

HIGH-LEVEL PETRI NETS

Kurt Jensen

DAIMI PB-151
September 1982

Computer Science Department
AARHUS UNIVERSITY
Ny Munkegade - DK 8000 Aarhus C - DENMARK
Telephone: 06 - 12 83 55



HIGH-LEVEL PETRI NETS

Kurt Jensen
Computer Science Department
Aarhus University, Ny Munkegade
DK-8000 Aarhus, Denmark

Abstract This paper combines two closely related net models, predicate/transition nets and coloured Petri nets, into a new net model called high-level Petri nets. The new model is intended to combine the qualities of the two old models into a single formalism, and we propose in future to use high-level Petri nets instead of both predicate/transition nets and coloured Petri nets.

1. INTRODUCTION

The practical use of Petri nets to describe concurrent systems has shown a demand for more powerful net types, to describe complex systems in a manageable way. In place/transition nets (PT-nets) it is often necessary to have several identical subnets, because a folding into a single subnet would destroy the possibility to distinguish between different processes.

The development of predicate/transition nets (PrT-nets) was in this respect a significant improvement [1]. In PrT-nets information can be attached to each token as a token-colour and each transition can fire in several ways represented by different firing-colours. The relation between a firing-colour and the involved token-colours is defined by expressions attached to the arcs. Restrictions on the possible firing-colours are defined by predicates attached to the transitions. It is now possible to distinguish between different processes, even though their subnets have been folded into a single subnet. It should be emphasized that the "colour" attached to a token or a firing can be a complex information unit, such as the entire state of a process or the contents of a buffer area. New colours can be created by transition firings and there may be an infinite number of them.

Although PrT-nets turned out to be very useful in the description of systems they have a serious drawback concerning formal analysis. One of the most important analysis methods in place/transition nets is the

This paper was presented at the 3rd European Workshop on Applications and Theory of Petri Nets, Varenna, Italy, September 1982. The proceedings of the workshop will be published as a volume in Informatik Fachberichte, Springer Verlag.

construction of linear place-invariants by means of homogeneous matrix equations [4, 5]. This method is generalized to PrT-nets in [1], but the place-invariants there contain free variables (over sets of colours). To interpret the place-invariants it seems necessary to bind the free variables via a substitution, where at least partial knowledge about the firing sequence leading to the marking in question must be used. Until now no satisfactory solution to the problem is published, although some substitution rules are sketched in [1] without a proof of their soundness.

To overcome this problem coloured Petri nets (CP-nets) were defined in [2]. The main ideas of CP-nets are directly inspired by PrT-nets, but the relation between a firing-colour and the involved token-colours is now defined by functions attached to the arcs and not by expressions. This removes the free variables, and place-invariants can now be interpreted without problems as demonstrated in [2]. Moreover CP-nets explicitly attach a set of possible token-colours to each place and a set of possible firing-colours to each transition. Compared to PrT-nets, where the colour-sets are only implicitly defined, this often gives a more comprehensible net. On the other hand the functions attached to arcs in CP-nets sometimes seem to be more difficult to understand than the corresponding expressions of PrT-nets.

As indicated above there is a strong relation between PrT-nets and CP-nets, and from the very beginning it has been clear that most descriptions in one of the net models can be informally translated to the other model and vice versa. This paper shows how to combine the qualities of PrT-nets and CP-nets into a single net model called high-level Petri nets (HL-nets). We propose in future to use HL-nets instead of both PrT-nets and CP-nets.

An HL-net can be represented in two different forms: by drawing a directed bipartite graph with inscriptions attached to nodes and arcs, or by defining a 6-tuple containing formal mathematical elements such as sets and functions. The first form uses mainly the notation known from PrT-nets, i.e. expressions and predicates containing free variables. It is appropriate for the description and informal explanation of a system. The second form uses mainly the notation known from CP-nets, i.e. functions and colour-sets. It is appropriate for the formal analysis of a system. e.g. by place-invariants. The two forms are equivalent in the sense that a formal translation between them exists.

