INRIA/IEEE Conference on Emerging Technologies and Factory Automation (ETFA'95) –
Novel Applications of Petri Nets in Automation; Paris, France, 10-13 Oct. 1995, pp.311-322.
Copyright © 1995 IEEE (DOI 10.1109/ETFA.1995.496672).

# APPLICATION OF TIMED PETRI NETS TO MODELING THE SCHEDULES OF MANUFACTURING CELLS

*W.M. Zuberek*

*Department of Computer Science*
*Memorial University of Newfoundland*
*St. John's, Canada A1B–3X5*
*e-mail: wlodek@cs.mun.ca*

**Abstract.** Timed Petri nets are proposed as models of simple and composite schedules for a large clas of manufacturing cells. Net models of simple schedules can easily be derived from the possible sequences of robot actions. Models of composite schedules can be obtained by different compositions of simple schedules. Timed net models can be evaluated using one of typical methods developed for analysis of timed Petri nets, for example, invariant analysis. Performance characterization (the cycle time or the throughput) obtained in this way can be used for maximization of the cell's performance. Because the number of different schedules grows very quickly with the number of machines as well as the length of the (composite) schedule, colored Petri nets are proposed as a uniform representation of entire classes of schedules. Simple examples illustrate the proposed approach.

## 1. Introduction

In flexible manufacturing systems, machines are often grouped into manufacturing cells (or robotic cells), in which a robot performs sequences of pickup, move, load, unload and drop operations, transporting the manufactured parts from one machine of the cell to another [S3BK92, Cl83]. The throughput of the cell depends on the sequence of robot activities as well as on the sequence in which different parts enter the cell [DH90]. Any approach to maximizing the throughput of a robotic cell must be able to deal efficiently with two issues: how to generate alternative schedules for a given cell, and how to evaluate these schedules. Usually the schedules are represented by models which capture the essential characteristics of the schedule, but which remove all details which are inessential for the evaluation process.

The behavior of manufacturing cells is represented by 'events' and 'activities'; an activity corresponds to an operation performed by a machine or the robot, while an event represents a change of cell's activities. Different sets of activities determine the 'states' of the system. In each state, several activities can occur concurrently, for example, several machines can perform their operations simultaneously and the robot can also transport a part. Petri nets provide a simple and convenient formalism for modeling systems that exhibit parallelism and concurrency [Mu89, Re85].

In order to study performance aspects of Petri net models, the duration of activities must also be taken into account and included into model specifications. Several types of Petri nets 'with time' have been proposed by assigning 'firing times' to the transitions or places of a net. In timed nets, transition firings are 'real–time' events, i.e., tokens are removed from input places at the beginning of the firing period, and they are deposited to the output places at the end of this period (sometimes this is also called a "three–phase" firing mechanism). The firing times may be either deterministic or stochastic, i.e., described by some probability distribution function. In both cases the concepts of state and state transitions have been formally defined and used in derivation of different performance characteristics of the model [Zu91].

Analysis of net models can be based on their behavior (i.e., the space of reachable states) or on the structure of the net; the former is called reachability analysis while the latter structural analysis. Invariant analysis [Re85] seems to be the most popular example of the structural approach. Structural methods eliminate the derivation of the state space, so they avoid the 'state explosion' problem of reachability analysis, but they cannot provide as much information as the reachability approach does. Quite often, however, all the detailed results of reachability analysis are not really needed, and more synthetic performance measures, that can be obtained by structural methods, are quite satisfactory [Hi89]. In particular, the throughput of a timed net model can easily be determined from the structure of a net if the net can be decomposed into a set of conflict–free or free–choice elementary nets [ZK93].

The steady-state behavior of manufacturing cells is considered for two types of schedules, the so called simple schedules in which exactly one (new) part enters the cell and one leaves the cell in each cycle, and composite schedules which deal with several (new) parts in each cycle. In both cases, timed Petri net models are derived and are solved using the invariant analysis. The solutions are obtained in symbolic form which means that the analysis needs to be performed only once, and then specific values of performance characteristics can easily be obtained by simply evaluating the symbolic solutions for different sets of parameter values. Examples of simple and composite schedules for a 3–machine cell illustrate the proposed approach.

## 2. Simple Schedules

For simple schedules, exactly one part enters and one leaves the cell in each cycle (although the part which leaves the cell may not be the same as the one which enters the cell). It is known [S3BK92] that for a cell with $m$ machines there are $m!$ different simple schedules. For $m = 3$ (Fig.1 shows a sketch of a 3–machine cell), there are six simple schedules, denoted here as A, B, C, D, E and F.

