JUNE 1984 LIDS-P-1386

PETRI NET REPRESENTATION OF DECISION MODELS*

by

Daniel Tabak†

Alexander H. Levis††

ABSTRACT

Models of decisionmaking organizations supported by command, control and communication systems, are represented using the Petri Net formalism. A small set of primitives, defining the correspondence between decision models signals and functions and their Petri Net counterparts, is proposed. A new decision signal routing demultiplexer is added to the Petri Net formalism to represent internal decisionmaking in the model. Using the above primitives, any decisionmaking structure can be modeled by a Petri Net diagram. An array is introduced that describes the interactions between decisionmakers and an algorithm is presented for the calculation of delay when synchronous protocols are used.

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support by the Office of Naval Research, Contract No. N00014-83-K-0185 and the Air Force Office of Scientific Research, Contract No. AFOSR-80-0229.

†School of Engineering, Boston University, Boston, MA 02115

††Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.

1. INTRODUCTION

Petri Nets [1], [2] have been extensively used in the representation and analysis of computing systems and processes. The Petri Net formalism is suitable for representing dynamic processes; particularly when some of the events may occur concurrently. Recently, the use of Petri Nets in the modeling of decisionmaking processes has been proposed [3], [4].

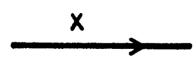
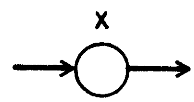
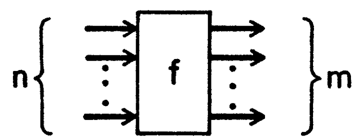
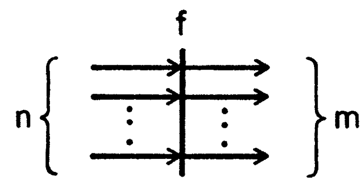
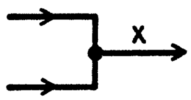
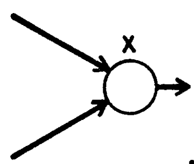
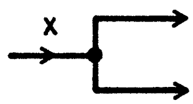
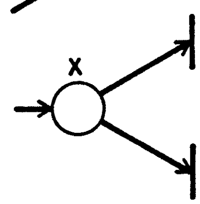
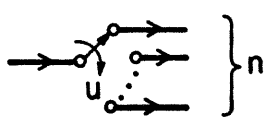
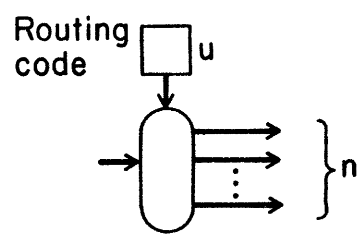
This paper describes the formalism of the representation of decisionmaking models by Petri Nets, by introducing a small set of system primitives and their corresponding Petri Net elements. Using this small number of primitives one can convert any previously used decisionmaking model into an equivalent Petri Net. The models used in [3], [4] include an internal decisionmaking process, represented by a switch that routes signals along alternative directions. Such a switch does not have a counterpart in the previously published Petri Net formalism [1]. In order to overcome this problem, a new demultiplexer element is introduced into the set of Petri Net primitives, to be used in conjunction with the decisionmaking model.

The primitives of the Petri Net representation of elements, appearing in decisionmaking models, are described in Section 2. An example of an equivalent Petri Net representation of a decisionmaking model is given in Section 3. In Section 4, an array representation of the decisionmaking organization is introduced that is based on the Petri Net description. In Section 5, an algorithm for the computation of delays is introduced and is then applied to two three-person organizations. Conclusions and directions for research are presented in Section 6.

2. THE SYSTEM PRIMITIVES

The representation of the decisionmaking system primitives by their Petri Net equivalents is shown in Table 1. These primitives will be discussed in the following paragraphs, in the same order as listed in Table 1.

TABLE 1. Primitives for the Petri Net Representation of Decision Models

Decision Model Primitives		Petri Net Representation	
Name	**Symbol**	**Name**	**Symbol**
(a) Signal		Circle node; place	
(b) Function		Bar node; transition	
(c) Signal convergence		Multiple input place	
(d) Signal divergence		Multiple output place	
(e) Decision switch		Demultiplexer	

(a) Signal. A signal transmitted within a decisionmaking system, or between such systems, represents a message, containing information. The information may be represented in a variety of forms, however, this is not an issue to be considered in this paper. The equivalent Petri Net representation of any signal in the decisionmaking system is a circle node, alternately denoted as a place [1]. An empty place: ○, represents the existence of a definite path, a medium for the signal (message) to be contained and subsequently transmitted by the place. The actual presence or availability of an information message in the place is denoted by replacing a token within it: ⦿.

Thus, a symbol ○ means that the signal (message) y may (can) appear at the indicated place. A symbol ⦿ means that the signal y is actually stored at the indicated place.

(b) Function. Any transformation, performed on a signal or message is considered as a function. In particular, a function may be just a simple addition of two signals or a complicated decision process, based on the information supplied by the input signals. In general, a function primitive can have n inputs and m outputs, as shown in Table 1, entry (b). The Petri Net element, corresponding to a function, is the bar node or the transmission. It represents within the Petri Net formalism any operation, process, or function available within the system under consideration.

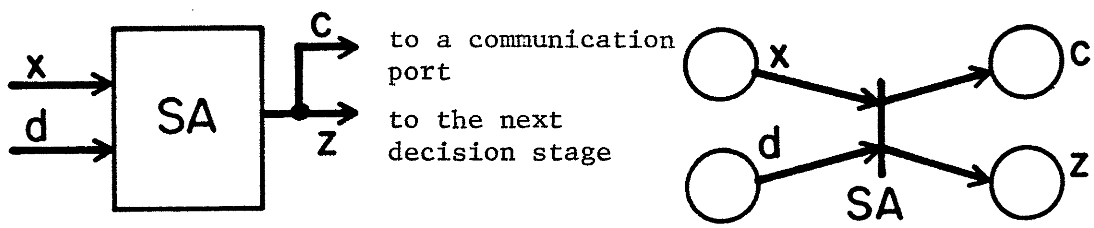
Consider a subtractor (Sb), performing the operation (x-y) on two signals x and y, Figure 1(a). Its equivalent Petri Net representation is shown in Figure 1(b). Note that the two input signals x and y, and the output difference signal z, are represented by circle nodes, or places.