HL-nets differ from PrT-nets in the following ways:

- The set of possible token-colours at a place is explicitly defined.
- The number of tokens added or removed at a given place may be different for two firing-colours of the same transition.
- The set of allowable expressions and predicates is not explicitly defined (but if desired this can be done by means of a many-sorted algebra, from which the allowable expressions, predicates, functions and sets can be built up).

HL-nets differ from CP-nets in the following way:

- The incidence-function is split into a negative and a positive part (which allows us to handle side-conditions).

Place-invariants can be interpreted without problems in HL-nets, but they may be difficult to find. The problem is that the elements of the incidence-matrix are no longer integers (contained in a field), but functions (which only constitute a non-commutative ring) and thus division of two elements may be impossible. For this situation no general algorithm is known to solve homogeneous matrix equations. To overcome this, four transformation rules were defined in [3]. The transformation rules are inspired by the method of Gauss-elimination used for matrices, where all elements belong to a field. They can be used to transform the incidence-matrix of a CP-net. It is proved that the transformation rules are sound, in the sense that they do not change the set of place-invariants. By the remarks above we cannot expect them to be complete, in the sense that all place-invariants can be found by them. But, as demonstrated in [3], it is often possible to transform the original incidence-matrix to such a degree that a number of place-invariants immediately can be found by inspection of the simplified matrix. An interactive Pascal program has been created, which allows a user sitting at a terminal to evoke the different transformation rules on a given incidence-matrix. This edp-system greatly enhances the practical value of the transformation rules.

In this paper we define the simple HL-nets to be those nets, where the number of tokens added or removed at a given place is the same for all firing-colours of a given transition. Each simple HL-net has an underlying PT-net, obtained by ignoring all colour-information. Analogously

we define a class of simple place-invariants for HL-nets, and it can be shown that each simple place-invariant of a simple HL-net, by ignoring all colour-information, transforms into a place-invariant of the underlying PT-net. Since all place-invariants of PT-nets can be found, e.g. by Gauss-elimination, this shows us where to look for simple place-invariants of simple HL-nets.

2. HIGH-LEVEL PETRI NETS IN TUPLE-FORM AND GRAPH-FORM

In this section we define the tuple-form and graph-form of HL-nets and we show how to translate between the two forms. But first we need to introduce bags.

A bag, over a non-empty set S , is a set, which may contain multiple occurrences of elements from S . In this paper we shall only deal with finite bags, and each bag b over S is represented as a formal sum

$$\sum_{s \in S} b(s)s$$

where the non-negative integer $b(s)$ denotes the number of occurrences of the element s in the bag b . The formal sum is convergent since b is finite, i.e.

$$\sum_{s \in S} b(s) < \infty$$

The set of all finite bags over the non-empty set S will be denoted by $BAG(S)$.

As an example $\{a, b, d, b\}$ is a finite bag over the set $\{a, b, c, d\}$, and it is represented by the formal sum $a+2b+d$.

Summation, scalar-multiplication, comparison, and multiplicity of bags are defined in the following way, where $b, b_1, b_2 \in BAG(S)$ and $n \in \mathbb{N}$

$$b_1 + b_2 = \sum_{s \in S} (b_1(s) + b_2(s))s$$

$$n \times b = \sum_{s \in S} (n b(s))s$$

$$b_1 \leq b_2 \Leftrightarrow \forall s \in S: b_1(s) \leq b_2(s)$$

$$|b| = \sum_{s \in S} b(s)$$

When $b_1 \leq b_2$ we also define subtraction:

$$b_2 - b_1 = \sum_{s \in S} (b_2(s) - b_1(s))s$$

A function $F \in [S \rightarrow BAG(R)]$, where S and R are non-empty sets, can be extended uniquely to a linear function $\hat{F} \in [BAG(S) \rightarrow BAG(R)]$ called the bag-extension of F :