Assuming that each part follows the same path from the input ($In$) to machine–1 ($M_1$), to machine–2 ($M_2$), to machine–3 ($M3$), and finally to the output of the cell ($Out$), the simple schedules can be described by sequences of cell configurations where each cell configuration discribes a distribution of parts among the machines of the cell; more specifically, each configuration is an $m$–tuple of machine descriptions, and each machine description is equal to "1" if the machine is loaded with a part in this configuration, otherwise it is equal to "0" (in the case of multiple machines performing exactly the same operations, the values describing each multi–machine station would assume the values from "0" to "$n$" where $n$ is the number of identical machines). The schedules are as follows:

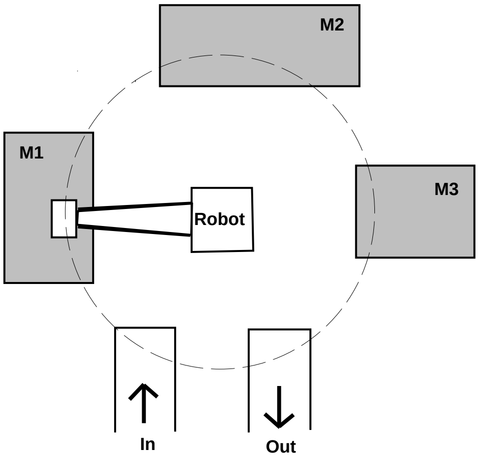

Fig.1. Layout of a three–machine cell.

A: $(0,0,0) \rightarrow (1,0,0) \rightarrow (0,1,0) \rightarrow (0,0,1) \rightarrow (0,0,0)$
B: $(0,0,1) \rightarrow (1,0,1) \rightarrow (0,1,1) \rightarrow (0,1,0) \rightarrow (0,0,1)$
C: $(0,0,1) \rightarrow (1,0,1) \rightarrow (1,0,0) \rightarrow (0,1,0) \rightarrow (0,0,1)$
D: $(0,1,0) \rightarrow (1,1,0) \rightarrow (1,0,1) \rightarrow (1,0,0) \rightarrow (0,1,0)$
E: $(0,1,0) \rightarrow (1,1,0) \rightarrow (1,0,1) \rightarrow (1,0,0) \rightarrow (0,1,0)$
F: $(0,1,1) \rightarrow (1,1,1) \rightarrow (1,1,0) \rightarrow (1,0,1) \rightarrow (0,1,1)$

The parts are transported between machines, the input and the output by the robot. The sequences of robot's actions can be derived from the sequences of configurations by "implementing" the moves of parts corresponding to changes of consecutive configurations. For example, schedule A begins be transporting a part from the input to $M_1$, and when the first operation is finished, unloading $M_1$, moving the part to $M_2$ and loading it, and so on. The sequences of robot actions are as follows (the robot moves from $X$ to $Y$ are denoted by $X \Rightarrow Y$ if the robot carries a part and by $X \rightarrow Y$ otherwise):

A: $In \Rightarrow M_1 \Rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow In$
B: $In \Rightarrow M_1 \Rightarrow M_2 \rightarrow M_3 \Rightarrow Out \rightarrow M_2 \Rightarrow M_3 \rightarrow In$
C: $In \Rightarrow M_1 \rightarrow M_3 \Rightarrow Out \rightarrow M_1 \Rightarrow M_2 \Rightarrow M_3 \rightarrow In$
D: $In \Rightarrow M_1 \rightarrow M_2 \Rightarrow M_3 \rightarrow M_1 \Rightarrow M_2 \rightarrow M_3 \Rightarrow Out \rightarrow In$
E: $In \Rightarrow M_1 \rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow M_1 \Rightarrow M_2 \rightarrow In$
F: $In \Rightarrow M_1 \rightarrow M_3 \Rightarrow Out \rightarrow M_2 \Rightarrow M_3 \rightarrow M_1 \Rightarrow M_2 \rightarrow In$

Timed Petri net models of simple schedules can easily be derived from the sequences of robot operations. In timed models, net transitions represent operations while net places represent 'conditions' (in the most general sense). A Petri net model of schedule A is shown in Fig.2. The three machines of Fig.1 (or rather machine operations) are represented by $t_1$, $t_2$ and $t_3$, each of these transition with its input and output place (for 'part loaded' and 'machine operation finished' conditions). The 'firing times' associated with these transitions, $f(t_1) = o_1$, $f(t_2) = o_2$ and $f(t_3) = o_3$, represent the (average) times of performing the operations on machines $M_1$, $M_2$ and $M_3$, respectively.

The operations of the robot are represented by the path $t_{01}$, $p_{11}$, $t_{12}$, $p_{22}$, $t_{23}$, $p_{33}$, $t_{34}$, $p_{40}$, $t_{40}$, $p_{04}$ modeling the sequence of consecutive steps of the schedule:

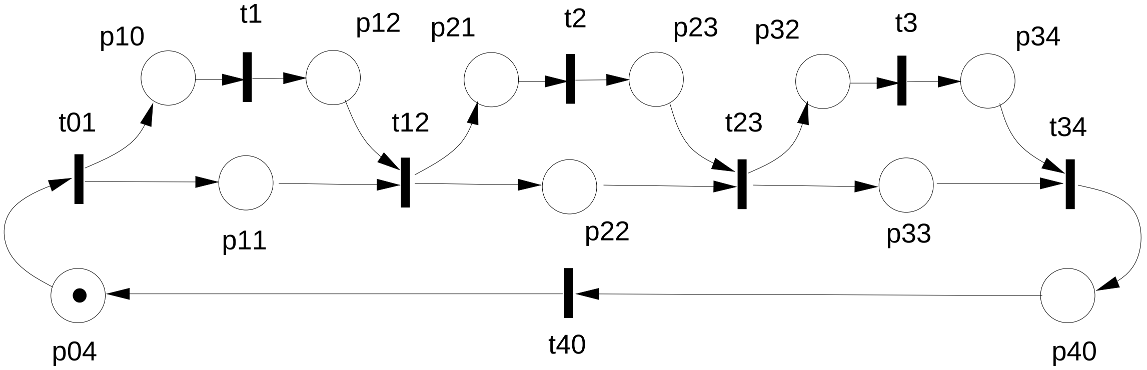

Fig.2. Petri net model of schedule A.

