COCURRENT SYSTEM MODELLING
BY CONDITIONAL PETRI NETS

by

Z. Barzilai, E. Strasbourger and M. Yoeli

Technical Report #124
March 1978
ABSTRACT

A tool has been defined to model the behavior of concurrent indeterminate systems. This tool, the Conditional Petri Net, combines the capabilities of Petri Nets with those of organiphase. Conditional Petri Nets model the input-output sequence of concurrent indeterminate systems, where transition fiability may be dependent on other than the input places. Models for two dynamic systems (a bank teller system and an elevator system) are presented, which illustrate the power and applicability of Conditional Petri Nets.

Keywords: modelling, behavior specification, Petri Nets, discrete dynamic systems, concurrent systems, indeterminate systems.
1. INTRODUCTION

A familiar method of modelling the behavior of discrete systems is finite automata (sequential machines). However, the sequential machine model has major drawbacks in describing systems with a high degree of concurrency and with non-deterministic features.

Conventional Petri nets suitably model concurrency and indeterminism, however they lack facilities to describe dynamic input-output relations.

R. Valette [1], [2] extends the Petri net concept to provide for the description of input-output relations of dynamic systems. Dumas and Prunet [3] introduce the concept of organiphase as a modification of Petri nets. This model provides convenient means to represent "conditioning", i.e. the dependence of transition firability on other than the input places. On the other hand their model does not allow indeterminism.

However, we consider the power of Petri nets to describe conflicts with indeterminate outcomes to be essential for specifying the desired behavior of highly concurrent systems (e.g. non-priority arbiters) without introducing unnecessary constraints.

In this paper we introduce the model of Conditional Petri Net, which combines the capabilities of the organiphase with the possibility of modelling conflicts.

We also illustrate the applicability of our model to a modified version of the elevator system described in [3].
2. BASIC CONCEPTS

A Mixed Petri Graph (MPG) is defined as $G = \langle P, T, R, B \rangle$ where:

- $P$ is a finite set of places
- $B \subseteq P$ is the set of Boolean places ($P - B$ is the set of Integer places)
- $T$ is a finite set of transitions ($T \cap P = \emptyset$)
- $R$ is a binary relation on $P \times T$ satisfying the condition $R \subseteq (P \times T) \cup (T \times P)$.

A marking $m$ of $G$ is a function $m: P \rightarrow \mathbb{N}$ (where $\mathbb{N}$ is the set of non-negative integers) satisfying the condition

$$\{m(p) | p \in B\} \subseteq \{0, 1\}.$$  

We denote by $M_G$ the set of all markings of $G$. In [4] the concept of Mixed Petri Net (MPN) was introduced as an ordered pair $(G, m)$ where $G$ is an MPG and $m \in M_G$.

In this paper we extend the concept of MPN to a Conditional Petri Net (CPN) $\bar{N} = \langle G, I, O, f, g, h, d \rangle$ where $G$ is defined as above and

- $I = \{0, 1\}^k$, $k \geq 0$, is the set of input signal levels
- $O$ is the set of output pulses
- $f$ is the input function, $f: T \rightarrow 2^I$
- $g$ is the output function, $g: T \rightarrow O \cup \{\lambda\}$
- $h$ is the condition function, $h: t \mapsto c_t (t \in T)$
  
  where $c_t$ is a function $c_t: M_G \times I \rightarrow \{\text{true, false}\}$,
- $d$ is the delay function, $d: T \rightarrow \mathbb{N}$.

CPNs are conveniently represented as diagrams where:

- a Boolean place $p \in B$ is shown as a circle
- an integer place $p \in P - B$ is shown as a double circle
A nondelayed transition \( t \in T \) such that \( d(t) = 0 \) is shown as a bar. A delayed transition \( t \in T \) such that \( d(t) > 0 \) is shown as a double bar. The condition \( c_t \) and output \( g(t) \) are shown in square brackets beside the corresponding transition bar for \( t \).

A directed edge from a place \( p \) to a transition \( t \) indicates that \( p \rightarrow t \), i.e. \((p,t) \in R\), and a directed edge from \( t \in T \) to \( p \in P \) indicates that \( t \rightarrow p \). The marking \( m \) is shown by placing \( m(p) \) dots (tokens) into the circle (or double circle) representing \( p \). Alternatively, the integer \( m(p) \) may be written into the double circle representing the integer-type place \( p \).

