

Hierarchies of Place/Transition Refinements in Petri Nets

W.M. Zuberek

Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada A1C-5S7

I. Bluemke

Institute of Computer Science
Warsaw University of Technology
00-661 Warsaw, Poland

Abstract – Place and transition refinements provide a convenient method of structuring complex net models by replacing single elements (places and transitions) at a ‘higher-level’ of abstraction with ‘lower-level’, more detailed, subnets. The concepts of static and dynamic place/transition refinements are introduced. Dynamic refinements do not increase the size of the (refined) model because no ‘expansion’ of the model is performed; instead, only a ‘logical’ association of higher-level elements with lower-level subnets is maintained and used in model analysis. Multiple applications of place/transition refinements results in hierarchical net models. The paper formalizes the concept of hierarchies of refinements in Petri nets and shows simple applications of the hierarchical approach to modeling of manufacturing cells.

1. INTRODUCTION

Petri nets are popular formal models of discrete-event systems in which events can occur concurrently, but which contain constraints on precedence and frequency of such occurrences [7, 6]. Multiprocessing systems, computer networks and distributed databases, as well as manufacturing systems are a few example of such systems. Like any formal models, Petri nets can be used for specification, modeling and analysis of concurrent systems; the popularity of Petri net models is due to a simple and accurate representation of concurrency. However, since the realistic systems tend to be complex, their models easily become too complicated to be handled efficiently. Therefore, an abstraction mechanism is needed to simplify large models by identifying submodels and replacing them by simple elements in a way that preserves the most important properties of the original models. Also, a refinement mechanism must be provided to enhance the general ‘higher-level’ models with more specific ‘lower-level’ details by substituting lower-level subnets for simple elements of the higher-level models. This second mechanism, refinement of places and transitions in Petri net models, is discussed in this paper in greater detail.

First approaches to structuring of Petri nets were proposed in late eighties [2, 4, 9]. The intention of these methods was to reduce the complexity of the model by

dividing it into a number of submodels. For colored (hierarchical) nets [4] used in modeling of complex systems (CPN/Design system), five structuring constructs were proposed: transition substitution, place substitution, transition invocation, place fusion and transition fusion. The first two of them, transition and place substitutions, are similar to transition and place refinements discussed in this paper, but they assume a different mechanism for interaction with the refinement environment. The remaining structuring constructs are used for composing several models into a single one.

This paper formalizes and unifies several informal methods proposed for net refinements with a variety of hierarchical and modular net models. The interaction between a refining (sub)net and its environment is described in a very general way by a pair of mappings; specialized versions of these mappings correspond to specific cases of place and transition refinements. The proposed mappings can easily be extended to cover a more general case of net refinements.

Place and transition refinements can be static or dynamic. In the static case, a simple element (place or transition) is actually replaced by a more detailed subnet, creating a new net model, usually much larger than the original one. In dynamic replacement the expansion of the model does not take place, so the original model does not change; dynamic replacements only introduce associations of (higher-level) elements with their (lower-level) expansions, so the same refining subnet can be associated with any number of elements without any increase in the model size. Moreover, the refining subnets can also contain further refinements, so a typical hierarchical structure of models is created.

2. PETRI NETS

A place/transition, marked Petri net \mathbf{N} [6, 7] can be defined as a quadruple $\mathbf{N} = (P, T, A, m_0)$, where:

P is a finite, nonempty set of places,

T is a finite, nonempty set of transitions,

A is a set of directed arcs which connect places with transitions and transition with places, $A \subseteq P \times T \cup$

$T \times P$, such that for each transition there exists at least one place connected with it,

m_0 is an initial marking function which assigns a (non-negative) number of tokens to each place of the net, $m_0 : P \rightarrow \{0, 1, \dots\}$.

For each place $p \in P$, its input set $Inp(p)$ contains all transitions connected to p by directed arcs, $Inp(p) = \{t \in T \mid (t, p) \in A\}$, while its output set is $Out(p) = \{t \in T \mid (p, t) \in A\}$. Input and output sets of transitions are defined similarly.

Let any function $m : P \rightarrow \{0, 1, \dots\}$ be called a marking in a net $\mathbf{N} = (P, T, A, m_0)$.