A Situation Assessment (SA) unit within a Decisionmaker (DM) model [4] is shown in Figure 2. It has two inputs: x from the outside

(a) Subtractor

(b) Equivalent Petri Net representation.

Figure 1. Petri Net Representation of a Subtractor



(a) Situation Assessment Unit

(b) Equivalent Petri Net representation. (n=m=2)

Figure 2. Petri Net Representation of a Situation Assessment Unit

(a)

(b)

Figure 3. Petri Representation of Two-Signal Convergence

and d from a local memory [5]. It has two outputs (in this example they are equal): output z goes to the next decision stage and output c, intended for external communication.

(c) Signal Convergence. In many decision processes a signal may be formed out of a variety of sources. In this case we have a convergence of several alternative signal paths into a single node, as shown in Table 1, entry (c). The equivalent Petri Net representation is a circle node or place with multiple inputs and a single output. Although only two inputs are shown, the same representation can be applied in cases with more than two inputs.

A signal z can be formed either as an output of function f_1 or of f_2 (the two functions never operate simultaneously). The block diagram representation and the Petri Net equivalent of such an arrangement, is shown in Figure 3.

(d) Signal Divergence. A signal transmitted along a single line, is transmitted along several liners (fan-out), starting at a given point. Such a case and its Petri Net representation are shown in Table 1, entry (d). The above lines must terminate at other transitions.

(e) Decision Switch. In some decisionmaking systems the information flow may be routed through a set of alternative paths. Such a routing is represented by a decision switch, shown in Table 1, entry (e) on the left. A signal (message) arrives through a single transmission path. It has to pass through an n-position switch, which would route the signal through any of the available n output paths, according to the position of the switch. The position of the switch is established by a decision input u. The Petri Net equivalent of such a switch is shown in Table 1(e), on the right. It is called a Demultiplexer, since according to its rules of operation, it functions as a logic device, denoted by the term demultiplexer [6], [7]. Its input is a single signal, arriving through a single path. This signal can be transmitted through only one of the n available output paths. The output path chosen depends on the decision u, expressed in this case as a binary code, which needs to have $\log_2 n$ bits. For instance, if n is 2, u has 1 bit, if n is 4, u has 2 bits, if n is 8, u has 3 bits, and so on. The decision code u can either be generated internally by the decisionmaker system, taking the code from an internal memory, or it can constitute the result of a functional operation of the decisionmaking system.

The decision subsystem (demultiplexer) is analogous to a decoder circuit [6], [7] shown in Figure 4. An output signal can appear on only one of the n output lines. The input code (binary), coming in on $\log_2 n$ lines, establishes on which of the n output lines the output signal will appear. Thus, the input code is analogous to the decision u. The decoding circuit will function only if there is an input signal along the ENABLE line. The ENABLE line is therefore analogous to the single input of the decision switch.