| | *robot operations* | *execution time* |
|---|---|---|
| $t_{01}$ | pick a part from *In*, move to $M_1$ and load | $u+w+y$ |
| $t_{12}$ | unload $M_1$, move to $M_2$ and load | $v+w+y$ |
| $t_{23}$ | unload $M_2$, move to $M_3$ and load | $v+w+y$ |
| $t_{34}$ | unload $M_3$, move to *Out* and drop | $v+x+y$ |
| $t_{40}$ | move from *Out* to *In* | $y$ |

where the *execution times* (or firing times of transitions) are given assuming that:

$u$ denotes the (average) pickup time,
$v$ denotes the (average) unload time,
$w$ denotes the (average) load time,
$x$ denotes the (average) drop time and
$y$ denotes the average ‘travel’ time between two adjacent machines (assuming, for simplicity, that this time is the same for all adjacent machines, and also the same for $M_3$ to *Out*, *Out* to *In* and *In* to $M_1$ moves).

It is also assumed that there is always an available part in *In* and that *Out* removes manufactured parts sufficiently quickly, so *In* and *Out* are not actually shown although they can easily be added to the model.

A Petri net model of schedule E is shown in Fig.3, in which $t_1$, $t_2$ and $t_3$ represent the machine operations, as in Fig.2, and the remaining transitions correspond to robot actions:

| | *robot operations* | *time* |
|---|---|---|
| $t_{01}$ | pick a part from *In*, move to $M_1$ and load | $u+w+y$ |
| $t_{12}$ | unload $M_1$, move to $M_2$ and load | $v+w+y$ |
| $t_{20}$ | move to $M_2$ to *In* | $2y$ |
| $t_{21}$ | move from $M_1$ to $M_2$ | $y$ |
| $t_{23}$ | unload $M_2$, move to $M_3$ and load | $v+w+y$ |
| $t_{34}$ | unload $M_3$, move to *Out* and drop | $v+x+y$ |
| $t_{41}$ | move from *Out* to $M_1$ | $2y$ |

Evaluation of net models using net invariants is described in [ZK93]. It appears that net models of simple schedules have only a few invariants subnets, and this invariant

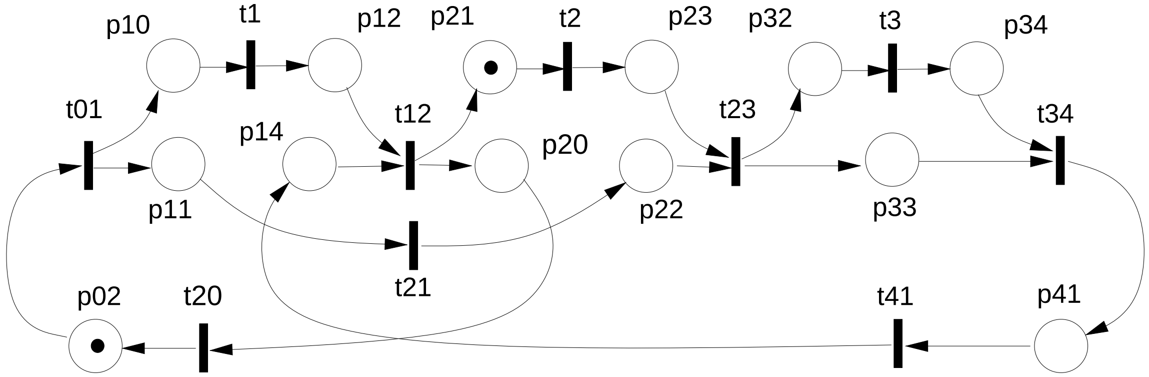

Fig.3. Petri net model of schedule E.

subnets determine the performance of the model. Symbolic formulas for performance characteristics can easily be derived using the invariant analysis. For example, the net shown in Fig.3 (schedule E) has 5 place invariants which imply the following subnets:

| $P$–$inv$ | $t_1$ | $t_2$ | $t_3$ | $t_{01}$ | $t_{12}$ | $t_{20}$ | $t_{21}$ | $t_{23}$ | $t_{34}$ | $t_{41}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 3 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

It should be observed that the set of transitions of invariant (3) is a subset of that of (2), and that the set of transitions of invariant (5) is a subset of that of (4). Consequently, the cycle time for this schedule is determined by the maximum cycle time of subnets (1), (2) and (4):

$$\tau_0 \;=\; \max(\tau_1, \tau_2, \tau_4)$$

where:

$$\begin{aligned}\tau_1 &= o_1 + u + v + 2w + 4y,\\ \tau_2 &= o_2 + o_3 + 3v + 2w + x + 5y,\\ \tau_4 &= o_3 + u + 3v + 3w + x + 9y.\end{aligned}$$