The place \( p \) is an input place of the transition \( t \) iff \( p \rightarrow t \) and an output place of \( t \), iff \( t \rightarrow p \).

A transition \( t \in T \) of a Conditional Petri Net \( \tilde{N} \) is firable under the marking \( m \in M_{\tilde{G}} \) and input \( i \in I \), iff (1) \( c_t(m,i) = \text{true} \) and (2) \( m(p) > 0 \) for every input place \( p \) of \( t \).

If \( t \) is firable, the firing of \( t \) consists of decreasing the marking \( m(p) \) of its input places \( p \) by 1, and replacing the marking \( m(q) \) of its output places \( q \) by \( m(q) + 1 \) where + denotes Boolean addition if \( q \in B \) and integer addition if \( q \in P-B \).

If \( t \in T \) is firable under \( m \in M_{\tilde{G}} \) and \( i \in I \), and the marking \( m' \) is obtained by firing \( t \), we write \((m,i) \xrightarrow{t} m'\).

Rather than giving a formal definition of the dynamic behavior of a CPN, we provide the following informal explanation.

A CPN is intended to represent a discrete dynamic system with \( M_{\tilde{G}}, I, O \) as state set, input set, and output set, respectively. Let \((m,i) \xrightarrow{t} m'\),
\( d(t) = 0, g(t) = 0 \), and assume that the system is in state \( m \) and the present input is \( i \). The system \textit{may} enter state \( m' \) and simultaneously produce output 0 provided the above firability condition (2) for \( t \) has persisted for \( \delta \) time units.

We now illustrate the concepts of MPN and CPN by suitable examples.

3. EXAMPLE OF MPN

An example of an MPN is shown in Figure 1. This MPN may be viewed as a model of a multi-customer, two-teller bank system. A particular marking

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{mpn_example.png}
\caption{Discrete System represented by MPN.}
\end{figure}
represents a particular state of the system. With reference to Fig. 1, the Boolean-type places represent conditions as specified below (i ∈ \{1,2\}): 

- **p_1^i** - Teller i is free
- **p_2^i** - A customer is being served by teller i.

The marking m of the integer-type places may be interpreted as follows:

- **m(p_0^i)** = # of customers requesting service
- **m(p_1^i)** = # of customers who have been served by teller i.

The transitions in this example may be interpreted as the following events:

- **t_1^i** - The teller i starts serving a customer
- **t_2^i** - The teller i finishes servicing.

4. EXAMPLE OF A CONDITIONAL PETRI NET (CPN)

Now consider the desirability of the number of tellers being dependent on the number of customers. If there are few customers (e.g. less than 4) then one teller could suffice freeing a bank employee for other tasks. If the number of waiting customers grows too large (e.g. more than 4) then a second teller could become active. This situation is conveniently represented by a CPN such as in Figure 2. The additional place **p_4^i** represents the condition of no second teller active. Additional transition **t_3^i** represents retiring of the second teller to other bank tasks. Additional transition **t_4^i** represents activating a second teller upon large customer queue. The input **i** represents a bank employee available to become a second teller. The output **0_1** represents the appearance of the CLOSED sign, and **0_2** its removal.
In [4] the concept of Extended Petri Net was introduced and its power to model dynamic systems was demonstrated. The model introduced in this paper combines the advantages of the Extended Petri Net with the possibility of representing "conditioning".

5. A MORE COMPREHENSIVE EXAMPLE OF A CPN

5.1 Problem Definition

Consider an elevator system (see Figure 3) consisting of an elevator operating between 3 floors. This elevator must respond to requests from any of the 3 floors where two buttons (U and D) are available to summon the elevator for going up or down, respectively. The elevator must also respond to internal requests of the 3 A-buttons within the elevator.
requesting movement to one or more of the 3 floors.

Figure 3: Elevator System

The following rules also apply:

A) The elevator may move only when its door is closed
   (Elevator doors are manually opened and closed.)

B) If the elevator is servicing requests and is directed simultaneously
   upwards and downwards, then it will travel in the same direction as
   its last previous motion.

