(Draft) Proceedings of the 7th IFAC/IFIP/IFORS/IMACS/ISPE Symposium on Information Control Problems in Manufacturing Technology. 1992

UPN: A Petri Net Based Graphical Representation for Company Policy Specifications in CIM

G. Harhalakis, C.P. Lin
Systems Research Center
University of Maryland

P.R. Muro-Medrano
Electrical Engineering and Computer Science Department
University of Zaragoza, Spain

August 30, 1991

Abstract

A graphical representation schema - Updated Petri Nets (UPN) - has been developed to model rule based company policy specifications, in the context of computer integrated manufacturing systems. UPN facilitates the modeling of relationships between operations of various related application systems and the database updates and retrievals among all the CIM databases. Based on this representation, a hierarchical modeling technique which includes refining and aggregating rules has also been developed. Application of the UPN is demonstrated in desinging rule based systems for controlling and integrating the information between manufacturing applications, including Computer Aided Design, Computer Aided Process Planning, Manufacturing Resources Planning, and Shop Floor Control.

1 Introduction

In a modern factory, besides *parts* being produced, there is also a tremendous amount of *data* being processed. For an efficient operation, it is necessary not only to control the manufacturing processes of products but also to manage and control the information flow among all the computerized manufacturing application systems that exist in a modern factory. The emphasis of most of the previous and current research projects is on individual aspects of CIM, such as developing a generic CIM architecture, creating a global database framework, or interfacing shop floor activities. However, the future in automation of modern factories will be based on a distributed environment which needs not only a generic database framework but also a controller, usually a knowledge rule-based system, to control the relationships between activities within all the computerized manufacturing application systems. Our approach is to develop such a control mechanism, in the form of a rule based system, for managing the information flow among all the existing and new manufacturing application systems, and to fill the gap between the high level production management and the low level factory automation [HARH 90] [HARH 91]. Similar approach has been used in [DILT 91] which, different from ours, emphasizes on the design of an integrated database framework and lacks of a formal modeling tool for validation and implementation.

Petri Nets [PETE 81] [MURA 89], which was adopted in the development of the UPN, are ideal for modeling dynamically and formally analyzing complex dynamic relationships of interacting systems. They were initially developed and used mainly for advanced computer integrated systems design, both in hardware and software, such as artificial intelligence in network systems [COUR 83], and for flexible manufacturing systems [CROC 86]. Most recent applications of Petri Nets in manufacturing systems are focusing again on the shop floor level, with a large number of work stations, robots, and transportation systems, to be handled by a central controller. Colored Petri Nets [ALLA 84] [KAMA 86] allow the model designer to work at different aggregation levels. The main advantage of Colored Petri Nets over General Petri Nets is the possibility of obtaining a compact representation of a large and complex system.

This paper is structured as follows. The second section defines the problem domain and research approach. Tthird section describes the features of UPN. The fourth section discuses UPN validation capabilities. The last section presents our conclusions with recommendations for future work.

2 Company Policy Specification for CIM

The objective of developing this UPN is to be able to model and validate the set of complex rules and procedures of the company policy specification and particularly apply to Computer Integrated Manufacturing. This paper, aiming at linking product and process design, manufacturing operations and production management, focuses on the control of information flow between each of the key manufacturing applications at the factory level, including Computer Aided Design (CAD), Computer Aided Process Planning (CAPP), Manufacturing Resource Planning (MRP II), and Shop Floor Control (SFC) systems. This linkage between manufacturing application systems involves both the static semantic knowledge of data commonalities and the dynamic control of functional relationships. The common data entities, which form the basis of the integrated system, include: *Parts, Bills of Material* in CAD, *Parts, Bills of Material, Work Centers, Routings* in CAPP, *Parts, Bills of Material, Routings, Work Centers, Manufacturing Orders* in MRP II, *Parts, Routings, Work Centers, Manufacturing Orders* in SFC. The functional relationships deal with the inter-relationships of functions within those applications.

An example, which represents the releasing of a work center in MRP II, is explained in natural language as follows. Invoking the work center release transaction in MRP II triggers a set of consistency checks, which are as follows: the WC I.D. provided must exist in MRP II with hold status; all the required data fields should have been filled, and any data fields left out by users are requested at this stage. If all these checks are satisfied, the system changes the work center status code from 'hold' to 'released', and a skeletal work center record is automatically created in the work center file in CAPP, with its status set to 'working'.