## 3. Composite Schedules

For composite schedules, several parts enter and leave the cell in each cycle. Models of composite schedules can be regarded as an interleaved composition of simple schedules, each of which corresponds to one part entering the cell within this cycle, moving from one machine to another, and finally leaving the cell. The "composition" can only occur in identical configurations of the schedules being composed. For example, a composition of simple schedules A and E can be as follows:

| *schedule A* | | *schedule E* |
|---|---|---|
| (0,0,0) | | |
| (1,0,0) | | |
| (0,1,0) | → | (0,1,0) |
| | | (1,1,0) |
| | | (1,0,1) |
| | | (1,0,0) |
| (0,1,0) | ← | (0,1,0) |
| (0,0,1) | | |
| (0,0,0) | | |

The sequence of robot's actions can be determined as before, by tracing the progress of parts through the cell:

$In \Rightarrow M_1 \Rightarrow M_2 \rightarrow In \Rightarrow M_1 \rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow M_1 \Rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow In$

The Petri net model is shown in Fig.4.

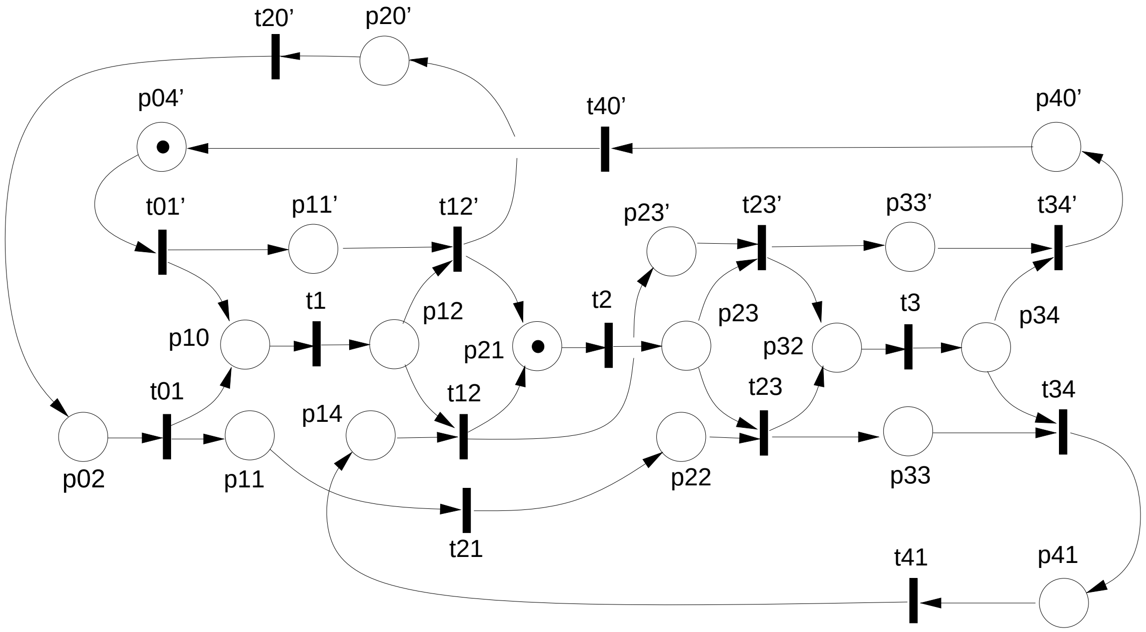

Fig.4. Petri net model of schedule (A+E).

Similarly, the composition of schedules B and F is:

| *schedule B* | | *schedule F* |
|---|---|---|
| (0,0,1) | | |
| (1,0,1) | | |
| (0,1,1) | | |
| (0,1,1) | → | (0,1,1) |
| | | (1,1,1) |
| | | (1,1,0) |
| | | (1,0,1) |
| (0,1,1) | ← | (0,1,1) |
| (0,1,0) | | |
| (0,0,1) | | |

and its Petri net model is shown in Fig.5.