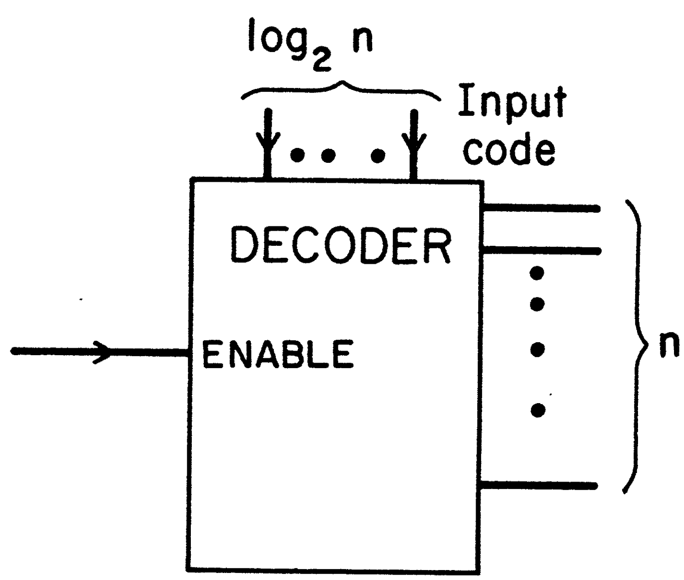


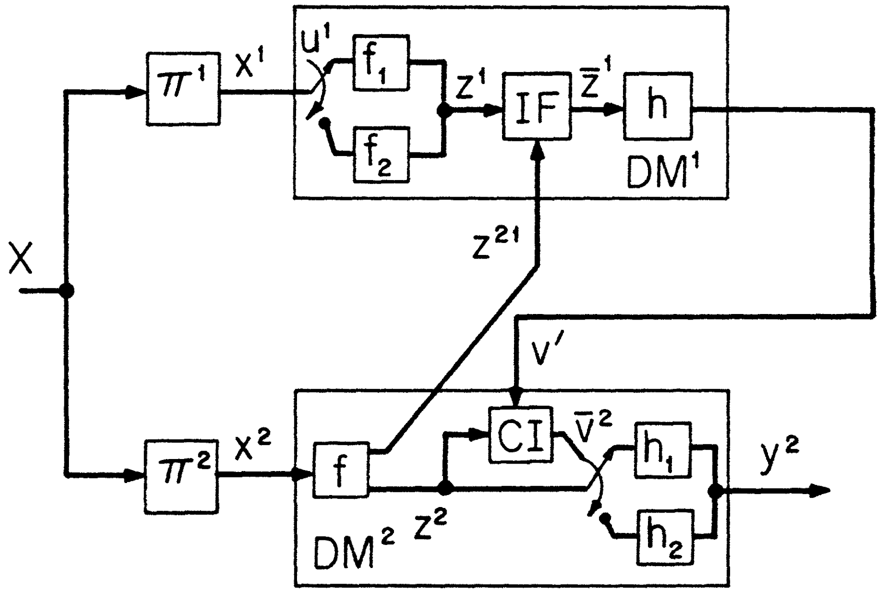
Figure 4. Decoder

3. A DECISIONMAKING ORGANIZATION EXAMPLE

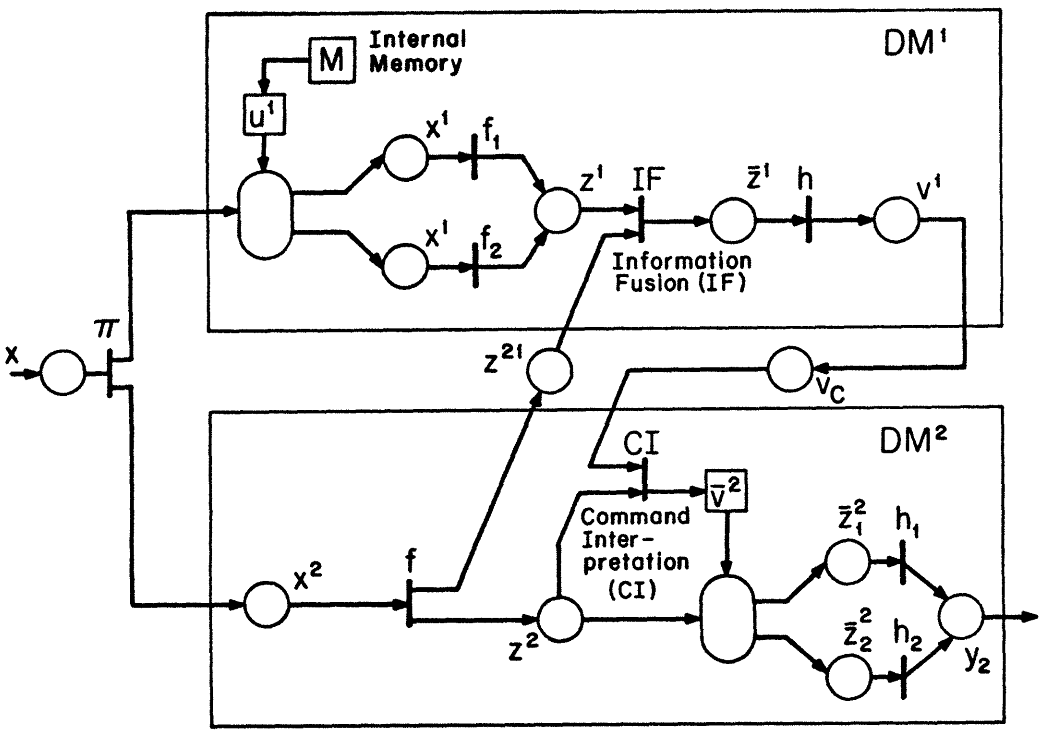
An example, showing a two-person organization, taken from [4], and its Petri Net equivalent, is shown in Figure 5(a), (b). The Petri Net representation has been assembled, step by step, using the equivalence of primitives established in Section 2.

The sequence of events, and the associated protocols, can be inferred directly from the Petri Net representation. Signal x is received concurrently by function π which generates signals x^1 and x^2, respectively. The decision switch in DM^1 determines the path x_1' will take. The selection is made according to the decision rule u which depends on data in the internal memory M. The signal x_1' is transformed into the signal z^1 either by function f_1 or f_2. However, no further activity can take place until signal z^{21} arrives at the Information Fusion (IF) function. The transition symbol with two inputs implies that both inputs must be present before the transition can occur. Thus, the protocol that requires information to be fused prior to a response being selected is made explicit by the Petri Net representation. In DM^2, the signal x^2 is processed by function f and then transmitted to DM^1 and z^{21}, and to the decision switch in DM^2. The Command Interpretation (CI) function cannot be executed until the signal v_C is received from DM^1. The inference is that DM^2 cannot act until he receives instructions or commands from DM^1. This is another explicit statement of the protocols that specify the interactions between organization members. While both representations depict the flows of information, only the Petri Net one, Figure 5(b), indicates which operations can be concurrent, which ones are sequential, and when coordination is necessary.

It should be noted that the z^{21} place and the v_C place (between place v^1 in DM^1 and CI in DM^2) are redundant from the standpoint of the Petri Net formalism. They serve to indicate the existence of communication links between the two decisionmakers. In future work, they will be used to model the properties of these communication links.



(a) Block diagram representation of a two person organization



(b) The equivalent Petri Net diagram

Figure 5. An Example of a Decisionmaking System

4. ARRAY REPRESENTATION OF DECISIONMAKING MODELS

The purpose of the array representation, described in this section, is to permit and efficient (eventually computerized) calculation of delays in the transmission of messages in a system consisting of interconnected Decisionmakers (DMs). Therefore, each DM, say DM^i, is represented by a sequence of delays. Each delay represents the time it takes for a message to be processed by a given subsystem of the DM. In other words, a delay is the time interval starting with the appearance of the message at the input of the subsystem, and ending with the appearance of the message at the subsystem's output. Each such subsystem is represented by a transition in the Petri Net representation of the DM.

The sequence for the delays within a DM is labeled in an ascending order, begining at the input. Thus, if DM^i has n transitions, his delay sequence will be:

$$\tau_{i1}\ \tau_{i2}\ \cdots\ \tau_{i(n-1)}\tau_{in}$$

where τ_{ij} is the delay of the j-th transition in DM^i.

In the proposed representation, n is chosen as a maximal number of transitions present in any DM. If another decisionmaker, DM^s, has fewer than n transitions, say $k < n$, his last (n-k) positions in he delay sequence are filled with zero delays:

$$\tau_{s1}\ \tau_{s2}\ \cdots\ \tau_{sk}\quad \underbrace{0\ \ldots\ 0}_{(n-k)\text{ times}}$$

The proposed representation is structured as a three-dimensional array $A_s(i,j,k)$, whose dimension indices represent:

i - the DM, $i = 1,2,\ldots,m$

j - the transition within the DM, $j = 1,2,\ldots,n$

k - the position within the information vector (or simply: the vector), associated with each transition at (i,j).

The first entry in the vector, representing the j-th transition in DM^i, is always the delay of this transition, τ_{ij}. The next entries represent the interconnections between the transition with other transitions, in other DMs. The structure of the vector is shown in Table 2.