C) If the elevator has no requests and at that time is directed simulta­
   neously upwards and downwards, then it is not important which request
   is satisfied first.

D) When the elevator reaches a requested floor it will delay there a
   prescribed minimum period of time $\delta$ before moving again even if doors
   are not opened and there is a request elsewhere.
5.2 System Control

The elevator system control is shown in Figure 4. Input signal levels are shown as open arrows $\rightarrow$ while output pulses are shown as closed arrows $\circlearrowleft$.

![Figure 4: Elevator System Control]

A group of similar input or output lines such as

\[ l_1 \xrightarrow{i=1} \]
\[ l_2 \xrightarrow{i=1} \]
\[ l_3 \xrightarrow{i=1} \]

will be shown as

\[ l_i \xrightarrow{i=1} 3 \]

Input signals are:

- $C_i$ indicates that the elevator is on the $i$-th floor.
- $A_i$ indicates that a button for the $i$-th floor has been depressed within the elevator.
0 indicates that the elevator door is open
\(D_i\) indicates that the Down button has been depressed on the \(i\)-th floor
\(U_i\) indicates that the Up button has been depressed on the \(i\)-th floor.

Output pulses are:

- **HALT** causes the elevator motor to stop.
- **LOWER** causes the elevator to start moving down
- **RAISE** causes the elevator to start moving up
- **RESET \(D_i\)** causes button \(D_i\) to be reset and available for another request
- **RESET \(U_i\)** causes button \(U_i\) to be reset and available for another request
- **RESET \(A_i\)** causes button \(A_i\) to be reset and available for another request.

### 5.3 System Behavior

A CPN describing the behavior of the elevator system (Figures 5, 6, 7 and 8) consists of 10 disjoint graphs with their labelings and markings. The elevator behavior is represented in Figure 5 and the request behavior for each of the three floors is represented in Figures 6, 7 and 8. There are three copies of the graphs of Figures 6, 7 and 8, one for each of the \(i\) floors \(i = 1, 2, 3\).

The elevator may be in one of seven activities or states, represented by a token in the correspondingly labeled place in Figure 5.

- **UP** going up
- **DOWN** going down
- **STOPUP** stopped with priority to continue up
- **STOPDOWN** stopped with priority to continue down
\[ h_1 = (C_1 \land (M_2 \lor M_3) \lor (C_2 \land M_3)) \land \overline{\delta} \]

\[ h_2 = (C_1 \land H_2) \land (C_2 \land M_2) \]

\[ h_3 = C_2 \land (M_2 \land M_2 \lor H_3) \lor C_3 \]

\[ h_4 = [(C_3 \land (H_1 \lor H_2)) \lor (C_2 \land M_1 \land H_3)] \land \overline{\delta} \]

\[ h_5 = (\overline{H}_1 \land \overline{H}_2 \land \overline{H}_3) \]

\[ h_6 = [C_3 \land (M_3 \land M_2) \lor (C_2 \land M_3)] \land \overline{\delta} \]

\[ h_7 = (C_3 \land \overline{M}_2) \lor (C_2 \land \overline{M}_2) \]

\[ h_8 = C_2 \land (\overline{M}_2 \land M_2) \lor (\overline{H}_1) \lor C_1 \]

\[ h_9 = [C_1 \land (M_2 \land M_3) \lor (C_2 \land M_3 \land H_1)] \overline{\delta} \]

where:

\[ M_1 = M_1 \lor M_2 \]

\[ M_2 = M_2 \lor M_2 \land M_2 \]

\[ M_3 = M_2 \land M_2 \]

**Figure 5:** Elevator Behavior
Figure 6: Internal Request Behavior

- $h_{10}^i = c_i \land (\text{STOPUP} \lor \text{STOPDOWN})$
- $h_{11}^i = a_i$

Figure 7: Up Request Behavior

- $h_{12}^i = c_i \land \text{STARTUP}$
- $h_{13}^i = u_i$

Figure 8: Down Request Behavior

- $h_{14}^i = c_i \land \text{STARTDOWN}$
- $h_{15}^i = d_i$
STOP stopped with no preferential direction
STARTDOWN stopped preparing to go down
STARTUP stopped preparing to go up

Each of the three buttons $A_i$ in the elevator may be "activated" (represented by a token in place $MA_i$ of the appropriate copy of the graph in Figure 6) or may be not activated (represented by a token in place $\overline{MA_i}$). Likewise each of the three up buttons $U_i$ and down buttons $D_i$ may be activated (represented by a token in place $MU_i$ or $MD_i$ of the appropriate copy of the graph in Figure 7 or 8) or may be not activated (represented by a token in place $\overline{MU_i}$ or $\overline{MD_i}$).