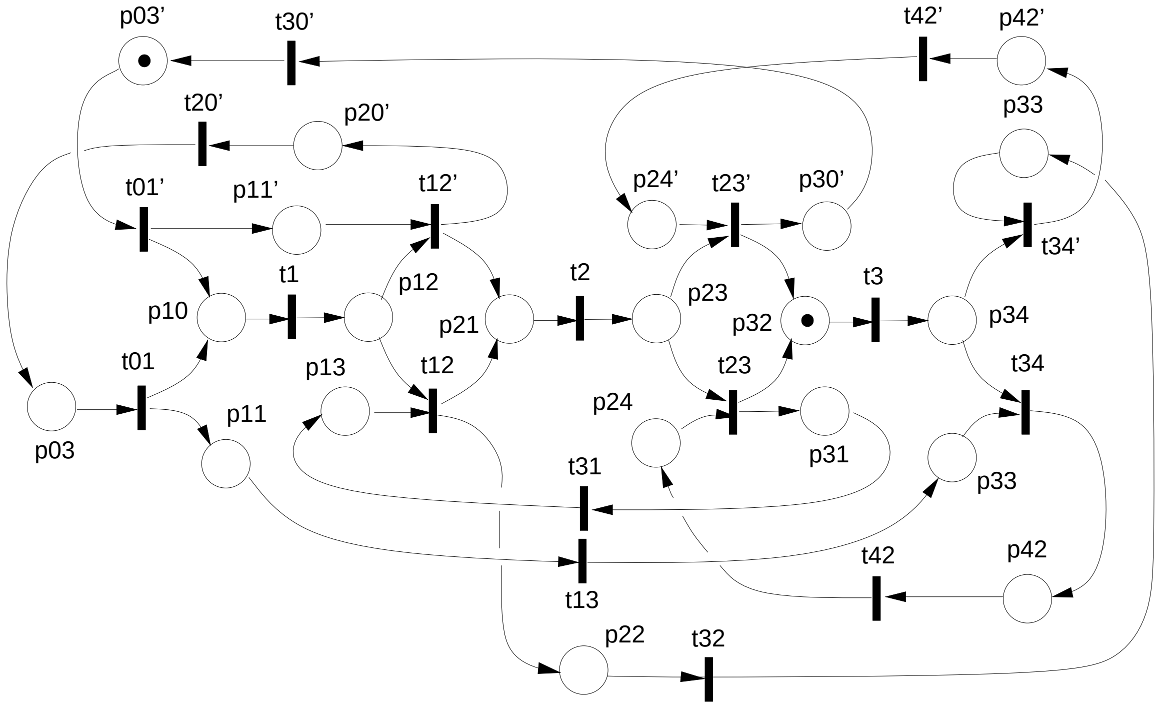

Fig.5. Petri net model of schedule (B+F).

More complex schedules can be derived in the same way. For example, a composition of schedules A, E and B can be as follows:

| *schedule B* | | *schedule A* | | *schedule E* |
|---|---|---|---|---|
| | | (0,0,0) | | |
| | | (1,0,0) | | |
| | | (0,1,0) | → | (0,1,0) |
| | | | | (1,1,0) |
| | | | | (1,0,1) |
| | | | | (1,0,0) |
| | | (0,1,0) | ← | (0,1,0) |
| (0,0,1) | ← | (0,0,1) | | |
| (1,0,1) | | | | |
| (0,1,1) | | | | |
| (0,1,0) | | | | |
| (0,0,1) | → | (0,0,1) | | |
| | | (0,0,0) | | |

## 4. Colored Net Models of Schedules

It can be observed that the number of schedules grows rather quickly with the number of machines as well as the length of composite schedules. Colored Petri nets [Je87, Zu90] are proposed as a uniform representation of entire classes of schedules, with token attributes (called 'colors') represeting individual schedules within the model.

A colored net model of all simple schedules for a 3–machine cell is shown in Fig.6. It can be observed that the three machines are represented, as before, by transitions $t_1$, $t_2$ and $t_3$, while the robot actions (for all six simple schedules) are represented by the remaining part of the net, some transitions of which are used selectively by some schedules (i.e., some colors).

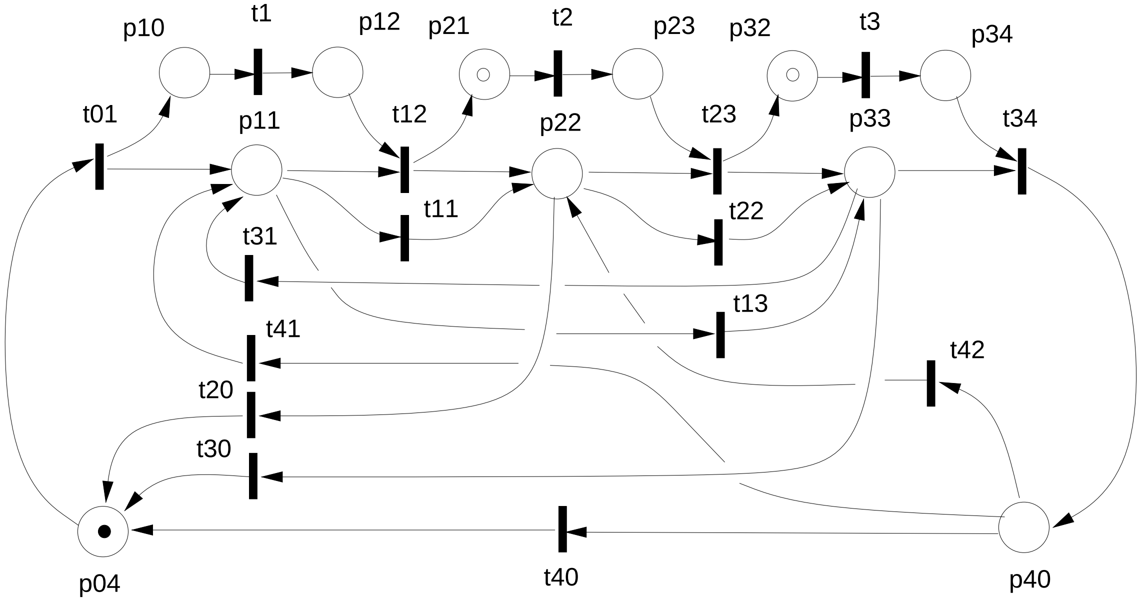

Fig.6. Colored net model of a three–machine cell.