The design and maintenance of this company policy specification starts from user defined rule specifications, reflecting a specific company policy, which is then modeled using UPN. The next step is to convert the UPN model into a set of General Petri Nets (GPN) for validation purposes, and feed the results back to the user to resolve (i) conflicting company rules and (ii) errors introduced during the modeling phase. After the model has been validated, a parser translates the UPN model into a rule specification language. In short, the input is a set of company rules and the output is an AI production system for controlling operations, accessibility and updates of data within the manufacturing applications involved.

3 Updated Petri Nets

We have developed the Updated Petri Nets (UPN), which is a specialized type of the Colored Petri Nets (CPN) [JENS 87], and a hierarchical modeling methodology with a systematic approach for the synthesis of separate nets. The use of UPN allows the model designer to work at different levels of abstraction. Once we have this net we can selectively focus the analysis effort on a particular level within the hierarchy of a large model.

An UPN is a directed graph with three types of nodes: places which represent facts or predicates, primitive transitions which represent rules or implications, and compound transitions which represent meta-rules (sub-nets). Enabling and causal conditions and information flow specifications are represented by arcs connecting places and transitions.

Formally, an UPN is represented as: $UPN =< P, T, C, I^-, I^+, M_0, I_o, MT >$, where:

1. P, T, C, I^-, I^+, M_0 represent the classic Color Petri net definition. They identify the part of the information system that provide the conditions for the information control. Only this part of the UPN net is used in the validation process. These terms are defined as follows [JENS 87]: $P = \{p_1, ..., p_n\}$ denotes the set of places (circles) and $T = \{t_1, ..., t_m\}$ the set of primitive transitions (black bars), where $P \bigcap T = \emptyset$ and $P \bigcup T \neq \emptyset$. C is the color function defined from $P \bigcup T$ into non-empty sets. It attaches to each place a set of possible token-data and to each transition a set of possible data occurrence. I^- and I^+ are negative and positive incidence functions defined on $P \times T$, such that $I^-(p,t), I^+(p,t) \in [C(t)_{MS} \rightarrow C(p)_{MS}]_L \;\; \forall (p,t) \in P \times T$ where S_{MS} denotes the set of all finite multisets over the non-empty set S, $[C(t)_{MS} \rightarrow C(p)_{MS}]$ the multiset extension of $[C(t) \rightarrow C(p)_{MS}]$ and $[\ldots]_L$ denote a set of linear functions. The net has no isolated places or transitions. The initial marking, M_0, is a funtion defined on P, such that: $M_0(p) \in C(p), \forall p \in P$.

2. I_o is an inhibitor function defined on $P \times T$, such that:

Attibute	*Color set*	*DB data type*	*Description*
wcid	$WCID$	identification	identification number
des	DES	text	description
dep	DEP	text	department
cap	CAP	integer	capacity
sts	$MSTS$	$\{h, r\}$ (hold, release)	work center status code
ste	$MSTE$	$\{na, av\}$ (not avail., avail.)	work center state code
res	RES	text	Resource code
esd	ESD	date	Effectivity start date
Complete data structure for work center in MRP II			
$Mwc(wcid, des, dep, cap, sts, ste, res, esd)$			

Table 1: Data information.

$$I_o(p,t) \in [C(t)_{MS} \rightarrow C(p)_{MS}]_L, \quad \forall(p,t) \in P \times T.$$

3. $MT = \{mt_1, ..., mt_l\}$ denotes the set of compound transitions (represented graphically as blank bars), these are transitions which will be refined into more detailed subnets.

We have divided the representation of the domain knowledge in the following four groups: *Data, Facts, Rules, Metarules. Data* and relations between different data are used in relational database management systems. *Facts* are used to declare a piece of information about some data, or data relations in the system. The control of information flow is achieved by *Rules*. Here, we are considering domains where the user specifies information control policies using "if then" rules. Rules are expressed in UPN by means of transitions and arcs. Metaknowledge, in the form of metarules, is represented by net aggregation and hierarchical net decomposition (compound transition), and will be detailed below.

The example shown in section 2, which represents the releasing of a work center in MRP II, is now modeled in UPN, as shown in figure 1 to illustrate the corresponding component of UPN.