A transition t is enabled by a marking m iff every input place of t is assigned at least one token by m . Every transition enabled by a marking m can fire. When a transition fires, a single token is removed (simultaneously) from each of its input places and a token is added to each of its output places. This determines a new marking in a net, a new set of enabled transitions, and so on. The set of all markings that can be derived from the initial marking is called the set of reachable markings of a net.

Net models of many real-life systems, like communication networks or multiprocessor systems, are quite complicated, and contain numerous simpler subnets (e.g., models of processors in a multiprocessor system). In such cases, it is usually advantageous to disregard many low-level details of the model (which do not affect its more significant properties) and to analyze an approximate model at a higher level of abstraction. A systematic conversion from such higher levels of abstraction into more detailed models can be done by place and/or transition refinements.

3. PLACE/TRANSITION REFINEMENTS

The concept of place/transition refinements can be formalized as a *refinement system*. A refinement system \mathcal{R} is defined as a 5-tuple, $\mathcal{R} = (\mathbf{N}_0, \mathcal{N}, \rho, \phi, \psi)$, where:

\mathbf{N}_0 is a place/transition (initial) marked net, $\mathbf{N}_0 = (P_0, T_0, A_0, m_{0,0})$,

\mathcal{N} is a family of place/transition (refinement) nets, $\mathcal{N} = \{\mathbf{N}_1, \dots, \mathbf{N}_k\}$,

ρ is a (partial) refinement function which associates elements of P_0 (place refinements) and T_0 (transition refinements) with nets from \mathcal{N} , $\rho : P_0 \cup T_0 \rightarrow \{1, \dots, k\}$, so each place $p \in P_0$ is refined by the net $\mathbf{N}_{\rho(p)}$ (if $p \in Dom(\rho)$, otherwise p remains a simple place), and each transition $t \in T_0$ is refined by $\mathbf{N}_{\rho(t)}$ (if $t \in Dom(\rho)$).

ϕ and ψ are (input and output) interface functions which define the interconnections between the input and output sets of a place (or transition) and its refinement determined by ρ ; for each $p \in P_0$, if

$p \in Dom(\rho)$, then $\phi(p) : T_0 \rightarrow 2^{P_{\rho(p)}}$ and $\psi(p) : T_0 \rightarrow 2^{P_{\rho(p)}}$; similarly, for each $t \in T_0$, if $t \in Dom(\rho)$, then $\phi(t) : P_0 \rightarrow 2^{T_{\rho(t)}}$ and $\psi(t) : P_0 \rightarrow 2^{T_{\rho(t)}}$.

A refinement system \mathcal{R} defines a Petri net $\mathbf{N} = (P, T, A, m_0)$ such that:

- $P = \{p_i \in P_0 \mid p_i \notin Dom(\rho)\} \cup \{p_{i,j} \mid p_i \in Dom(\rho) \wedge p_j \in P_{\rho(p_i)}\}$,
- $T = \{t_i \in T_0 \mid t_i \notin Dom(\rho)\} \cup \{t_{i,j} \mid t_i \in Dom(\rho) \wedge t_j \in T_{\rho(t_i)}\}$,
- $A = \bigcup_{x_\ell \in P_0 \cup T_0} \{a_{\ell,i} \mid a_i \in A_{\rho(\ell)}\} \cup \{(t_\ell, p_{i,j}) \mid p_i \in Dom(\rho) \wedge t_\ell \in Inp(p_i) \wedge p_j \in \phi(p_i, t_\ell)\} \cup \{(p_{i,j}, t_\ell) \mid p_i \in Dom(\rho) \wedge t_\ell \in Out(p_i) \wedge p_j \in \psi(p_i, t_\ell)\} \cup \{(p_\ell, t_{i,j}) \mid t_i \in Dom(\rho) \wedge p_\ell \in Inp(t_i) \wedge t_j \in \phi(t_i, p_\ell)\} \cup \{(t_{i,j}, p_\ell) \mid t_i \in Dom(\rho) \wedge p_\ell \in Out(t_i) \wedge t_j \in \psi(t_i, p_\ell)\}$,
- $\forall (p \in P) m_0(p) = \begin{cases} m_{0,0}(p), & \text{if } p \notin Dom(\rho); \\ m_{\rho(p_i),0}(p_j), & \text{if } p = p_{i,j}. \end{cases}$

An illustration of place and transition refinements is shown in Fig.1 for a very simple net model. Fig.1(a) shows a high-level model containing a simple cycle composed of two steps; for example, this net can model a “producer” (or a production system) that cyclically produces an item (t_1) and stores it (t_2); the place p_2 represents the condition “ready to produce another item” and the place p_2 the condition “a new item is produced”.