TABLE 2. Definition of Transition Vector

k	Entry	Comment
1	τ_{ij}	the delay
2	MINP	code indicating the presence of multiple inputs from other DMs into this transition
3	INUM	number of inputs
4	LI	lowest index from of DM providing input
5	LIJ	transition index from DM^{LI}, connected to current transition
6	...	same entry as (4,5) for next DM's input
7	...	.
.		.
.		.
.		.
2*INUM+4	MOUT	Code indicating the presence of multiple outputs from this transition to other DMs
2*INUM+5	ONUM	number of outputs
2*INUM+6	LO	lowest index of DM^{LO} to which the output is directed
2*INUM+7	LOJ	transition index of DM^{LO} receiving the output
.	.	.
.	.	.
.	.	.
2*INUM+2*ONUM+5		last entry

If the transition has only one input and one output, the vector structure will be more simple:

τ_{ij}	delay
I	index of DM providing the input
IJ	index of the transition in DM providing the input
O	index of DM receiving the output
OJ	index of the transition in DM receiving the output.

If in the above an output or input to another DM is absent, their respective two entries are filled with zeros. If the (i,j) transition is not connected at all to outside DMs, its 2nd to 5th vector positions are zero:

$$[\tau_{ij}, 0, 0, 0, 0]'$$

Thus, the minimal vector size in any transition is five. If a switch is present that directs a signal to any one of s possible transitions, then the vector will take the form

$$[\tau_{ij}, 0, 0, 0, f^{1}_{j+1}/f^{2}_{j+1}/\ldots/f^{s}_{j+1}]'$$

The A_s array will be called the System Array. Its general structure is:

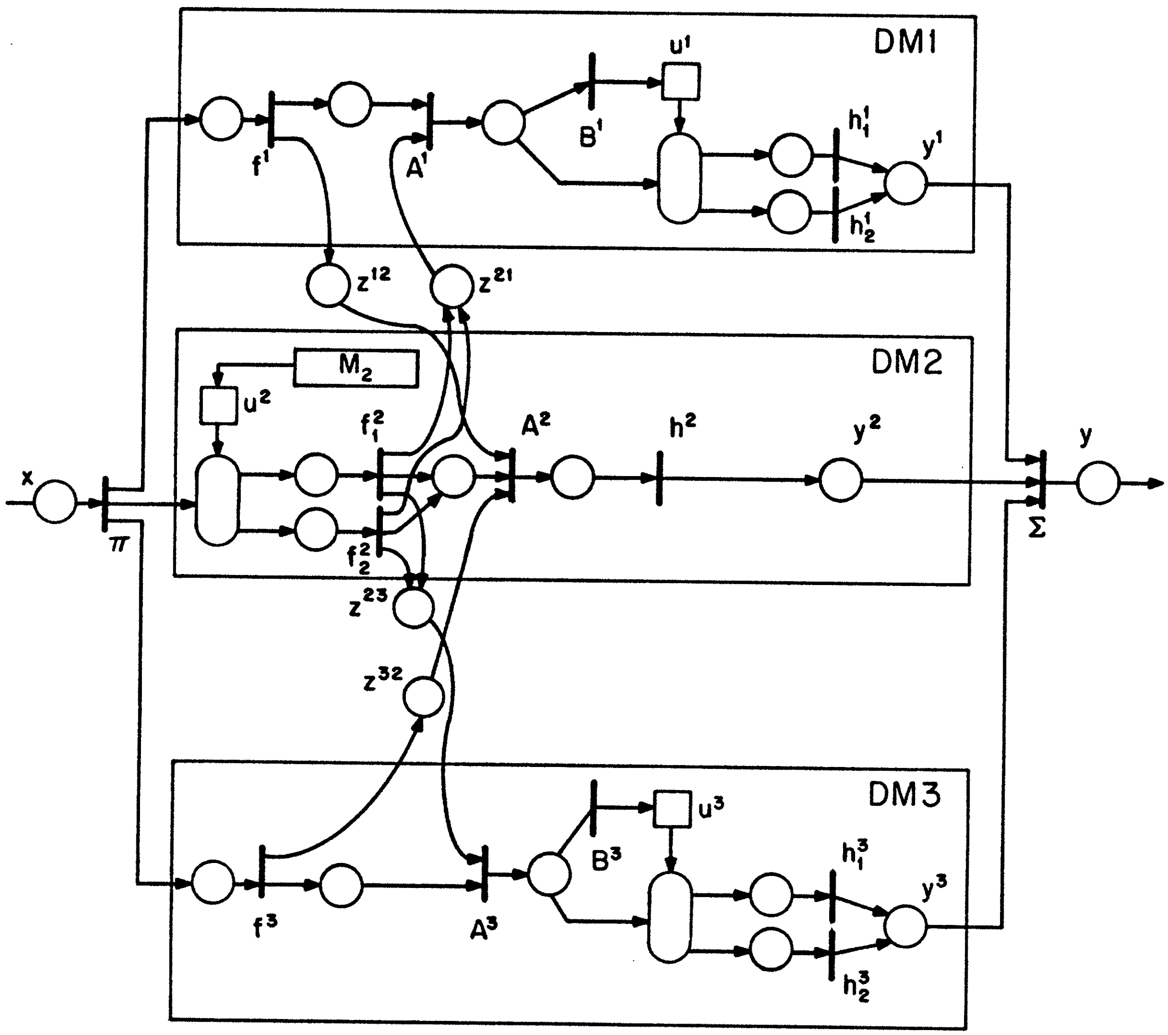
$$
\begin{array}{rc}
 & \begin{array}{cccc} \text{Transition index:}\ 1 & 2 & \cdots & n \end{array} \\
\text{DM index} & \\
A_s = \begin{array}{c} 1 \\ \\ \\ \\ 2 \\ \\ \\ \\ \vdots \\ m \\ \\ \\ \\ \end{array} &
\left[\begin{array}{cccc}
\tau_{11} & \tau_{12} & \cdots & \tau_{1n} \\
- & - & & - \\
- & - & & - \\
- & - & & - \\
\tau_{21} & \tau_{22} & & \tau_{2n} \\
- & - & & - \\
- & - & & - \\
- & - & & - \\
\vdots & \vdots & & \vdots \\
\tau_{m1} & \tau_{m2} & & \tau_{mn} \\
- & - & & - \\
- & - & & - \\
- & - & & -
\end{array}\right]
\end{array}
$$

The structure of the A_s array will be illustrated by two examples of DM systems.