There are six basic colors representing the schedules (and also denoted A, B, ..., F), and five auxiliary colors which are used for elimination of potential conflicts when the models of different schedules are combined together. These auxiliary colors are needed for schedules B, C, D, E and F, and are denoted by b, c, d, e i f; they are used as token colors only, so there are eleven token colors and only six occurrence colors (formally, the arc functions are partial functions which are undefined for the auxiliary occurrence colors).

The transitions correspond to the following actions (the column *schedules* indicates the occurrence colors of the corresponding transitions):

| | *robot's operation* | *schedules* | *execution time* |
|---|---|---|---|
| $t_1$ | $M_1$ operation | A,B,C,D,E,F | $o_1$ |
| $t_2$ | $M_2$ operation | A,B,C,D,E,F | $o_2$ |
| $t_3$ | $M_3$ operation | A,B,C,D,E,F | $o_3$ |
| $t_{01}$ | pick from $In$, move to $M_1$ and load | A,B,C,D,E,F | $u + w + y$ |
| $t_{11}$ | move from $M_1$ to $M_2$ | D,E | $y$ |
| $t_{12}$ | unload $M_1$, move to $M_2$ and load | A,B,C,D,E,F | $v + w + y$ |
| $t_{13}$ | move from $M_1$ to $M_3$ | C,F | $2y$ |
| $t_{20}$ | move from $M_2$ to $In$ | E,F | $2y$ |
| $t_{22}$ | move from $M_2$ to $M_3$ | B,D | $y$ |
| $t_{23}$ | unload $M_2$, move to $M_3$ and load | A,B,C,D,E,F | $v + w + y$ |
| $t_{30}$ | move from $M_3$ to $In$ | B,C | $2y$ |
| $t_{31}$ | move from $M_3$ to $M_1$ | D,F | $2y$ |
| $t_{34}$ | unload $M_3$, move to $Out$ and drop | A,B,C,D,E,F | $v + x + y$ |
| $t_{40}$ | move from $Out$ to $In$ | A,D | $y$ |
| $t_{41}$ | move from $Out$ to $M_1$ | C,E | $2y$ |
| $t_{42}$ | move from $Out$ to $M_2$ | B,F | $2y$ |

The *execution times* are the same for all transition occurrences.

The arc functions $w$ are mappings $C \to C \to \mathbf{N}$; for most of the arcs, these functions are (partial) identity functions for the basic colors A, B, ..., F, i.e., for an arc $a$, an occurrence color $o \in C$ and a token color $c \in C$:

$$w(a)(o)(c) \;=\; \begin{cases} 1, & \text{if } o, c \in \{\text{A,B,C,D,E,F}\} \wedge o = c, \\ 0, & \text{if } o, c \in \{\text{A,B,C,D,E,F}\} \wedge o \neq c, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

The definitions of all ‘non–standard’ arc functions are shown in the following table

| *arc* | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| $(t_{01}, p_{11})$ | A:1 | B:1 | c:1 | d:1 | e:1 | f:1 |
| $(t_{12}, p_{22})$ | A:1 | b:1 | C:1 | d:1 | e:1 | f:1 |
| $(t_{23}, p_{33})$ | A:1 | b:1 | c:1 | d:1 | E:1 | f:1 |
| $(t_{34}, p_{40})$ | A:1 | b:1 | c:1 | D:1 | e:1 | f:1 |
| $(p_{11}, t_{11})$ | — | — | — | d:1 | e:1 | — |
| $(p_{11}, t_{13})$ | — | — | c:1 | — | — | f:1 |
| $(p_{22}, t_{20})$ | — | — | — | — | e:1 | f:1 |
| $(p_{22}, t_{22})$ | — | b:1 | — | d:1 | — | — |
| $(p_{33}, t_{30})$ | — | b:1 | c:1 | — | — | — |
| $(p_{33}, t_{31})$ | — | — | — | d:1 | — | f:1 |
| $(p_{40}, t_{41})$ | — | — | c:1 | — | e:1 | — |
| $(p_{40}, t_{42})$ | — | b:1 | — | — | — | f:1 |

in which the occurrence colors correspond to columns and each entry is a function $g : C \to \mathbf{N}$, shown here using a simplified notation “$X : i$” for:

$$g(c) \;=\; \begin{cases} i, & \text{if } c = X, \\ 0, & \text{otherwise;} \end{cases}$$

It can be observed that, in addition to the representation of machines, there is a systematic structure of the net model shown in Fig.6:

- there is a place corresponding to each of the machines ($p_{11}$, $p_{22}$ and $p_{33}$ in Fig.4), and a single place corresponding to *Input* and *Output* ($p_{40}$ and $p_{04}$ in Fig.4); in general, for an $m$–machine cell, there are $m$+2 such places;