The net shown in Fig.1(b) is a simple free-choice net \mathbf{N}_1 that can be used as a refinement of the place p_1 . It can represent a more elaborate additional phase of producing an item, which in some cases involves operation represented by t_3 and in other cases operation represented by t_4 . Moreover, Fig.1(c) shows a cyclic net \mathbf{N}_2 which can represent a “consumer”; it is also composed of two steps: t_5 representing fetching an item and t_6 representing consuming it. This net can be used as a refinement of transition t_2 of the net shown in Fig.1(a), creating a simple producer-consumer model (with no buffering; the buffer can easily be introduced as another refinement of t_2 in Fig.1(a)). The net with both refinements (performed in any order) is shown in Fig.1(d).

For this example, the refinement function ρ is:

$P_0 \cup T_0$	ρ
p_1	1
p_2	undefined
t_1	undefined
t_2	2

and the interface functions ϕ and ψ are:

ϕ	t_1	t_2	p_1	p_2
p_1	$\{p_{1,3}\}$	Φ	Φ	Φ
t_2	Φ	Φ	$\{t_{2,5}\}$	Φ

ψ	t_1	t_2	p_1	p_2
p_1	Φ	$\{p_{1,4}\}$	Φ	Φ
t_2	Φ	Φ	Φ	$\{t_{2,5}\}$

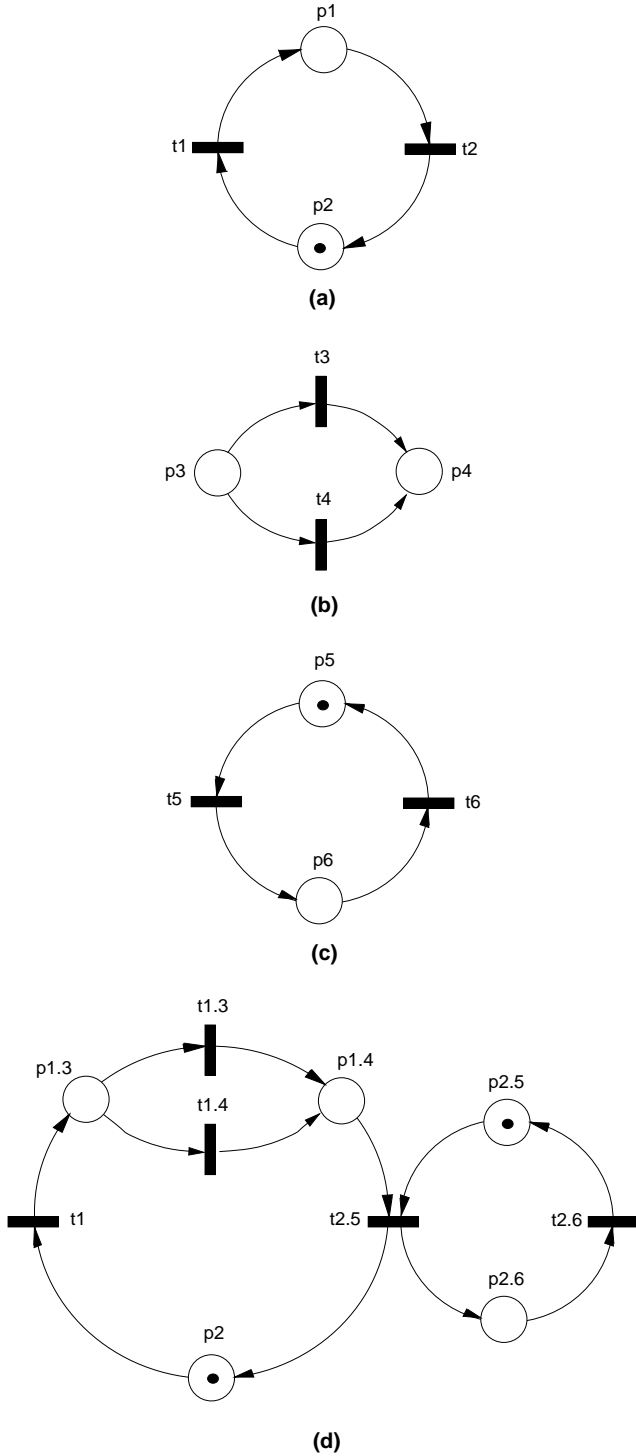


Fig.1. Place/transition refinements of a simple model.