$$\forall b \in BAG(S): \hat{F}(b) = \sum_{s \in S} b(s) \times F(s)$$

Functions F and \hat{F} defined as above are said to be simple, with multiplicity $n \in \mathbb{N}$, iff

$$\forall s \in S: |F(s)| = n$$

which is equivalent to

$$\forall b \in BAG(S): |\hat{F}(b)| = n|b|$$

When F and \hat{F} are simple, their multiplicity is denoted by $|F|$ and $|\hat{F}|$. It can be shown that F and \hat{F} are simple iff there exists a bag of functions $b \in BAG([S \rightarrow R])$ such that

$$F = \sum_{f \in [S \rightarrow R]} b(f) f$$

where the sum and product denote the normal sum and scalar-multiplication of functions. If $|F| = |\hat{F}| = 0$ the functions are said to be trivial.

We shall use $[...]_S$, $[...]_L$ and $[...]_{SL}$ to denote the sets of all simple/linear functions.

Definition An HL-tuple is a 6-tuple $H = (P, T, C, I_-, I_+, M_0)$ where

- (1) P is a set of places
- (2) T is a set of transitions
- (3) $P \cap T = \emptyset$, $P \cup T \neq \emptyset$
- (4) C is the colour-function defined from $P \cup T$ into non-empty sets
- (5) I_- and I_+ are the negative and positive incidence-function defined on $P \times T$, such that $I_-(p, t), I_+(p, t) \in [BAG(C(t)) \rightarrow BAG(C(p))]_L$ for all $(p, t) \in P \times T$
- (6) M_0 the initial marking is a function defined on P , such that $M_0(p) \in BAG(C(p))$ for all $p \in P$.

Next we define the graph-form of HL-nets, but first we need to define expressions and predicates.

Let $D_0, D_1, D_2, \dots, D_n$ be a sequence of non-empty sets, where $n \geq 1$. When the typed lambda-expression

$$\lambda v_1:D_1, v_2:D_2, \dots, v_n:D_n. \text{ EXP}:D_0$$

defines a function from $D_1 \times D_2 \times \dots \times D_n$ into D_0 we say that EXP is an expression with type D_0 and with the set of free variables $V = \{v_1:D_1, v_2:D_2, \dots, v_n:D_n\}$. A predicate is an expression with type Boolean.

Definition An HL-graph is a graph with two disjoint sets of nodes called places and transitions. Any pair of a place and a transition may be connected with a set of directed arcs (which may go in both directions).

- (1) Each place p has attached to it a non-empty set of token-colours $C(p)$ and an initial marking $M_0(p) \in \text{BAG}(C(p))$.
- (2) Each arc (with place p as source/destination) has attached to it an arc-expression with type $\text{BAG}(C(p))$ and with an arbitrary set of variables.
- (3) Each transition has attached to it a predicate called the guard. The guard can only have those variables which are already in the immediately surrounding arc-expressions. To avoid degenerate transitions with no firing-colours the guard must differ from the constant predicate FALSE.

Places are drawn as ellipses, transitions as rectangles, and arcs as arrows.

Next we show how to translate HL-graphs into HL-tuples:

- (1) P, T, M_0 , and C 's restriction to P , are immediately defined by the places, transitions, initial-markings and token-colours of the HL-graph.

- (2) For each $t \in T$ we define:

$$C(t) = \{(d_1, d_2, \dots, d_n) \in D_1 \times D_2 \times \dots \times D_n \mid (\lambda (v_1, v_2, \dots, v_n). \text{PRED})(d_1, d_2, \dots, d_n)\}$$

where PRED is the predicate attached to t and $V(t) = \{v_1:D_1, v_2:D_2, \dots, v_n:D_n\}$ is the set of all variables in the immediately surrounding arc-expressions.