A transition from a no request place $\overline{MA}_i$ (or $\overline{MU}_i$ or $\overline{MD}_i$) to request place $MA_i$ (or $MU_i$ or $MD_i$) occurs when button $A_i$ (or $U_i$ or $D_i$) is depressed. A transition from $MA_i$ to $MA_i$ occurs when the elevator stops at floor $i$, i.e. $(C_i \land (STOPUP \lor STOPDOWN)) = true$, where if $p$ is a Boolean place then $\mathcal{p} = m(p) = 1$ and if $x$ is a binary input then $\mathcal{x} = x = 1$.

A transition from $MU_i$ to $\overline{MU_i}$ occurs when the elevator starts up after being stopped at floor $i$, i.e. $(C_i \land STARTUP) = true$. Likewise a transition from $MD_i$ to $\overline{MD}_i$ occurs when the elevator starts down after being stopped at floor $i$, i.e. $(C_i \land STARTDOWN) = true$.

Twelve transitions may occur between the seven elevator states:

1. If the elevator is in state STOPUP and a request exists on a higher floor $((C_i \land (M_2 \lor M_3)) \lor (C_2 \land M_3))$ (where $M_i$ is defined as $\overline{MA_i} \lor \overline{MU_i} \lor \overline{MD_i}$) and the door is closed ($\overline{D}$) and the elevator has been stopped a delay period of time to allow the door to be opened, then a transition occurs to state STARTUP. State STARTUP allows an up request to be cancelled from the current floor of the elevator ($MU_i \rightarrow \overline{MU}_i$ transition).
2. If the elevator is in state STARTUP and all requests up from the current floor of the elevator have been cancelled \[((C_1 \land MU_1) \lor (C_2 \land MU_2))\] then a transition occurs to state UP and a RAISE output pulse is given to the elevator motor. While the elevator is in state UP, it is moving upwards.

3. If the elevator is in state UP then either of the following conditions cause a transition to state STOPUP and a HALT output pulse to the elevator motor:
   a) The elevator arrives at the 2nd floor \((C_2)\) where one of the following conditions occurs:
      i) there is a request to exit from the elevator \((MA_2)\)
      ii) there is a request from the floor to go up from there \((MU_2)\)
      iii) there is no request to or from a higher floor \((M_3)\)
   b) the elevator arrives at the highest floor \((C_3)\)

4. If the elevator is in state STOPUP and there is no request to or from a higher floor but there is a request to or from lower floor \(((C_2 \land M_1 \land \neg M_3) \lor (C_3 \land (M_1 \lor M_2)))\) and an appropriate delay has passed to allow opening elevator door, then a transition occurs to state STARTDOWN.

5. If the elevator is in state STOPUP and there is no request at all, \((\neg M_1 \land \neg M_2 \land \neg M_3)\), then after appropriate door opening delay there is a transition to place STOP.

6. If the elevator is in state STOP, then the presence of a request to or from a higher floor will cause a transition to state STARTUP. Or a request to or from a lower floor will cause a transition to state STARTDOWN. If conflicting requests occur simultaneously, then the elevator behavior is indeterminate.
7. The other five transitions are similar except that the movement priority is downward rather than upward.

Initial conditions have the elevator stopped (token in place STOP) and no requests (tokens in the nine places $\overline{MA_1}$, $\overline{MU_1}$, and $\overline{MD_1}$).