In a refinement system \mathcal{R} , each net $\mathbf{N}_i \in \mathcal{N}$ can be another refinement system, $\mathbf{N}_i = \mathcal{R}_i = (\mathbf{N}_{i,0}, \mathcal{N}_i, \rho_i, \phi_i, \psi_i)$, so a hierarchical Petri net model can be derived from the concept of a refinement system. For example, the net shown in Fig.1(a) can be obtained from an “elementary” net, containing just one place and one transition, shown in Fig.2(a), by using the net shown in Fig.2(b) as a re-

finement of the transition t_3 .

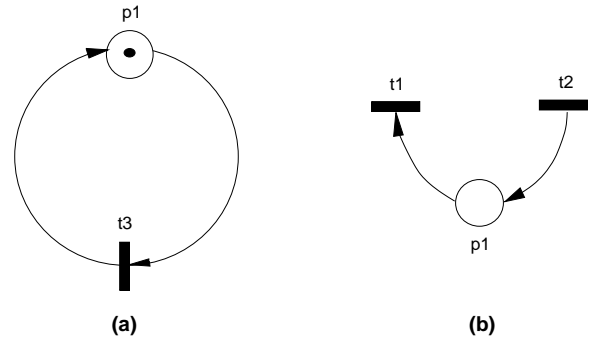


Fig.2. Transition refinement in a basic net.

A hierarchy of models can be represented by a lattice in which the collection of (partially refined) models is the set of elements, the top (the least upper bound) is the most abstract model (usually it is a trivial, elementary net), while the bottom (the greatest lower bound) is the most detailed, fully expanded model. Moving ‘top-down’ in this lattice corresponds to the refinement operations, while moving ‘bottom-up’ corresponds to the operation of abstraction. The structure of the lattice reflects different ways (i.e., different orders) of refinements which result in the same model.

For the nets shown in Fig.1 and Fig.2, such a hierarchy of models is shown in Fig.3(a,) where the top element, n_0 , is the net of Fig.2(a), the bottom one, n_5 , is the net of Fig.1(d), the net corresponding to n_1 is obtained by the refinement of p_1 in the net shown in Fig.2(a) using the net of Fig.1(b), the net n_2 is obtained by the refinement of t_3 using the net of Fig.2(b) (n_2 is isomorphic with the net shown in Fig.1(a)), and the net n_3 is shown in Fig.3(b).

4. HIERARCHICAL MODELING

The idea of hierarchical modeling can be applied very conveniently to systematic development of Petri net models of manufacturing cells [3]. It is known that for a manufacturing cell with m machines there are $m!$ simple schedules [8]. One of these schedules is optimal for any given set of temporal specifications [11]. Petri net models of these schedules can be systematically derived from a state-based analysis of the cell [10]. On the other hand, the same models can also be systematically derived by a series of place/transition refinements.

The net shown in Fig.4(a) is the simplest possible model of a cyclic process which (at a very high level of abstraction) represents processing of consecutive parts in a manufacturing cell. This ‘elementary’ net can be expanded using the subnet shown in Fig.4(b) as a refinement of the transition t_1 ; the resulting net, shown in Fig.4(d), is a more detailed model of a one-machine cell; the transition $t_{1.1}$ represents the machine performing its operation, while $t_{1.2}$ and $t_{1.3}$ model the operations performed by the