- places representing *Input* and all machines (i.e., $p_{04}$ and $p_{11}$, $p_{22}$, $p_{33}$) have three input transitions each (or $m$ input transitions in general) representing the possible moves from the ‘other’ machines and *Output* (and from *Input* in the case of $M_1$); the ‘other’ machines do not include the ‘next’ machine (*Output* is ‘the next machine’ for $M_3$); so, for $p_{04}$ the input arcs are from $M_2$, $M_3$ and *Output*, for $p_{11}$ the input arcs are from $M_3$, *Output* and *Input*, for $p_{22}$ the input arcs are from *Output*, $M_1$ (carrying a part) and $M_1$ (without a part), and for $p_{33}$ the input arcs are form $M_1$, $M_2$ (carrying a part) and $M_2$ (without a part);

- places representing *Output* and all machines have three output transitions each (or $m$ output transitions in general) representing the possible moves to ‘other’ machines and *Input* (and *Output* in the case of $M_3$); the ‘other’ machines do not include the ‘previous’ machine (*Input* is ‘the previous machine’ for $M_1$); so for $p_{11}$ the output arcs are to $M_2$ (carrying a part), $M_2$ (without a part) and $M_3$, for $p_{22}$ the output arcs are to $M_3$ (carrying a part), $M_3$ (without a part) and *Input*, for $p_{33}$ the output arcs are to *Output*, *Input* and $M_1$, and for $p_{40}$ the output arcs are to *Input*, $M_1$ and $M_2$;

- the place representing *Input* ($p_{04}$) has only one output transition ($t_{01}$, representing 'pick a part, move and load $M_1$');
- the place representing *Output* ($p_{40}$) has only one input transition ($t_{34}$, representing 'unload $M_3$, move and drop'),
- the total number of 'scheduling' transitions, resulting from the above rules, is equal to (counting either the input or output arcs) $1+m*(m+1)$, so for a three–machine cell there are 13 transitions modeling the possible robot schedules (see Fig.4); a net model of a four–machine cell needs 21 transitions (and 6 places) to represent all possible robot's schedules, and a model of a five–machine cell, needs 31 such transitions (and 7 places).

Place–invariants of the net shown in Fig.6 can be grouped in sections corresponding to different colors (i.e., different schedules). There are 8 invariants for color/schedule A, 5 invariants for color/schedule B, 6 invariants for color/schedule C, etc.; total number of place–invariants for this model is 33. As before, the minimum cycle time of each schedule is determined by the invariant subnet with the maximum total cycle time. Since all invariant subnets are simple cyclic nets, each subnet cycle time is equal to the sum of firing times assigned to all transitions of the (invariant) subnet. The cycle times of the six schedules are as follows (since the sets of transitions of some invariant subnets are subsets of those of other invariant subnets, not all invariants are used in the formulas):

| *schedule* | *cycle time* |
|---|---|
| A | $\tau_A = \tau_1$ |
| B | $\tau_B = \max(\tau_9, \tau_{10}, \tau_{12})$ |
| C | $\tau_C = \max(\tau_{14}, \tau_{16}, \tau_{17}, \tau_{19})$ |
| D | $\tau_C = \max(\tau_{20}, \tau_{21}, \tau_{22}, \tau_{23}, \tau_{24})$ |
| E | $\tau_E = \max(\tau_{25}, \tau_{26}, \tau_{28})$ |
| F | $\tau_E = \max(\tau_{30}, \tau_{31}, \tau_{32}, \tau_{33})$ |

where

$$\begin{aligned}
\tau_1 &= o_1 + o_2 + o_3 + u + 3v + 3w + x + 5y \\
\tau_9 &= o_1 + o_2 + u + 2v + 3w + 5y \\
\tau_{10} &= o_1 + u + 3v + 3w + x + 9y \\
\tau_{12} &= o_3 + v + w + 5y \\
\tau_{14} &= o_1 + o_2 + u + 2v + 2w + 5y \\
\tau_{16} &= o_2 + o_3 + 3v + 2w + x + 5y \\
\tau_{17} &= o_2 + u + 3v + 3w + x + 7y \\
\tau_{19} &= u + 3v + 3w + x + 10y \\
\tau_{20} &= o_1 + o_2 + o_3 + u + 3v + 3w + x + 5y \\
\tau_{21} &= o_1 + u + 2v + 2w + x + 5y \\
\tau_{22} &= o_2 + 2v + 2w + 4y
\end{aligned}$$

$$\begin{aligned}
\tau_{23} &= o_3 + u + 2v + 2w + x + 5y \\
\tau_{24} &= u + 2v + 2w + x + 8y \\
\tau_{25} &= o_1 + u + v + 2w + 4y \\
\tau_{26} &= o_2 + o_3 + 3v + 2w + x + 5y \\
\tau_{28} &= o_3 + u + 3v + 3w + x + 9y \\
\tau_{30} &= o_1 + u + v + 2w + 4y \\
\tau_{31} &= o_2 + 2v + 2w + 4y \\
\tau_{32} &= o_3 + 2v + w + x + 4y \\
\tau_{33} &= u + 3v + 3w + x + 12y
\end{aligned}$$