- (3) Let $(p, t) \in P \times T$ be given and let $C(t)$ and $V(t)$ be defined as above. If a single arc from p to t exists, with expression EXP, we define $I_-(p, t)$ to be the bag-extension of the

following function with domain $C(t)$:

$$\lambda (v_1, v_2, \dots, v_n): C(t) . \text{ EXP}$$

If several arcs from p to t exist, $I_-(p, t)$ is the sum of the corresponding bag-extensions, constructed above. If no arc from p to t exists, $I_-(p, t)$ is the trivial function.

- (4) I_+ is defined in the same way as I_- , but by means of the arcs from transitions to places.

Next we show how to translate HL-tuples into HL-graphs. We only use a single variable $x: \bigcup_{t \in T} C(t)$ typed with the union of all firing-colours:

- (1) Places, transitions, initial-markings and token-colours are immediately defined by P, T, M_0 and by C 's restriction to P .
- (2) Each transition t gets the predicate $x \in C(t)$ as guard.
- (3) There is an arc from place p to transition t iff $I_-(p, t)$ is non-trivial. The expression attached to the arc is $I_-(p, t)(x)$.
- (4) There is an arc from transition t to place p iff $I_+(p, t)$ is non-trivial. The expression attached to the arc is $I_+(p, t)(x)$.

Let GRAPH and TUPLE be the set of all HL-graphs and HL-tuples respectively, and let

$$T_1: \text{GRAPH} \rightarrow \text{TUPLE}$$

$$T_2: \text{TUPLE} \rightarrow \text{GRAPH}$$

be the two translations defined above.

Theorem 1

- T_1 is surjective, but not injective.
 T_2 is injective, but not surjective.

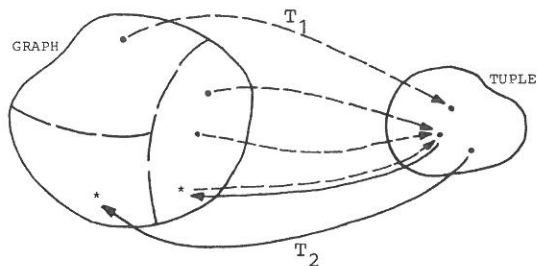
Proof From the definition of T_1 and T_2 it can be checked, that $T_1 \circ T_2$ is the identity function on TUPLE. This implies surjectivity of T_1 and injectivity of T_2 . Proof of the two negative properties is trivial. \square

T_1 induces, by its preimages, an equivalence relation on GRAPH:

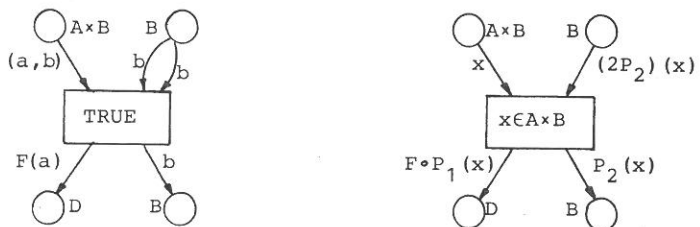
$$g \approx g' \Leftrightarrow T_1(g) = T_1(g')$$

For each equivalence class there exists exactly one element, which is contained in $T_2(\text{TUPLE})$ and this element is said to be in normal form. The

translation $T_2 \circ T_1$ maps each HL-graph into an equivalent HL-graph in normal-form. The normal-form is, except for minor differences, identical to the graph-form used for CP-nets in [2]. In the following figure normal-forms are shown by *.



As an example the following HL-graphs are equivalent. The right-most one is in normal-form:



3. PLACE-INVARIANTS FOR HL-NETS

In this section we define the dynamic properties of HL-nets. We do this in terms of the tuple-form, but all definitions can easily be translated to cover the graph-form.

Let an HL-net $H = (P, T, C, I_-, I_+, M_0)$ be given. For convenience we shall assume P and T to be finite.