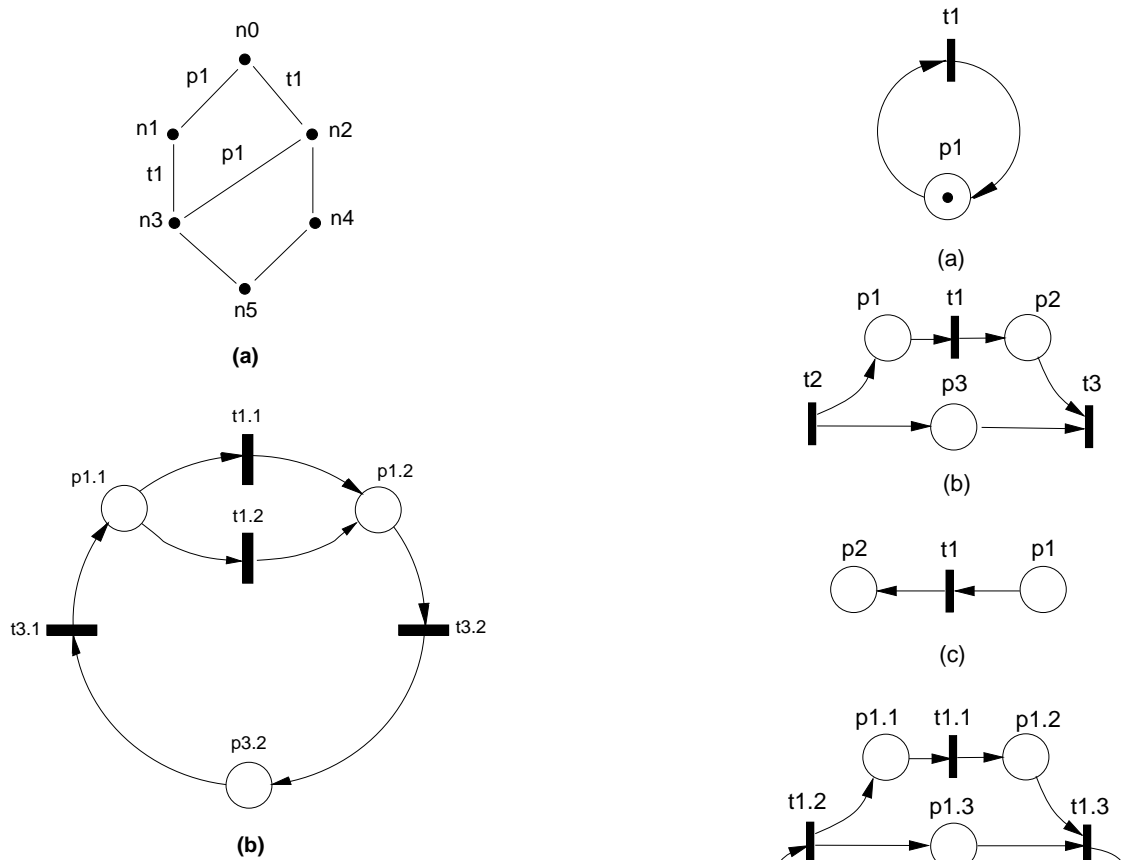


Fig.3. Hierarchy of place/transition refinements.

robot: it picks a new part from an input conveyor, carries it to the machine and loads it; the place $p_{1.1}$ represents the condition “part loaded” (so the machine can begin its operation), and $p_{1.2}$ represents the condition “machine operation finished”, so the robot (‘waiting’ in place $p_{1.3}$) can unload the part, carry it to the output conveyor and drop it (the transition $t_{1.3}$). In order to represent the move of the robot from the output ($t_{1.3}$) to the input ($t_{1.2}$) of the cell, the place p_1 is further refined using the subnet shown in Fig.4(b). The resulting net (after renaming the transitions and places) is shown in Fig.4(e); it clearly shows the cyclic behavior of the robot ($t_{01}, p_{11}, t_{12}, p_{20}, t_{20}, p_{02}, t_{01}$) as well as the path followed by each part entering the cell ($t_{01}, p_{10}, t_1, p_{12}, t_{12}$).

The model of a one-machine cell (Fig.4(e)) can be further refined into a model of a two-machine cell by using the same subnet from Fig.4(b) as a refinement of the transition t_{12} in Fig.4(e). The resulting net is shown in Fig.5(a). It should be observed that an identical (in fact, isomorphic) model is obtained if the same subnet is used as a refinement of the transition t_{01} in Fig.4(e).