3.1 Data

In an information system environment, the user needs to refer to atomic data, and establish relations between different data by structuring information into composed data objects. UPN allows the specification of atomic and composed data objects. As an example, let us suppose that a work center record in MRP II can be in one of two different status: r (release), h (hold). An atomic data object can be illustrated by the status set: $sts = \{r, h\}$. Furthermore, composed data objects used in UPN are a subset of the Cartesian product $S_1 \times S_2 \times \ldots \times S_n$, where S_i is a set of atomic data. An example of composed data object can be illustrated by the work center relation in MRP II with the record name as Mwc. Due to the specialized domain of this representation schema and database update, a special syntax is used to identify database relations: $< R > (< A_1 >, ..., < A_n >)$, where $< R >$ is the database relation and $< A_i >$ is the ith attribute of that relation. An example of the work center relation in MRP II is listed in table 1.

3.2 Facts

Facts in UPN are represented by places and the tokens in the places. The fact asserted by one place is determined by the place name and its content (the colors of tokens in it). We represent facts about a work center record in MRP II with two places: $EMwc$, to describe the records that have been already introduced in the MRP II database, and $NMwc$, which represents the negation of this fact. The UPN syntax of a fact within the Database is $< R > (< A_1 >=< Val_1 >, ..., < A_n >=< Val_n >)$, where $< R >$ is the database relation, $< A_i >$ is the ith attribute of that relation, and $< Val_i >$ is the value or a corresponding variable of the ith attribute. These facts can be seen in figure 1 where they are used to represent some user specifications (places $p_1, p_2, p_3, p_4, p_5, NMwc, EMwc, NPwc$ and $EPwc$).

3.3 Rules

Rules are expressed in UPN as the combination of two entities: transitions and the arcs with their associated functions connecting the transition with its input/output places. Arcs identify information flow and flow conditions. UPN provide different types of arcs:

Enabling arcs are directed arcs which connect a place with a transition defining a precondition for that transition. They indicate which data must mark each place in order to enable a transition as well as which data must be removed from that place on firing. In order to be closer to the formal view of the net, let us focus for example on transition t_5 from figure 1. Firstable, the color sets for the involved places and transitions must be identified:

$C(EMwc) = MWC = WCID \times DES \times DEP \times CAP \times MSTS \times MSTE \times RES \times ESD$
$C(NPwc) = WCID$
$C(EPwc) = PWC = WCID \times DES \times DEP \times PSTS$
$C(p_5) = WDDCS = WCID \times DES \times DEP \times CAP$
$C(t_5) = MWCSDD = WCID \times DES \times DEP \times CAP \times MSTE \times RES \times ESD \times DES \times DEP$
Color sets $WCID, DES, DEP, CAP, MSTS, MSTE, RES, ESD, PSTS$ are as specified in table 1.

Functions in I^- and I^+ are defined in terms of lambda expresions having the form $f(c) = \lambda(V)exp(c)$, where $c \in C(t)$. For transition t_5, V and $c \in MWCSDD$ can be represented as follows:

$V = wcid\#, des0, dep0, cap\#, mste0, res0, esd0, des\#, dep\#$ and
$c = wcid, deso, depo, cap, msteo, reso, esdo, des, dep$

The enabling arcs for t_5 are:

- $I^-(t_5, p_5) : exp = \begin{bmatrix} wcid\# \\ des\# \\ dep\# \\ cap\# \end{bmatrix}$, $\lambda(V)exp \in [MWCSDD_{MS} \rightarrow WDDCS_{MS}]_L$ such that
  $\lambda(V)exp(c) = wcid, des, dep, cap$

- $I^-(t_5, EMwc) : exp = [\ wcid = wcid\#\ ]$, $\lambda(V)exp \in [MWCSDD_{MS} \rightarrow MWC_{MS}]_L$ such that $\lambda(V)exp(c) = wcid, despo, depo, -, -, msteo, reso, esdo$
- $I^-(t_5, NPwc) : exp = [\ wcid\#\ ]$, $\lambda(V)exp \in [MWCSDD_{MS} \rightarrow WCID_{MS}]_L$ such that $\lambda(V)exp(c) = wcid$

Causal arcs are directed arcs which connect a transition/action with a place/fact and define a post-condition for the transition/action. Causal arcs describe modifications to be performed in the state of the net when the transition/rule is fired, and more concretely, they indicate which colors must be added to a place on firing. For example, these are the causal arcs for transition t_5 :