5. EXAMPLE: THREE-PERSON ORGANIZATIONS

To illustrated the procedure for determing the system arrays, two three-person organizations will be described. Considered a simple air defense problem. The organization designer has available up to three batteries of surface-to-air missiles and the associated sensors. Each unit can sense threats in a sector and respond only to threats that are within that sector. Let the trajectory of a threat be defined by two measurements, each an ordered pair of coordinates. From the two ordered pairs, the location, direction, and speed of the threat can be determined. On the basis of that information, the sector battery can respond to the threat. However, threat trajectories can straddle sectors; consequently, the adjacent sectors must communicate with each other to pass threat information to the sector that can best respond to it. Two such organizational structures have been considered [8].

In the first structure, Organization A, the designer is using three batteries in parallel; one third of the area is assigned to each battery. The second decisionmaking unit, DM^2, has to coordinate with both decisionmaking units, DM^1 and DM^3. The coordination takes the form of information sharing, as shown in Figure 6.



Organization A

Figure 6. Petri Net Representation of Organization A

In the second structure, Organization B, the designer is using only two batteries, but a supervisor, or coordinator, has been introduced. Threat information near the boundaries of the two sectors is communicated to the supervisor, DM^2, who then decides which battery should be assigned to that particular threat. There is no information sharing between DM^1 and DM^3. However, there are command inputs from DM^2 to both DM^1 and DM^3. The Petri Net representation of Organization B is shown in Figure 7.

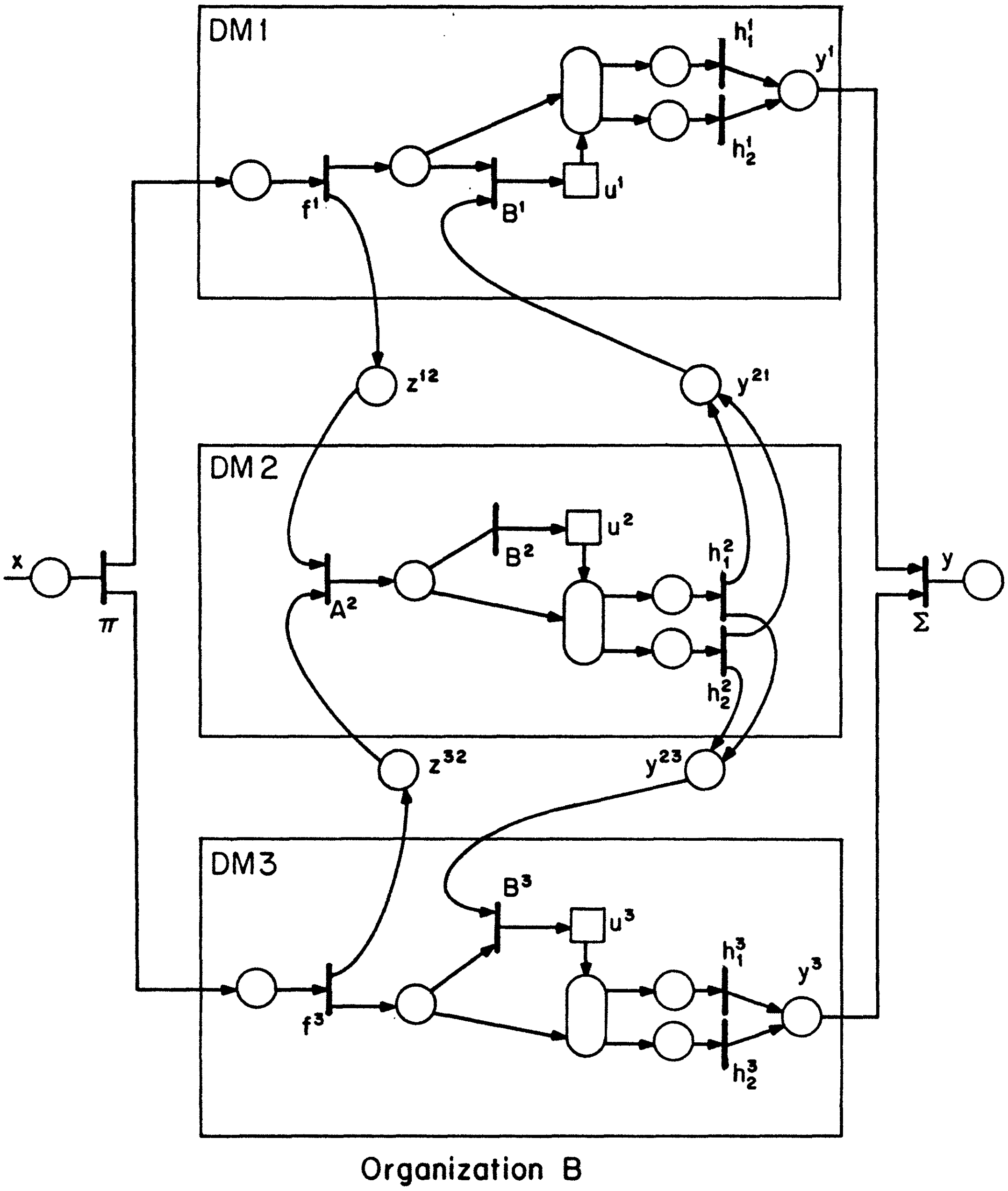


Figure 7. Petri Net Representation of Organization B

In order to construct the system array for each organization, each overall system is divided into subsystems. The system array matrix for each subsystem is determined first and then these arrays are combined to form the overall system array.

Consider first Organization A. Five subsystems are identified (Fig. 8). The first one is the model of the input source (SI) and the distribution of the inputs -- the threat information -- to the three decisionmaking units. It consists of the place x and the transition π. The three DMs constitute the next three subsystems. The fifth subsystem is the organization's output, which consists if the combined output of the three DMs. It is represented by the transition $\sum$ and is denoted by SO. Each subsystem's array constitutes a block in the overall system array, A_s, shown in Table 3.