A different model of a two-machine cell, shown in Fig.5(b), is obtained by refining the transition t_{12} in the net shown in Fig.4(e) using the subnet shown in Fig.4(f). A subsequent refinement of the place p_{11} , using the sub-

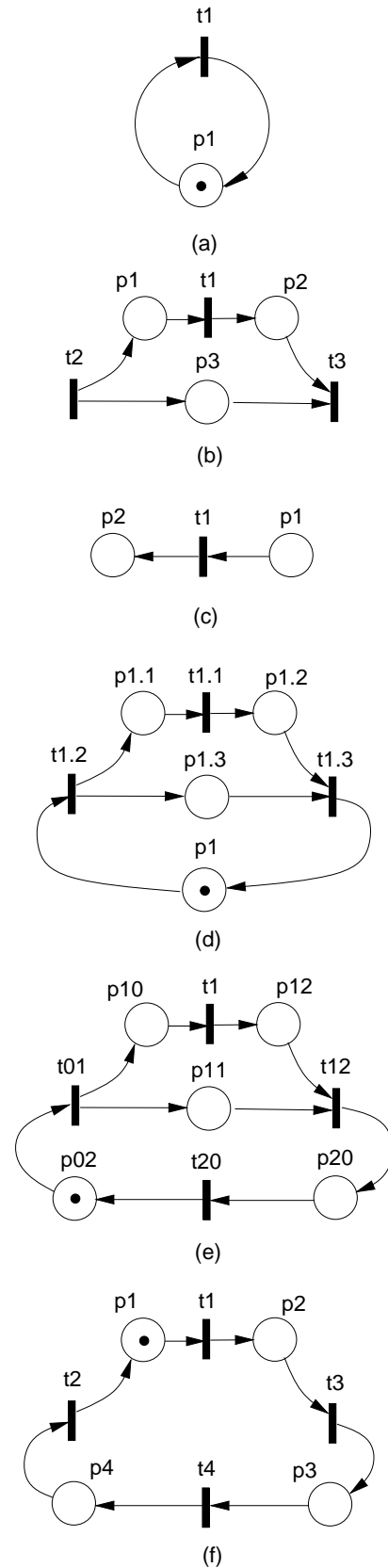


Fig.4. Refinements of models of a manufacturing cell.

net shown in Fig.4(c), produces the net shown in Fig.5(c). This model represents a more complicated cycle of the robot, when an attempt is made to use the two machines of the cell concurrently. The cycle of the robot is composed of the following operations:

t_i	operations
t_{01}	pick from input, move to machine-1 and load
$t_{11.1}$	move from machine-1 to machine-2
$t_{12.2}$	unload machine-1, move to machine-2, load
$t_{12.3}$	unload machine-2, move to output and drop
$t_{12.4}$	move from output to machine-1
t_{20}	move from machine-2 to input