- $I^+(t_5, EMwc) : exp = \begin{bmatrix} wcid = wcid\# \\ des = des\# \\ dep = dep\# \\ cap = cap\# \\ sts = r \end{bmatrix}$ $\lambda(V)exp \in [MWCSDD_{MS} \rightarrow MWC_{MS}]_L$ such that $\lambda(V)exp(c) = wcid, deso, depo, cap, r, msteo, reso, esdo$
- $I^+(t_5, EPwc) : exp = \begin{bmatrix} wcid = wcid\# \\ des = des\# \\ dep = dep\# \\ sts = w \end{bmatrix}$ $\lambda(V)exp \in [MWCSDD_{MS} \rightarrow PWC_{MS}]_L$ such that $\lambda(V)exp(c) = wcid, des, dep, cap, w$

Checking arcs indicate which data must mark each place in order to enable a transition but no data is removed. It can be represented as an enabling and causal arc together. The arcs connecting $EMwc$ and t_4 is an example being shown in figure1.

Additional **predicates** can be attached to the transitions, which represent additional conditions applied on the values of variables used in the surrounding arcs. For example: a predicate, $cap\# \leq 1000$, may be attached to transition t_4 to assure that the capacity entered by the user is within a valid range.

3.4 Meta-Rules

Metaknowledge and hierarchical net descriptions are represented by *Metarules* (expressed by compound transitions of the UPN) and mainly used in UPN as a mechanism to define sub-nets. They are used in two different directions to allow a structural and hierarchical composition of the domain knowledge:

Horizontal metarules relate rules at the same level of abstraction and allow the aggregation of rules under specific criteria. For example, the relationship of rules shown in figure 1 is a horizontal metarule. The formal representation of that subnet is specified by its incidence functions. Tables 3 and 2 illustrate the expresions (from the lambda expresions) for that incidence functions, where:

$E = \{\varepsilon\}$
$MWCSC = WCID \times DES \times DEP \times CAP \times MSTE \times RES \times ESD \times DES \times DEP \times CAP$
$WDDC = WCID \times DES \times DEP \times CAP$

exp for I^-		$t_{2,1}$	$t_{2,2}$	$t_{2,3}$	$t_{2,4}$	$t_{2,5}$
		E	WCID	MWCS	MWCSC	MWCS
NMwc	WCID	0	[*wcid#*]	0	0	0
EMwc	MWC	0	0	$\begin{bmatrix} wcid = wcid\# \\ sts = r \end{bmatrix}$	$\begin{bmatrix} wcid = wcid\# \\ des = des\# \\ dep = dep\# \\ sts = h \end{bmatrix}$	[*wcid* = ...]
NPwc	WCID	0	0	0	0	[*wcid* ...
EPwc	PWC	0	0	0	0	0
$p_{2,1}$	E	*abs*	0	0	0	0
$p_{2,2}$	WCID	0	[*wcid#*]	[*wcid#*]	[*wcid#*]	0
$p_{2,3}$	WCID	0	0	0	0	0
$p_{2,4}$	WCID	0	0	0	0	0
$p_{2,5}$	WDDC	0	0	0	0	[*wci... des dep cap*]

Table 2: Expresions for the negative incidence fuctions.

exp for I^+		$t_{2,1}$	$t_{2,2}$	$t_{2,3}$	$t_{2,5}$	
		E	WCID	MWCS	MWCSC	MV...
NMwc	WCID	0	[*wcid#*]	0	0	
EMwc	MWC	0	0	$\begin{bmatrix} wcid = wcid\# \\ sts = r \end{bmatrix}$	$\begin{bmatrix} wcid = wcid\# \\ des = des\# \\ dep = dep\# \\ sts = h \end{bmatrix}$	[*wcid# des... dep... cap... s*]
NPwc	WCID	0	0	0	0	
EPwc	PWC	0	0	0	0	[*wcid# des# dep# cap# s*]
$p_{2,1}$	E	0	abs	0	0	
$p_{2,2}$	WCID	[*wcid#*]	0	0	0	
$p_{2,3}$	WCID	0	[*wcid#*]	0	0	
$p_{2,4}$	WCID	0	0	[*wcid#*]	0	
$p_{2,5}$	WDDC	0	0	0	$\begin{bmatrix} wcid\# \\ des\# \\ dep\# \\ cap\# \end{bmatrix}$	

Table 3: Expresions for the positive incidence fuctions.

Vertical metarules establish relationships between one rule and other rhich define knowledge at a lower level of abstraction and allow a structure of rules to fo[r]bstraction hierarchy.