A marking of H is a function M defined on P , such that $M(p) \in \text{BAG}(C(p))$ for all $p \in P$. A step of H is a function X defined on T , such that $X(t) \in \text{BAG}(C(t))$ for all $t \in T$. The step X has concession in the marking M iff

$$\forall p \in P: \sum_{t \in T} I_-(p, t) (X(t)) \leq M(p)$$

which can also be written

$$I_- * X \leq M$$

where I_- , X and M are viewed as matrices (of size $|P| \times |T|$, $|T| \times 1$, and $|P| \times 1$, respectively), $*$ denotes generalised matrix-multiplication (to be defined below) and \leq denotes element-wise comparison of matrix-elements (which are bags).

Let $A = (a_{ij})_{1 \leq i \leq r, 1 \leq j \leq s}$ be a matrix with elements which are linear functions mapping bags into bags and let $B = (b_{jk})_{1 \leq j \leq s, 1 \leq k \leq t}$ be a matrix with elements which are bags or linear functions mapping bags into bags. Then we define the generalised matrix-multiplication such that $A * B = (c_{ik})_{1 \leq i \leq r, 1 \leq k \leq t}$, where

$$\forall i \in 1..r \forall k \in 1..t: c_{ik} = \sum_{j=1}^s a_{ij} b_{jk}$$

The juxtaposition $a_{ij} b_{jk}$ means function composition (when b_{jk} is a function) or function application (when b_{jk} is a bag). We shall only use the generalised matrix-multiplication in situations where the matrix-elements fit together, in the sense that the function compositions/applications and sums are possible. The generalised matrix-multiplication was already introduced in [2] and it is a standard construction in the theory of non-commutative rings.

When X has concession it may fire and thus transform M into a directly-reachable marking M' , such that

$$M' = (M - I_- * X) + I_+ * X$$

Reachability is the reflexive, symmetric and transitive closure of direct-reachability. M is reachable iff it is reachable from M_0 .

A weight-function of the HL-net H , with respect to a non-empty set U , is a function W defined on P , such that $W(p) \in [\text{BAG}(C(p)) \rightarrow \text{BAG}(U)]_{\perp}$ for all $p \in P$.

Theorem 2 If a weight-function W satisfies $W * I_- = W * I_+$, we have $W * M = W * M'$ for all markings M and M' reachable from each other.

Proof If M' is directly-reachable from M by firing of step X we get, due to distributivity and associativity of the generalised matrix-multiplication:

$$\begin{aligned}
W * M' &= W * (M - I_- * X + I_+ * X) \\
&= W * M - W * I_- * X + W * I_+ * X \\
&= W * M - (W * I_- - W * I_+) * X \\
&= W * M
\end{aligned}$$

Thus the desired property is satisfied when M' is directly-reachable from M and the proof is finished by induction over the number of steps between M and M' . \square

Corollary If $W * I_- = W * I_+$, the equation $W * M = W * M_0$ is satisfied for all reachable markings M , and it is called the linear place-invariant induced by W .

4. SIMPLE HL-NETS AND SIMPLE PLACE-INVARIANTS

When A is a matrix, of bags or simple linear functions, we shall use $|A|$ to denote the matrix obtained from A by replacing each matrix-element by its multiplicity.

Lemma

$$|A_1 * A_2| = |A_1| * |A_2| \text{ whenever } A_1 * A_2 \text{ is defined.}$$

Proof The composition of two simple linear functions $F_1 \circ F_2$ is a simple linear function with multiplicity $|F_1 \circ F_2| = |F_1| |F_2|$ and the application $F(b)$ of a simple linear function to a bag is a new bag with multiplicity $|F(b)| = |F| |b|$. \square

An HL-net $H^* = (P, T, C, I_-, I_+, M_0)$ is simple iff $I_-(p, t)$ and $I_+(p, t)$ are simple for all $(p, t) \in P \times T$, and we then define its underlying PT-net by $P^* = (P, T, |I_-|, |I_+|, |M_0|)$. A weight-function W of H^* is simple iff $W(p)$ is simple for all $p \in P$, and we then define its underlying weight-function by $|W|$, which is a weight-function of P^* .