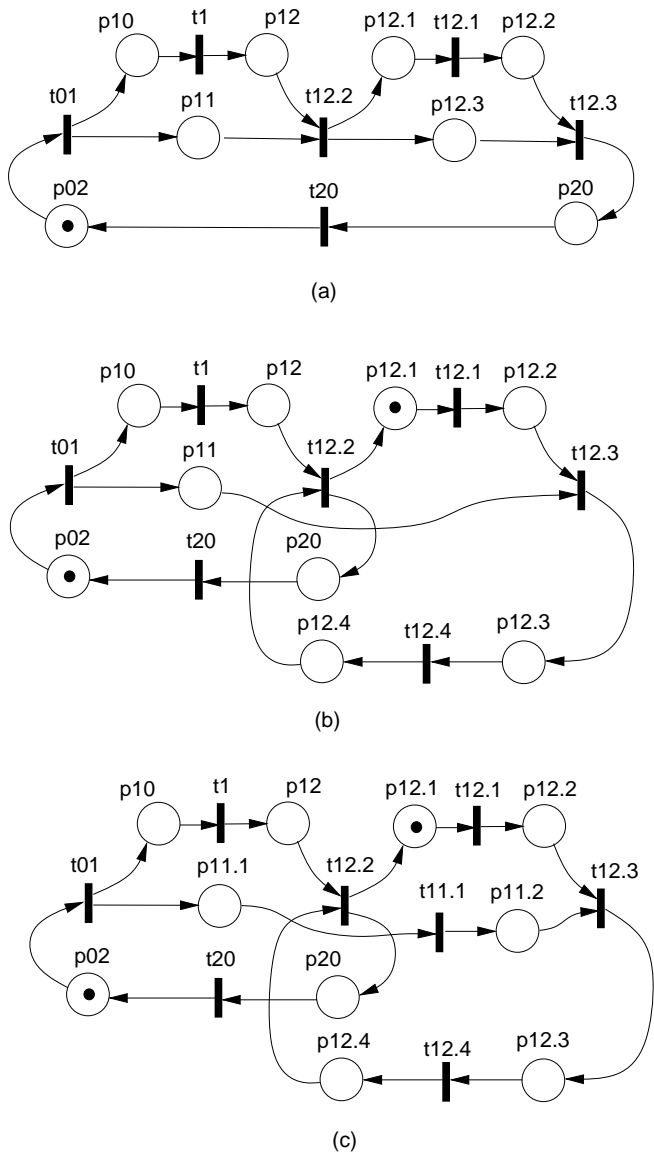


Fig.5. Models of two-machine manufacturing cells.

Petri net models of manufacturing cells with three machines can be obtained by (similar) refinements of two-machine models. Fig.6(a) shows one such model which

can be obtained by the refinement of any one of transitions $t_{12.3}$, $t_{12.2}$ or t_{01} in Fig.5(a) using the subnet of Fig.4(b). Similarly, the model shown in Fig.6(b) is obtained by using subnet shown in Fig.4(f) as a refinement of $t_{12.3}$ in Fig.5(a), while Fig.6(c) shows the result of refinement of $t_{12.3}$ in Fig.5(c) using the subnet shown in Fig.4(b).

It is expected that place/transition refinements can be used for a systematic derivation of models of composite schedules [11], i.e., schedules in which more than one part enters (and leaves) the cell in one cycle. It has been shown that models of such schedules can be systematically derived from simple schedules [11]; since simple schedules can be obtained by place and transition refinements, a combination of such refinements with a composition of simple schedules should provide an alternative method of deriving composite schedules.

5. CONCLUDING REMARKS

The hierarchical structure of place and transition refinements can be used for very efficient analysis of net models, especially in the case of structural methods; in this case the properties of a more abstract, simpler model are combined with the properties of the refining subnet. A number of general properties (liveness, boundedness) are preserved for large classes of the refining subnets.

The class of net refinements can easily be extended to more general subnet refinements (rather than just refinements of a place or transition) by appropriate modifications of the interface functions ϕ and ψ in the definition of the refinement system.

An implementation of hierarchical modeling with dynamic refinement of places and transitions as an extension of an existing set of software tools for analysis of Petri net models is being investigated.

The concept of hierarchical refinements applied to the class of timed net models, i.e., net models in which the duration of activities is also taken into account, opens an interesting area of hierarchical performance analysis [1, 5].

REFERENCES

- [1] Buchholz, P., "Hierarchical Markovian models: symmetries and reduction"; Performance Evaluation, vol.22, no.1, pp.93-110, 1995.
- [2] Fehling R., "A concept for hierarchical Petri nets with building blocks"; 12-th Int. Conf. on Application and Theory of Petri Nets, Aarhus, Denmark, pp.370-389, 1991.
- [3] Hillion, H.P.: "Timed Petri nets and application to multi-stage production system"; in: **Advances in Petri Nets 1989** (Lecture Notes in Computer Science 424); pp.281-305, Springer Verlag 1989.
- [4] Huber, P., Jensen, K., Shapiro, R.M., "Hierarchies in colored Petri nets"; in: **Advances in Petri Nets 1990** (Lecture Notes in Computer Science 483), pp.313-341, Springer-Verlag 1991.

