + b$, $M(\text{EXTREQ}) = \langle a, b \rangle + \langle b, a \rangle$. So we observe a double activation for transitions 5 and 6. If transition 5 fires twice, for the follower marking M' $M'(\textbf{active}) = a + b$, $M'(\text{ACK-}) = \langle a, b \rangle + \langle b, a \rangle$ holds. No updating can be performed before at least one manager has been back in *passive*. If under M transition 6 fires twice for the follower marking M'' $M''(\textbf{soaking}) = a + b$ holds. Again, no updating can be performed before a or b has been back to passive. Because this double firing of transition 5 or 6 can be repeated without any intermediate updating, the possibility of critical blocking has to be taken into account. But this drawback can be eliminated by adding mechanisms guaranteeing fair schedules. According to Theorem 4.3 deadlock freeness is guaranteed for any resolution of conflicts between competing requests.

As a major drawback the lack of *partial operability* (cf. [2]) has to be viewed. Let, again for two managers, $M(\textbf{active}) = a + b$, $M(\text{EXTREQ}) = \langle a, b \rangle + \langle b, a \rangle$. Now we

assume b abandoning its request in favour of a by firing transition 6. Then the current marking is M''' where $M'''(\textbf{active}) = a$, $M'''(\textbf{soaking}) = b$, $M'''(\text{ACK}+) = \langle a, b\rangle$, M''' (EXTREQ) $= \langle b, a\rangle$. If now a does not send $\langle b, a\rangle$ back to b, by firing transition 5 putting $\langle b, a\rangle$ on ACK−, b starves. In case of a crash of manager a the system dies – a violation of partial operability.

5. Conclusion

We have presented a new technique for modelling organizational systems which adds to the descriptive and analytical power of Petri nets a new dimension: the formal treatment of individuals and their changing properties and relations. The technique provides a whole spectrum of degrees of abstraction such that the user is no longer forced to deal with larger systems at an unacceptable level of detail as it could happen with ordinary Petri nets. Aspects of the modelled systems which are of no immediate concern can be transferred into parameters as, e.g., the actual number of identical components ruled by an organizational scheme.

The step from ordinary Petri nets to our predicate/transition-nets was strongly influenced by the development leading from propositional to first-order predicate logic (which is an integral part of our language about systems due to the notion of facts). Accordingly, the linear-algebraic techniques for analyzing Petri nets were raised to the new level by means of integer functions generalizing the integers.

Within the development of the General Net Theory of processes and systems, the purpose of our work is to connect the conceptual and mathematical foundations of the theory closer to the levels of practical systems organization. We hope that we have achieved some progress in this direction.

References

[1] W. Brauer, Editor, *Net Theory and Application, Proc. Advanced Course of General Net Theory of Processes and Systems*, Hamburg, 1979, Lecture Notes in Computer Science **84** (Springer, Berlin, 1980).

[2] C.A. Ellis, Consistency and correctness of duplicate database systems, *Proc. 6th Symposium on Operating System Principles*, Purdue University, November 1977, *ACM Operating Systems Rev.* **11** (5) (1977).

[3] H.J. Genrich and K. Lautenbach, Facts in place/transition-nets, in: J. Winkowski, Ed., *Mathematical Foundations of Computer Science 1978*, Lecture Notes in Computer Science **64** (Springer, Berlin, 1978).

[4] H.J. Genrich and K. Lautenbach, The analysis of distributed systems by means of predicate/transition-nets, in: G. Kahn, Ed., *Semantics of Concurrent Computation*, Lecture Notes in Computer Science **70** (Springer, Berlin, 1979) 123–146.

[5] H.J. Genrich, K. Lautenbach and P.S. Thiagarajan, Elements of general net theory, in: W. Brauer et al., Eds., *Net Theory and Application* (Springer, Berlin, 1980).

[6] H.J. Genrich and G. Thieler-Mevissen, The calculus of facts, in A. Mazurkiewicz, Ed., *Mathematical Foundations of Computer Science 1976*, Lecture Notes in Computer Science **45** (Springer, Berlin, 1976).