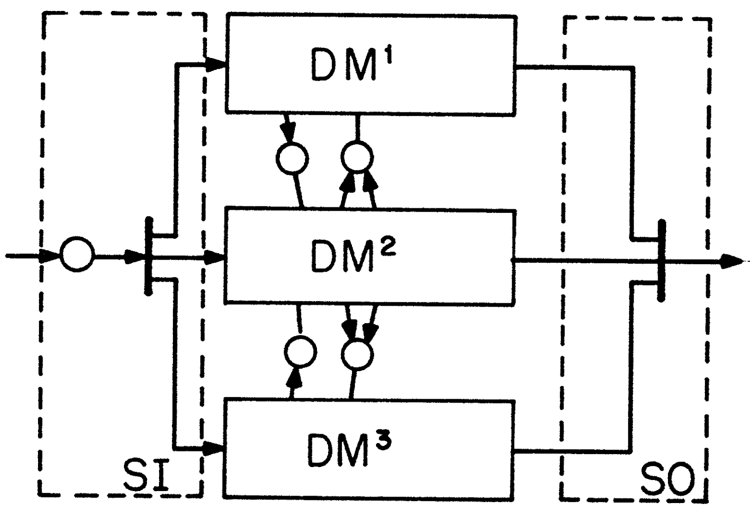


Figure 8. Five Subsystems in Organization A

TABLE 3. <u>The System Array for Organization A</u>

$$
A_S = \begin{array}{cc}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} \mathrm{SI} \\ \\ \mathrm{DM}^1 \\ \\ \mathrm{DM}^2 \\ \\ \mathrm{DM}^3 \\ \\ \mathrm{SO} \end{array} &
\left[\begin{array}{ccccc}
\begin{matrix} \tau \\ 0 \\ 0 \\ \mathrm{MOUT} \\ 3 \\ 1 \\ 1 \\ 2 \\ 1A/1B \\ 3 \\ 1 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \\
\begin{matrix} \tau \\ \mathrm{SI} \\ 1 \\ 2 \\ 2 \end{matrix} &
\begin{matrix} \tau \\ 2 \\ 1A/1B \\ 0 \\ 0 \end{matrix} &
\begin{matrix} \tau \\ 0 \\ 0 \\ 0/0 \\ 0/0 \end{matrix} &
\begin{matrix} \tau \\ 0 \\ 0 \\ \mathrm{SO} \\ 1 \end{matrix} &
\begin{matrix} \tau \\ 0 \\ 0 \\ \mathrm{SO} \\ 1 \end{matrix} \\
\begin{matrix} \tau \\ \mathrm{SI} \\ 1 \\ \mathrm{MOUT} \\ 2 \\ 1 \\ 2 \\ 3 \\ 2 \end{matrix} &
\begin{matrix} \tau \\ \mathrm{SI} \\ 1 \\ \mathrm{MOUT} \\ 2 \\ 1 \\ 2 \\ 3 \\ 2 \end{matrix} &
\begin{matrix} \tau \\ \mathrm{MINP} \\ 2 \\ 1 \\ 1 \\ 3 \\ 1 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} \tau \\ 0 \\ 0 \\ \mathrm{SO} \\ 1 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \\
\begin{matrix} \tau \\ \mathrm{SI} \\ 1 \\ 2 \\ 2 \end{matrix} &
\begin{matrix} \tau \\ 2 \\ 1A/1B \\ 0 \\ 0 \end{matrix} &
\begin{matrix} \tau \\ 0 \\ 0 \\ 0/0 \\ 0/0 \end{matrix} &
\begin{matrix} \tau \\ 0 \\ 0 \\ \mathrm{SO} \\ 1 \end{matrix} &
\begin{matrix} \tau \\ 0 \\ 0 \\ \mathrm{SO} \\ 1 \end{matrix} \\
\begin{matrix} \tau \\ \mathrm{MINP} \\ 3 \\ 1 \\ 4A/4B \\ 2 \\ 3 \\ 3 \\ 4A/4B \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}
\end{array}\right]
\end{array}
$$

The first column in the SI subsystem corresponds to transition π. The input x is processed by transition π with time delay τ and the result is transmitted to the first transition of the three DMs; this is indicated by the entry (MOUT,3) in the fourth and fifth rows of the first column. Since there is a switch in DM^2 that directs the flow to either one of two transitions in parallel, f_1^2 and f_2^2, they are denoted by 1A and 1B, respectively. The slash (/) in the entry in the ninth row indicates the existence of the switch; either 1A or 1B are possible.

The first column of the DM^1 array represents transition f^1. The delay is τ. the next entry, SI, is the index indicating the external source of the input to this transition. The third entry, 1, denotes the transition in SI, namely, π. The fourth entry, 2, denotes the destination of the output of f^1, in this case DM^2, while the last entry, also 2, denoted the second transition (A^2) in DM^2. The second column shows that a signal may come from either f_1^2(1A) or f_2^2(1B) of DM^2. The third column models transition B^1, while the last two columns model h_1^1(4A) and h_2^1(4B), respectively. The output of these transitions goes to the first transition Σ, of the SO subsystem.

The first column of the DM^2 array illustrates a more complex case. The delay is τ and the input comes from the transition π of the SI subsystem. There are two external outputs. Therefore, MOUT is present followed with the entry 2 below it. One external output, goes to DM^1, to his second transition (1,2) while the other goes to DM^3, to his second transition (3,2).

In a similar manner, the complete matrix is constructed and interpreted. The corresponding array for Organization B, denoted by B_S, is shown in Table 4.

The System Array representation, facilitates the calculation of delays of signal and message propagation along various paths within an organization. It also suggests a procedure for an automated, computer-based calculation. A compuational approach to the calculation of the delays is described in the next section.