Theorem 3 If a simple weight-function W induces a linear place-invariant of a simple HL-net H^* , then its underlying weight-function $|W|$ induces a linear place-invariant of the underlying PT-net P^* .

Proof

$$|W| * |I_-| = |W * I_-| = |W * I_+| = |W| * |I_+|. \quad \square$$

The above theorem shows us where to look for simple place-invariants in a simple HL-net H^* . First we calculate a maximal set of linear independent weight-functions inducing place-invariants of the underlying PT-net P^* , e.g. by means of Gauss-elimination. Then, for each of these weight-functions W , we try to construct an overlying simple weight-function W^* inducing a place-invariant of H^* ; and to do this, we know that each element of W^* must be a simple function with a multiplicity which equals the corresponding element of W . Since functions do not constitute a field, but only a non-commutative ring, there may in some cases be several "independent" weight-functions of H^* overlying the same weight-function of P^* .

Given a weight-function W of P^* , there is always at least one overlying weight-function \bar{W} of H^* :

$$\forall p \in P: \bar{W}(p) = \text{ABS} \circ W(p)$$

where ABS is a function which counts the number of tokens in a bag by mapping each token-colour into the only element of a singleton $E = \{\epsilon\}$. This means that we in \bar{W} forget about all colour-information and thus \bar{W} is that, of the weight-functions overlying W , which provides us with the least information.

5. AN EXAMPLE: THE TELEPHONE SYSTEM

In this section we describe a telephone system, as it may be viewed by users. The status of a phone may change from INACTIVE to the situation, where the receiver has been LIFTed and you hear a CONTINUOUS tone. NEXT a number may be DIALled and you hear NO TONE, until you either hear a tone with SHORT intervals (indicating that the dialled phone is ENGAGED) or a tone with LONG intervals (indicating that the dialled phone is RINGING). In the latter situation the receiver may be LIFTed at the called phone and the two phones are CONNECTED until one of the two receivers are REPLACED. When a number is DIALled, a REQUEST is set up at the telephone exchange. If the dialled phone is INACTIVE, the REQUEST may be transformed to a CALL, and next to a CONNEXION, if the receiver is LIFTed at the called phone.

An HL-graph for the telephone system is shown in Figure 1. There are two different colour-sets: the set of all phone numbers, U , and the set of all pairs $U \times U$, where the first component represents the calling phone, while the second component represents the called phone. REQUEST, CALL and CONNEXION have $V = U \times U$ as the set of token-colours, while all

other places have U as the set of token-colours. ENGAGED is the complement of INACTIVE. This means that ENGAGED is marked with a colour $u \in U$ iff INACTIVE is not. In Figure 1 we have omitted the arcs which update ENGAGED.

Inspired by [6] three different kinds of arcs are used to indicate the three possible expressions: $x:U$, $y:U$, $(x,y):U \times U$. This visually splits the graph in three superposed parts, which describe the actions of a calling phone, a called phone and the telephone exchange, respectively. Initially $M_0(\text{inactive}) = \sum u = \sum u$ while all other places are unmarked.

The HL-graph in Figure 1 and its corresponding HL-tuple constitute a formal model, which allows us to determine even the more subtle properties of the specified telephone system. As an example, we can investigate what happens when a phone is calling itself; and it can be seen that a CONNEXION can be removed only by the calling phone and not by the called phone. If only an informal description was given, it would be easy to overlook some of these special cases.

The HL-graph in Figure 1 can easily be translated to an HL-tuple with the incidence-matrix shown in Figure 2. We have represented the negative and positive incidence-function in a single matrix, where negative terms belong to I_- , while positive belong to I_+ . By means of the transformation rules from [3] we can obtain a simplified matrix containing only four columns (c1-c4 in Figure 3). By simple inspection we find six weight-functions ($w1-w6$ in Figure 3) inducing six place-invariants ($i1-i6$ in Figure 4). It is easy to interpret the invariants in terms of the described system. As an example, $i6$ says that the RINGING phones are exactly those for which a CALL is waiting; $i4$ says that a phone is CONNECTED or REPLACED iff it is contained in a CONNEXION.