5.4 Sample Behavior Response

Floors

<table>
<thead>
<tr>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\dowarrow$</td>
<td>$\uparrow$</td>
<td>$\dowarrow$</td>
</tr>
<tr>
<td>$\dowarrow$</td>
<td>$\dowarrow$</td>
<td>$\dowarrow$</td>
</tr>
<tr>
<td>$\dowarrow$</td>
<td>$\uparrow$</td>
<td>$\dowarrow$</td>
</tr>
</tbody>
</table>

Legend: $\bigcirc$ indicates position of elevator
$\dowarrow$ up-request
$\dowarrow$ down-request
$\uparrow$ $A_1$-request

Starting from the initial conditions of the elevator stopped (token in place STOP) at floor 2, the following sequence of actions and responses could occur:

a) Action: There is a simultaneous request from floor 3 ($D_3$) and from floor 1 ($U_1$) causing transition $\overline{MD_3} \rightarrow \overline{MD_3}$ and transition $\overline{MU_1} \rightarrow \overline{MU_1}$.

Response: In this conflict situation, conditions for both transition $\text{STOP} \rightarrow \text{STARTUP}$ and transition $\text{STOP} \rightarrow \text{STARTDOWN}$ are satisfied. Assume that the latter transition fires...
and then transition \( \text{STARTDOWN} \rightarrow \text{DOWN} \) which outputs a \( \text{LOWER} \) pulse to the motor. Upon reaching floor 1, condition for transition \( \text{DOWN} \rightarrow \text{STOPDOWN} \) is satisfied and the elevator is HALTed.

b) Action: Man enters elevator and presses button \( A_2 \) causing transition \( \overline{\text{MA}}_2 \rightarrow \text{MA}_2 \).
Response: Transition \( \text{STOPDOWN} \rightarrow \text{STARTUP} \) then fires followed by transition \( \overline{\text{MU}}_1 \rightarrow \text{MU}_1 \), and then transition \( \text{STARTUP} \rightarrow \text{UP} \) which causes the elevator to rise.

c) Action: Man on floor 2 requests to go down \( D_2 \).
Response: Transition \( \overline{\text{MD}}_2 \rightarrow \text{MD}_2 \) fires. Upon arrival at floor 2, condition for transition \( \text{UP} \rightarrow \text{STOPUP} \) is satisfied (due to presence of \( \text{MA}_2 \); note that \( \text{MD}_2 \) would not satisfy the condition in the presence of \( \text{M}_3 \)) and fires (HALTing the elevator), followed by transition \( \overline{\text{MA}}_2 \rightarrow \overline{\text{MA}}_2 \).

d) Action: Man leaves the elevator.
Response: because of up priority condition only transition \( \text{STOPUP} \rightarrow \text{STARTUP} \) may fire (from \( \text{MD}_3 \) ignoring \( \text{MD}_2 \)), followed by transition \( \text{STARTUP} \rightarrow \text{UP} \) raising the elevator to floor 3. At floor 3 transition \( \text{UP} \rightarrow \text{STOPUP} \) fires HALTing the elevator.

e) Action: Man enters elevator and presses button \( A_1 \), causing transition \( \overline{\text{MA}}_1 \rightarrow \text{MA}_1 \).
Response: Transition \( \text{STOPUP} \rightarrow \text{STARTDOWN} \) fires followed by transition \( \overline{\text{MD}}_3 \rightarrow \text{MD}_3 \) followed by transition \( \text{STARTDOWN} \rightarrow \text{DOWN} \), LOWERing the elevator to floor 2 where transition \( \text{DOWN} \rightarrow \text{STOPDOWN} \) fires HALTing the elevator.
f) *Action:* Man enters the elevator (if not already in).

*Response:* Transition STOPDOWN → STARTDOWN fires followed by

MD₂ → MD₂ followed by STARTDOWN → DOWN, LOWERing the elevator to

floor 1 where transition DOWN → STOPDOWN fires HALTing the elevator,

followed by transition MA₁ → MA₁.

g) *Action:* Men leave the elevator. No additional request.

*Response:* Transition STOPDOWN → STOP fires leaving the elevator

system in an initial condition.

6. CONCLUSIONS

We have introduced a powerful extension of the Petri net concept which provides for:

1) description of input-output behavior of discrete dynamic systems with

   high degrees of concurrency;

2) conditional and delayed firability of transitions. This feature is

   particularly suitable to describe complex systems in a conveniently

   decomposed form.

3) Modelling of behavioral conflicts.
REFERENCES