TABLE 4. The System Array of Organization B

B_S =

	1	2	3	4
SI	τ 0 0 MOUT 2 1 1 3 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
DM^1	τ SI 1 2 1	τ 2 3A/3B 0/0 0/0	τ 0 0 SO 1	τ 0 0 SO 1
DM^2	τ MINP 2 1 1 3 1 0 0	τ 0 0 0/0 0/0	τ 0 0 MOUT 2 1 2 3 2	τ 0 0 MOUT 2 1 2 3 2
DM^3	τ SI 1 2 1	τ 2 3A/3B 0/0 0/0	τ 0 0 SO 1	τ 0 0 SO 1
SO	τ MINP 2 1 3A/3B 3 3A/3B 0 0	0 0 0 0/0 0/0	0 0 0 0 0	0 0 0 0 0

6. DELAY CALCULATION USING THE SYSTEM ARRAY

The delays in the transmission of information in a decisionmaking system are calculated starting with any input point and ending with any output point. If there are any transition subsystems at the input or output of the organization, their delays are also taken into account. This is the case for both organizations A and B; there is an input subsystem SI and an output subsystem SO.

The delays can be calculated by direct inspection of the Petri Net reprsentation of a decisionmaking model, provided that the individual delays of each transition are specified. The System Array, introduced in the previous section, contains sufficient information to permit an orderly calculation of message propagation delays within the system, from any input point to any output point. A basic approach to perform such a calculation is discussed in this section.

Suppose that the delay of message propagation from the input point of DM^i to the output point of DM^j is to be calculated. The set of vectors in the system array corresponding to the i-th DM will be scanned for any outgoing (presence of an MOUT code) interconnections. A list of destination DMs (that is, DM to which DM^i is transmitting information) is formed, along with the transition indices of these DMs, receiving the message from DM^i. The transitions of DM^i, out of which the messages emanate, are also noted. The forward path delay from the input to DM^i to the various output interconnections is computed and stored.

Suppose, for the sake of the argument, that DM^i, due to the existing interconnections, transmits information to DM^1 and DM^3, at the input of transition 2 in each. The sub-arrays for DM^1 and DM^3 will then be scanned, starting with the second element, in a manner similar to the one used on DM^i. The forward path delay is calculated and any further interconnections to other DMs are noted and pursued. The order of the calculation of the delays is arranged in a three-structure which permits an exhaustive search through all possible paths (Fig. 9).

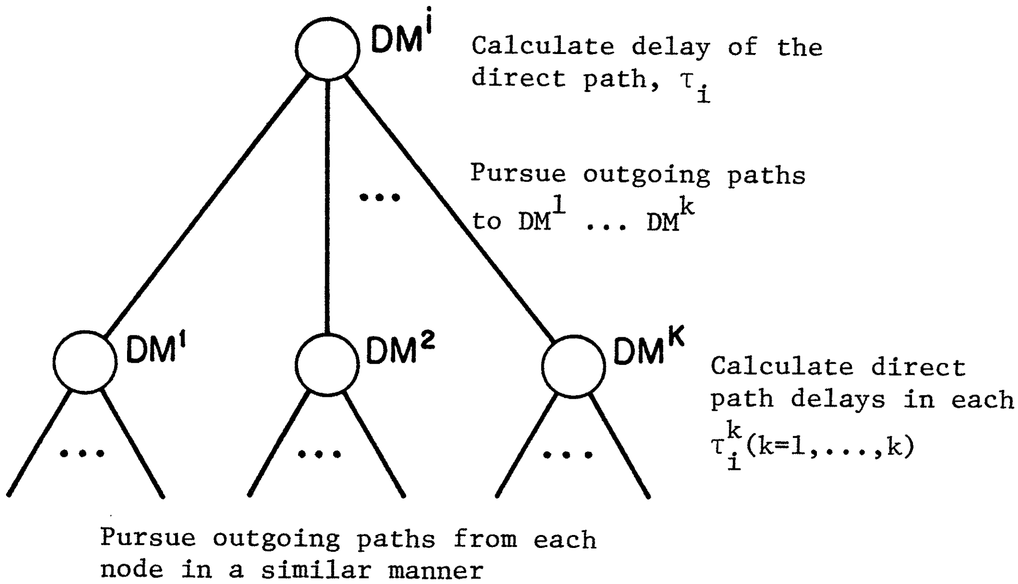


Figure 9. The Tree-Structured Delay Calculation in the Decisionmaking Model

Starting with each DM, represented by a certain node in the tree structure, the pursuit of all outgoing paths is done by an ascending order of indices. In this way one makes sure that no possible path is omitted. The DM where the interconnection path originates will be called the Source DM, and the DM where the path terminates the Destination DM. One has to arrange for an appropriate array to store the transition index in the source DM and the transition index in the destination DM for each interconnection path.

The above procedure will be illustrated by the two examples, Organizations A and B, used in the previous section.

Example A: Calculate the delay from the input x to DM^2 to the output due to DM^1 in Organization A (Fig. 6).

The appropriate system array, A_S, is listed in Table 3. Since the problem statement specifies the input x, the sub-array representing the input subsystem, SI, is scanned first. The second row is zero implying no external inputs; therefore the next row to be scanned is the fourth one to determine whether there is an output to DM^2, i.e., whether number 2 is present. There is a 2 in the eight row of the first column; this signifies that the output of SI is transmitted to DM^2 and, specifically to either one of the two transitions in parallel, 1A and 1B. Since no other column has 2 as an output destination the delay associated with this first step is read from the first entry of column 1; it is τ.

Now attention is focused on the third sub-array, the one corresponding to DM^2. The second row is scanned ti identify the columns, or transitions, that receive inputs from SI. As expected, both columns 1 and 2 have SI as their second element. However, as indicated earlier, the presence of the slash in 1A/1B indicates that these are two alternative paths. The sequence of events and the existence of alternative paths is recorded in the form of a tree, as shown in Figure 10. Scanning of the fourth row of column 1 establishes that this transition has multiple outputs (MOUT,2); the "1" in the sixth row denotes DM^1 and the "2" that follows indicates the second transition. Scanning of the second column in the sub-array for DM^2 produces identical information. Scanning of the remaining columns shows that there are no other paths from DM^2 to DM^1. Therefore, the appropriate forward delay, if transition 1A is selected, is τ, it is also τ if transition 1B is selected. Now attention shifts to the second sub-array in A_S that corresponds to DM^1. The second row is scanner for either a "2" or an indication of multiple targets inputs, "MINP". Only the second column passes the test. The forward delay is calculated up to the point that there is an output to the output subsystem, SO. This occurs on the fourth and fifth columns. The presence of the "0/0" in the third column signifies that columns four and five are alternative paths (see Fig. 10). Therefore, the forward delay id 3τ.