Because the optimal schedule is the schedule with the minimum cycle time, so:

$$\tau_{opt} \;=\; \min(\tau_A, \tau_B, \tau_C, \tau_D, \tau_E, \tau_F).$$

Since all simple schedules are modeled by the same (colored) net (as shown in Fig.6), the same modeling net will also represent the composite schedules of a manufacturing cell. For example, to model a composite schedule A+B obtained by combining schedules A and B, two additional basic colors *Ab*, *aB* and an auxiliary color *ab* should be introduced with the following occurrences of transitions (the occurrences of *Ab* correspond to part A of the schedule AB, while occurrences *aB* to part *B* of this schedule):

| *arc* | ... | Ab | aB |
|---|---|---|---|
| $(t_{12}, p_{22})$ | ... | aB:1 | ab:1 |
| $(t_{23}, p_{33})$ | ... | Ab:1 | ab:1 |
| $(t_{34}, p_{40})$ | ... | Ab:1 | ab:1 |
| $(p_{22}, t_{22})$ | ... | — | ab:1 |
| $(p_{33}, t_{30})$ | ... | — | ab:1 |
| $(p_{40}, t_{42})$ | ... | ab:1 | — |

so that the complete robot's path is ($p_{04}$, $t_{01}$, $p_{11}$, $t_{12}$, $p_{22}$, $t_{22}$, $p_{33}$, $t_{34}$, $p_{40}$, $t_{42}$, $p_{22}$, $t_{23}$, $p_{33}$, $t_{30}$, $p_{04}$, $t_{01}$, $p_{11}$, $t_{12}$, $p_{22}$, $t_{23}$, $p_{33}$, $t_{34}$, $p_{40}$, $t_{40}$, $p_{04}$). Systematic generation of such composite schedules and their analysis need to be investigated in greater detail.

## 5. Concluding Remarks

A systematic approach to modeling and analysis of simple and composite schedules for a large class of manufacturing cells is proposed and is illustrated by example schedules for a 3–machine cell. The derived net models are conflict-free, composed of a relatively small number of subnets, which can easily be analyzed by net invariants [ZK93].

Invariant analysis of net models provides the performance characteristics (the throughput or the average cycle time) in symbolic form which means that specific values of performances can easily be obtained by evaluating the symbolic results for specific values of parameters (i.e., symbols).

A number of simplifying assumptions were made during the derivation of Petri net models, e.g., the all parts are identical, that the robot travel times between adjacent machines are the same, etc. It should be noted that all these assumptions were made to simplify the discussion and they can easily be removed by simple modifications of the presented approach. In particular, composite schedules can be used to describe scheduling problems when several different parts enter and leave the cell in one cycle. A decomposition of such a schedule into a number of simple components identifies operations performed on parts of different types.

The number of schedules (both simple and composite) increases very quickly with the number of machines, and the number of composite schedules also increases rather quickly with the the length of the schedule; for a 3–machine cell, there are 6 simple schedules, 34 different 2–schedules and 198 different 3–schedules. Instead of analyzing all these schedules one after another, a more general approach can be developed, using colored Petri nets for modeling the whole sets of schedules, with different colors representing different schedules. Most likely new methods of analyzing such models need to be developed to reduce, as soon as possible, all those cases which cannot affect the final solution.

## Acknowledgement

Collaboration with Dr. W. Kubiak of the School of Business Administration, Memorial University of Newfoundland, is gratefully acknowledged.

The Natural Sciences and Engineering Research Council of Canada partially supported this research through Research Grant A8222.

## References

[Cl83] Claybourne, B.H.: "Scheduling robots in flexible manufacturing cells"; CME Automation, vol.30, no.5, pp.36–40, 1983.

[DH90] Dixon, C., Hill, S.D.: "Work–cell cycle–time analysis in a flexible manufacturing system"; Proc. Pacific Conf. on Manufacturing, Sydney–Melbourne, Australia, vol.1, pp.182–189, 1990.

[Hi89] 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.

[Je87] K. Jensen, "Coloured Petri nets"; in: "Advanced Course on Petri Nets 1986" (Lecture Notes in Computer Science 254), G. Rozenberg (ed.), pp.248–299, Springer Verlag 1987.

[Mu89] Murata, T.: "Petri nets: properties, analysis and applications"; Proceedings of IEEE, vol.77, no.4, pp.541–580, 1989.

[Re85] Reisig, W.: "Petri nets - an introduction" (EATCS Monographs on Theoretical Computer Science 4); Springer Verlag 1985.

[S3BK92] 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.

[Zu90] Zuberek, W.M., "Performance evaluation using timed colored Petri nets", Proc. 33-rd Midwest Symp. on Circuit and Systems (Special Session on Petri Net Models), Calgary, Alberta, pp.779-782, 1990.

[Zu91] Zuberek, W.M.: "Timed Petri nets – definitions, properties and applications"; Microelectronics and Reliability (Special Issue on Petri Nets and Related Graph Models), vol.31, no.4, pp.627–644, 1991.

[ZK93] Zuberek, W.M., Kubiak, W., "Timed Petri net models of flexible manufacturing cells"; Proc. 36-th Midwest Symp. on Circuits and Systems, Detroit MI, August 16–18, 1993.