4 Hierarchical Modeling Approach

Generally speaking, any "company policy" starts from the specification [ev]eral global rules which describe aggregate operations for a given entity within the sys[tem. T]hese rules are then further refined into more detailed specifications on a step by ste[p basi]s, until no aggregate operations are left. Following a similar concept, a hierarchical [model]ing method using UPN has been developed which allows the system designer to sta[rt fro]m abstract global nets and continue with successive refinements until the desired degree of detail has been reached. In addition to the refinement of rules within each scenario, a technique is needed to synthesize all the scenarios to form a coherent net representing the company-wide policy for all entities in the system. Some work in hierarchical representation using Petri Nets has been done for various applications [NARA 85]. The hierarchical modeling methodology adopted here incorporates:

Top-down stepwise refinement technique for the modeling of each scenario from an abstract and aggregate level to a detailed level. This approach necessitates the development of new Petri Net modeling entities which include two types of transitions as mentioned in the previous section; one to represent primitive rules, and the other to represent metarules which can be further refined into sub-nets. The connections are represented by calls from one compound transition of the net at the abstract level to the sub-nets at the more detailed level, and an example is shown in figure 2. The transition,where the call was made, is formed by a calling net which contains one input transition (`ti`), one waiting place (`pw`), and one output transition (`to`). There are, for each sub-net being called, an arc connecting from the input transiton to that sub-net and a returning arc back to its output transition. The interface between the input transition and the sub-net being called is a place representing the initiation of the sub-net and the interface between the output transition and that sub-net is a place representing the satisfaction of the sub-net.

Synthesis technique for synthesizing separate nets, which represent different scenarios of the system, to form a coherent net. Our modeling approach is capable of incorporating the modeling of the databases of the manufacturing application systems involved, using UPN, by defining the database states as global variables and interfacing the application procedures (company policy) through the default modification procedure (system dependent) and places representing database states, and synthesizing nets through them systematically. More details can be found in [HARH 91]

5 Knowledge verification

One of the major objective of creating a KBS using Petri Nets is the ability of validating the KBS mathematically and systematically. Completeness (dead-end rules, unfirable rules), consistency (redundant rules, subsumed rules, under-constrained rules), and conflicts, are the major issues in knowledge/rule validation [NGUY 87], [LOPE 90]. The incidence matrices of Petri Nets representing the rule base can be used to perform some of these validation checks and verify them with the aid of specific domain knowledge. Several other analysis techniques for Petri Nets, including, reachability trees, behavioral nets, and net invariants, are also used [MURA 89] [MART 82]. The net invariants, which represent mutually exclusive conditions within the "company policy", can reveal logical conflicts in the specification of the original rules and possibly errors introduced during the modeling process. The reachability tree can be used to detect any deadlocks or inconsistencies in the model. The behavioral net can be used to detect redundancies in the net and is a useful tool for reducing the complexity of the model. Some reduction rules have also been investigated for reducing the complexity of nets prior to the analysis phase [HARH 91]. However, these analysis techniques were initially developed for Generalized Petri Nets (GPN), and do not apply to Colored Petri Nets (CPN) which are characterized by a great diversity of linear functions that are associated to their arcs. This fact introduces a high complexity in the development and execution of these algorithms. We have taken the approach of *unfolding* UPN into GPN before they can be analyzed [HARH 91].

6 Conclusions

A formal structured representation schema for rule based systems has been developed and demonstrated with information integration for manufacturing applications. The representation schema, called UPN, is based on the graphical and formal capabilities of colored Petri nets to express and validate if-them rules. The UPN is capable of representing user specification rules as well as database updates and retrievals, which is necessary for controlling and integrating information within current and future distributed database systems. Related rules can be aggregated at the same level of abstraction and the relation between one rule at a given level of abstraction and a set of aggregated rules at a lewer level of abstraction is also allowed. These facilities provide a mechanism for step wise refinement in modeling and validation.

References

[ALLA 84] Alla, H, Ladet, P., Martinez, J., and Silva, M., "Modeling and validation of complex systems by coloured Petri nets: application to a flexible manufacturing system", *Advances in Petri Nets 1984, pp. 15-31*, 1984.

[COUR 83] M. Courvoisier, R. Vallette, et al. "A Programmable Logic Controller Based on A High Level Specification Protocol", *Proceedings of IECON Conference on Industrial Electronics, pp. 174-179*, 1983.