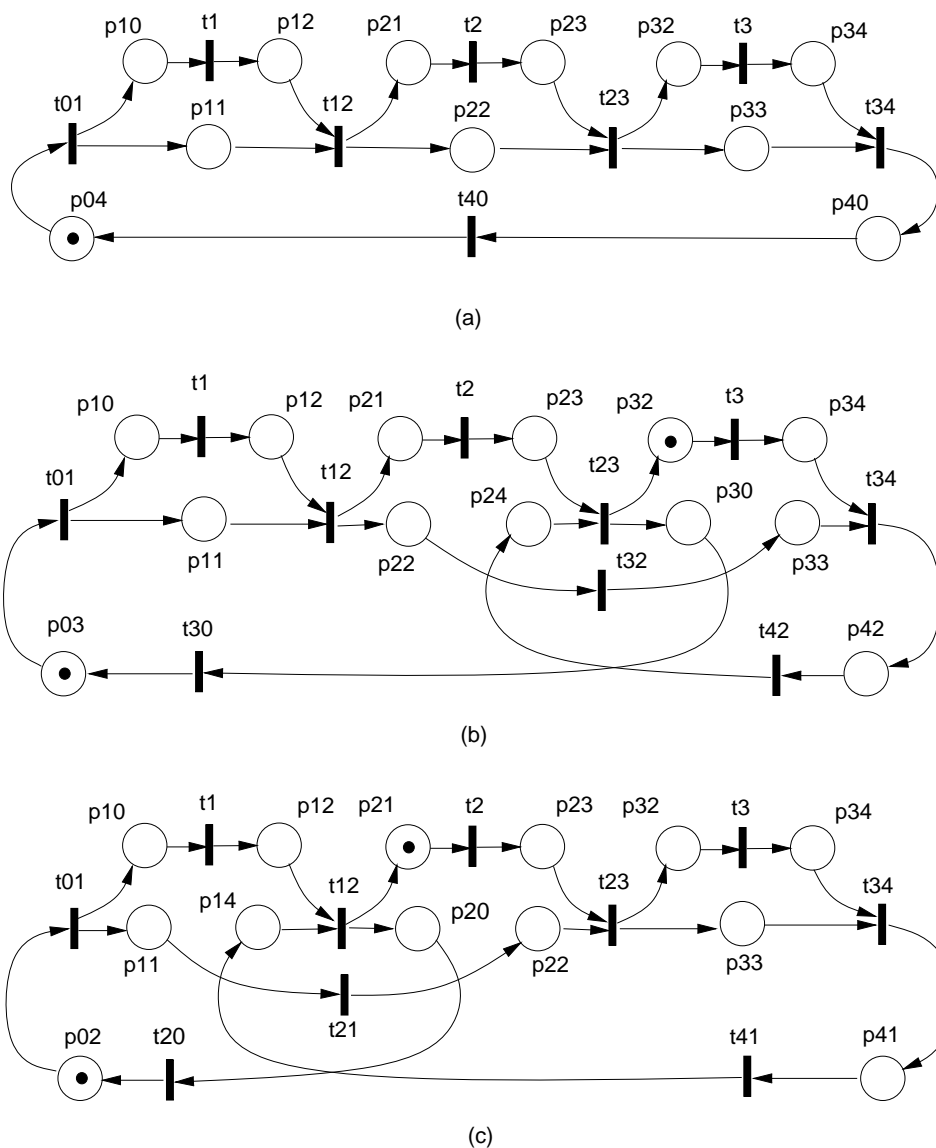


Fig.6. Models of three-machine manufacturing cells.

[5] Hariri, S., Mutlu, H., “Hierarchical modeling of availability in distributed systems”; *IEEE Trans. on Software Engineering*, vol.21, no.1, pp.50-56, 1995.

[6] Murata, T.: “Petri nets: properties, analysis and applications”; *Proceedings of IEEE*, vol.77, no.4, pp.541-580, 1989.

[7] Reisig, W.: **Petri nets - an introduction** (EATCS Monographs on Theoretical Computer Science 4); Springer Verlag 1985.

[8] Sethi, S.P., Sriskandarajah, C., Sorger, G., Blazewicz, J., Kubiak, W.: “Sequencing of parts and robot moves in a robotic cell”; *Int. Journal of Flexible Manufacturing Systems*, vol.4, pp.331-358, 1992.

[9] Vogler, W., **Modular construction and partial order semantics of Petri nets** (Lecture Notes in Computer Science 625); Springer Verlag 1992.

[10] Zuberek, W.M., Kubiak, W.: “Modeling simple schedules of manufacturing cells using timed Petri nets”; *Proc. Int. Workshop on Intelligent Systems and Innovative Computations*, Tokyo, Japan, 1994.

[11] Zuberek, W.M., “Applications of timed Petri nets to modeling and analysis of flexible manufacturing cells”; Technical Report #9503, Department of Computer Science, Memorial University of Newfoundland, St.John’s, Canada A1B 3X5, 1995 (available through anonymous ftp on ftp.cs.mun.ca in pub/techreports/tr_9503.ps.Z).