The functions ID , P_1 and P_2 are all simple linear functions, with multiplicity 1. This means that the underlying PT-net becomes the net of Figure 1, with all colour-information removed and all arcs having weight 1. The simplified incidence-matrix of Figure 3 has an underlying PT-matrix with 10 rows and 4 linear independent columns. Thus there is a basis of exactly 6 weight-functions inducing place-invariants of the underlying PT-net. The weight-functions underlying $w1-w6$ (in Figure 3) constitute such a basis.

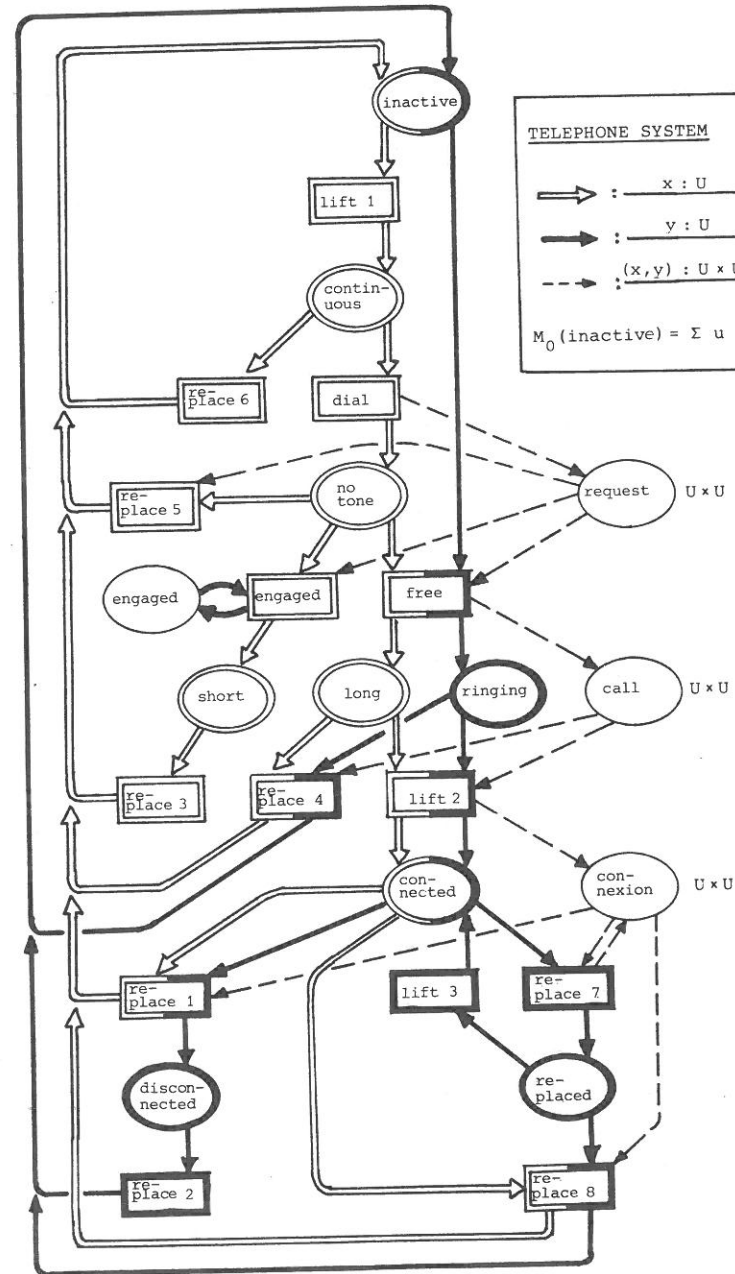


Figure 1: HL-graph describing a telephone system

TELEPHONE SYSTEM		lift 1	dial	free	engaged	lift 2	replace 1	replace 2	replace 3	replace 4	replace 5	replace 6	replace 7	lift 3	replace 8
		U	V	V	V	V	V	U	U	V	V	U	V	U	V
inactive	U	-ID		-P ₂			P ₁	ID	ID	P ₁ +P ₂	P ₁	ID			P ₁ +P ₂
continuous	U	ID		-P ₁								-ID			
no tone	U		P ₁	-P ₁	-P ₁						-P ₁				
long	U			P ₁			-P ₁								
short	U				P ₁						-P ₁				
ringing	U						-P ₂			-ID					
connected	U						P ₁ +P ₂							-P ₂	ID
disconnected	U														-P ₁
replaced	U														
engaged	U	ID			P ₂	-P ₂ +P ₂	-P ₁	-ID	-ID	-P ₁ -P ₂	-P ₁	-ID			-P ₁ -P ₂
request	V		ID	-ID	-ID										
call	V			ID							-ID				
connexion	V						ID	-ID						-ID+ID	-ID

Figure 2: Incidence-matrix for HL-net

TELEPHONE SYSTEM		c1	c2	c3	c4	M ₀	w1	w2	w3	w4	w5	w6
		U	V	V	V		U	U	U	U	U	U
inactive	U	-ID				Σ u	ID	ID				
cont, short, disc	U	ID	P ₁	P ₁ +P ₂	P ₁ +P ₂		ID					
no tone	U			-P ₁			ID		-ID			
long	U				-P ₁		ID				-ID	
ringing	U				-P ₂		ID					-ID
connec, replaced	U				-P ₁ -P ₂		ID				-ID	
engaged	U	ID						ID				
request	V		-ID						P ₁			
call	V				-ID						P ₁	P ₂
connexion	V				-ID						P ₁ +P ₂	