[CROC 86] Crockett, D.H. and Desrochers, A.A., "Manufacturing Workstation Control Using Petri-Nets", *Technical Report 83, Robotics and Automation Laboratory, Department of Electrical, Computer, and System Engineering, Rensselaer Polytechnic Institute*, August 1986.

[DILT 91] Dilts, D.M. and Wu, W. "Using Knowledge-Based Technology to Integrate CIM Databases ", *IEEE Transaction on Data and Knowledge Engineering, vol.3, no. 2, pp. 237-245* ,1991.

[HARH 90] Harhalakis, G., Lin, C., Hillion, H., and Moy, K., "Development of a Factory Level CIM Model", *Journal of Manufacturing Systems, vol. 9, no. 2, pp. 116-128*,1990.

[HARH 91] Harhalakis, G., Lin, C.P., Mark, L., and Muro, P., "Formal Representation, Verification and Implementation of Rule Based INformation Systems for Integrated Manufacturing (INSIM)", *Technical Report TR 91-19, Systems Research Center*, University of Maryland, College Park, 1991.

[HSU 87] Hsu, C., Angulo, C., Perry, A., and Rattner, L., "A Design Method for Manufacturing Information Management", *Proceedings of Conference on Data and Knowledge Systems for Manufacturing and Engineering, Hartford, Connecticut, pp. 93-102*,1987.

[JENS 87] Jensen, K., "Colored Petri Nets", *Petri Nets: Central Models and Their Properties. Advances in Petri Nets 1986, Part I. Proceedings of an Advanced Course, Bad Honnef, 8-19. September 1986, pp. 248-299*, Edited by G. Goos and J. Hartmanis. Springer-Verlag Berlin Heidelberg 1987.

[KAMA 86] Kamath, M. and Viswanadham, N., "Applications of Petri net based models in the modeling and analysis of flexible manufacturing systems". *Proc. IEEE Int. Conf. on Robotics and Automation., pp. 312-316*, 1986.

[LOPE 90] Lopez B., Meseguer P. and Plaza E.: "Knowledge Based Systems Validation: A State of the Art". *AI Communications, Vol.3, No. 2, pp. 58-72*. June, 1990.

[MARK 87] Mark, L. and Roussopoulos, N., "Operational Specification of Update Dependencies", *Systems Research Center Technical Reprot No. SRC TR-87-37, University of Maryland*, 1987.

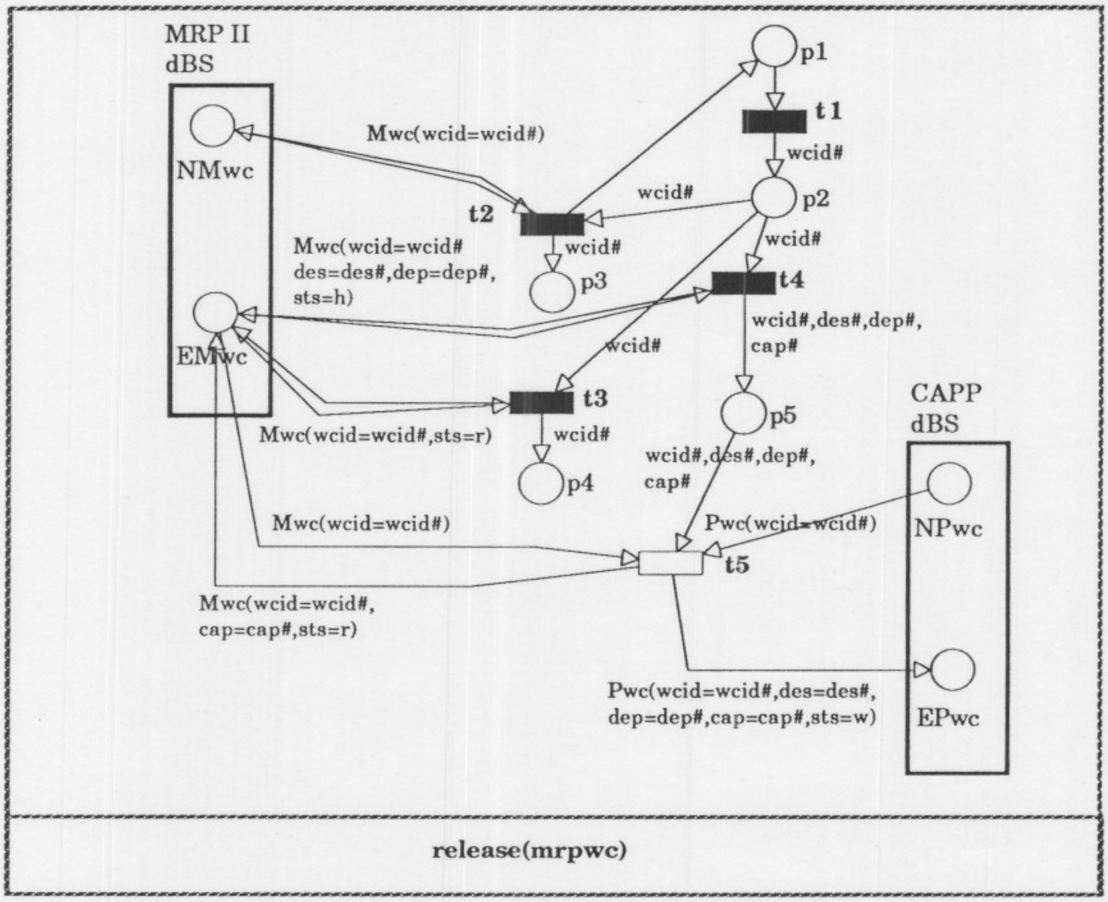
[MART 82] Martinez, J. and Silva, M." A Simple and Fast Algorithm to Obtain All Invariants of A Generalised Petri Net",. *Second European Workshop on Application and Theory of Petri Nets, pp. 301-310*, 1982.