Figure 3: Simplified matrix, initial-marking and weight-functions for HL-net

TELEPHONE SYSTEM

$$(i1) \quad M(\text{inactive}) + M(\text{continuous}) + M(\text{short}) + M(\text{disconnected}) + M(\text{no tone}) + M(\text{long}) + M(\text{ringing}) + M(\text{connected}) + M(\text{replaced}) = \Sigma u$$

$$(i2) \quad M(\text{inactive}) + M(\text{engaged}) = \Sigma u$$

$$(i3) \quad M(\text{no tone}) = P_1 \circ M(\text{request})$$

$$(i4) \quad M(\text{connected}) + M(\text{replaced}) = (P_1 + P_2) \circ M(\text{connexion})$$

$$(i5) \quad M(\text{long}) = P_1 \circ M(\text{call})$$

$$(i6) \quad M(\text{ringing}) = P_2 \circ M(\text{call})$$

Figure 4: Place-invariants for HL-net

Acknowledgements

A number of useful suggestions for this work were made by Morten Kyng, Kurt Lautenbach, Mogens Nielsen, and P.S. Thiagarajan.

References

- [1] H.J. Genrich and K. Lautenbach, System modelling with high-level Petri nets, Theoretical Computer Science 13 (1981) 109-136.
- [2] K. Jensen, Coloured Petri nets and the invariant-method, Theoretical Computer Science 14 (1981) 317-336.
- [3] K. Jensen, How to find invariants for coloured Petri nets, in: Mathematical Foundation of Computer Science 1981, J. Gruska and M. Chytil (eds.), Lecture Notes in Computer Science, vol. 118, Springer Berlin 1981, 327-338.
- [4] K. Lautenbach, Liveness in Petri nets, Interner Bericht ISF-75-02.1, GMD Bonn (1975).
- [5] G. Memmi and G. Roucairol, Linear algebra in net theory, in: W. Brauer (ed.), Net theory and applications, Hamburg 1979, Lecture Notes in Computer Science vol. 84, Springer Berlin 1980, 213-223.
- [6] H. Oberquelle, Communication by graphic net representations, Bericht Nr. 75, Fachbereich Informatik, Universität Hamburg, 1981.