[MURA 89] Murata, T., "Petri Nets: Properties, Analysis and Applications", *Proceedings of The IEEE, vol. 77, no. 4, pp. 541-580*, April 1989.

[NARA 85] Narahari, Y. and Viswanadham, N., "A Petri Net Approach to the Modeling and Analysis of Flexible Manufacturing Systems", *Annala of Operations Research, vol. 3, pp. 381-391*, 1985.

[NGUY 87] Nguyen T.A., Perkins W.A., Laffey T.J. and Pecora D.: "Knowledge Base Validation". *AI Magazine, summer, pp. 67-75*. 1987.

[PETE 81] Peterson, J.L., "Petri Net Theory and the Modeling of Systems", Prentice Hall, Englewood Cliffs, New Jersey, 1981.



t1: request and read wcid
t2: write error message and restart
t3: write error message
t4: request other information
t5: update work center record in MRP II dBase with sts=r, and additional data, insert a work center record in CAPP dBase

p1: user starts the transaction
p2: wcid is provided
p3: work center ID does not exist in MRP II
p4: work center already has 'r' status in MRP II
p5: all the necessary data is provided
EMwc: existence of work center in MRP II dBase
NMwc: non-existence of work center in MRP II dBase
EPwc: existence of work center in CAPP dBase
NPwc: non-existence of work center in CAPP dBase

Figure 1: Subnet of the work center creation scenario "Release of a work center in MRP II".

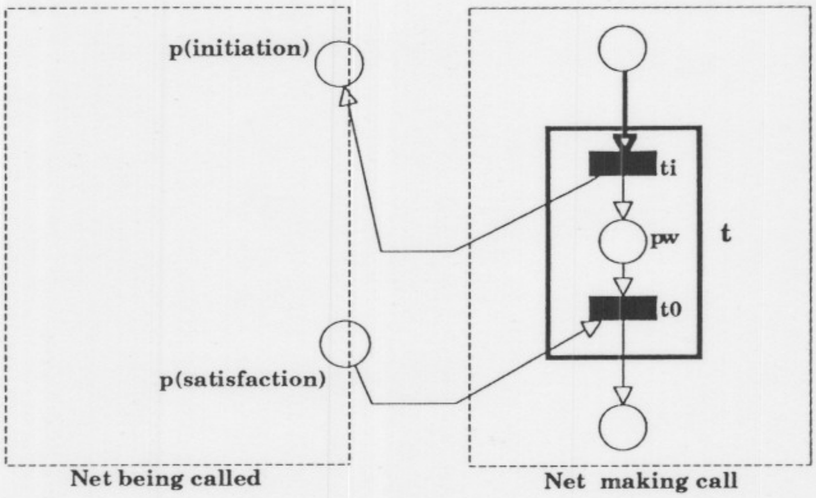


Figure 2: Example of call between UPN sub-nets