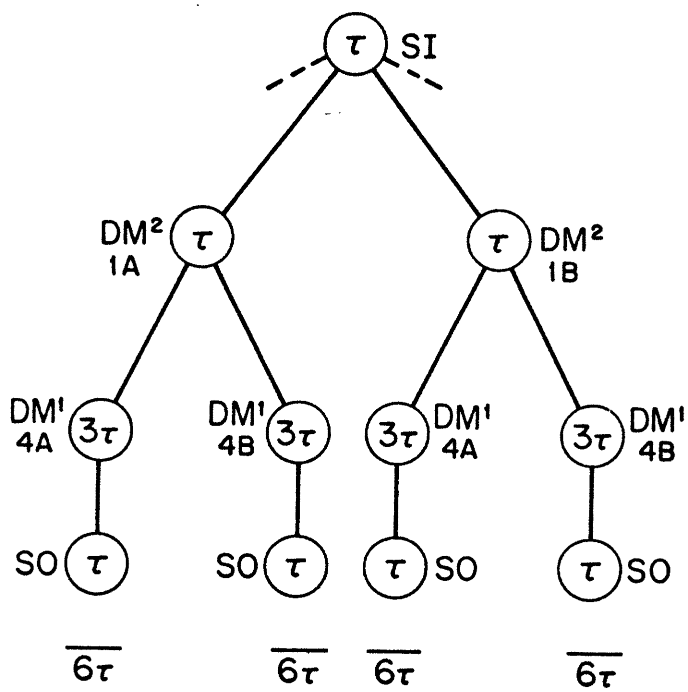


Figure 10. The Delay Calculation Tree for Example A

Finally, the SO subsystem is canned, the entries (1,4A/4B) are recognized, and the forward delay, τ, is noted.

Thus, the total delay is the sum of the forward delays in each subsystem, or 6τ, as shown in Fig. 10. In this case, all alternative paths yield in the same delay. However, this will not be the case in general.

<u>Example B</u>: Calculate the Delay for the input x to DM^1 to its own output in Organization B (Fig. 7).

The scanning starts with the SI subsystem; its forward path delay is τ. The output path from SI tp DM^1 is observed in the sixth entry of the first column. The scanning of columns then shifts to the sub-array for DM^1. The first column shows clearly that there is an output path to transition 1 of DM^2. The forward delay to that point is τ. The scan of the columns of the DM^2 subarray shows that there are two alternative paths (see 0/0 in column 2) that within DM^2, but both lead back to the second transition in DM^1. The forward delay of either path is 3τ. Similarly, there are two paths within DM^1, each leading to the output to SO. The forwrd delay is calculated to be

2τ. Finally, the delay in the SO subsystem is τ. The total delay is then $\tau + \tau + 3\tau + 2\tau + \tau$, or 8τ. The corresponding tree is shown in Figure 11.

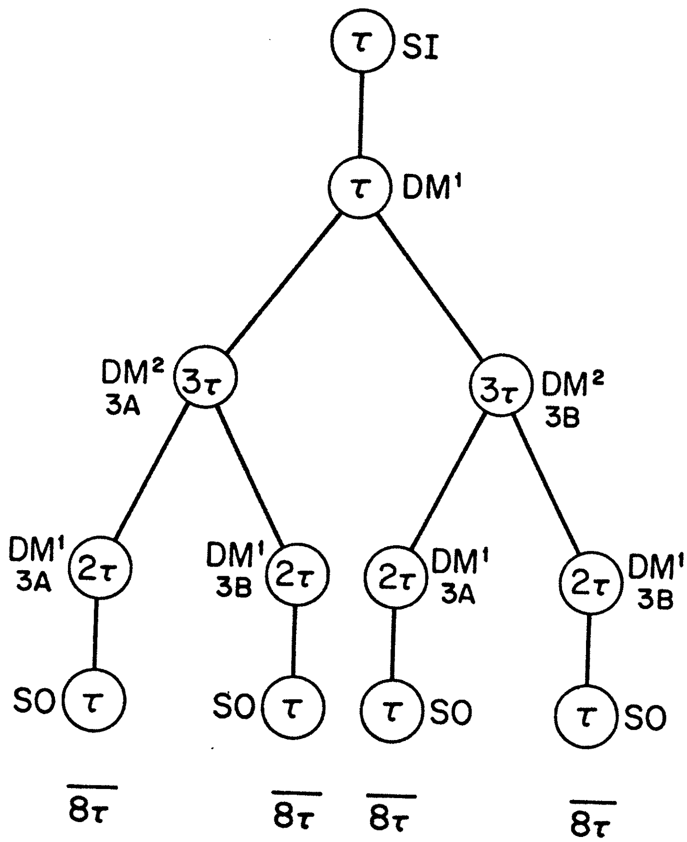


Figure 11. The Delay Calculation Tree for Example B

Following down the <u>root of the tree</u>, representing the strating subsystem for the delay calculation, and continuing along the intermediate nodes of the tree the terminal ones, one actually follows through all of the possible paths of message transmission in the system, starting at a specified point. Stored along each node (which represents the path within a specific subsystem), is the delay accumulated in that particular subsystem. One should provide, of course an appropriate array to store the delays along the possible paths, in order to be able to sum them up at the end. The total delays is the sum of the individual delays, accumulated along the nodes of the tree-structure.

7. CONCLUDING RESULTS

The formalism of representing decisionmaking models by equivalent Petri Nets has been presented. A table of basic equivalence primitives has been established. A new element in the Petri Net formalis, the Decision Switch, has been defined to satisfy the special needs of representing decisionmaking processes. Several examples have been offered.

Once a decisionmaking organization has been represented as a Petri Net, it is possible to introduce procedures for the calculation of delays within an organization. An array has been introduced that contains the structural information contained in the Petri Net and the delays associated with transitions. An algorithm for computing delays in the simple case of synchronous protocols has been introduced and illustrated on two three-person organizations. Current work is focused on the development of the necessary formalism for expressing unambiguously complex communication protocols between decisionmakers and the calculation of delays when the individual transition delays are arbitrary (asynchronous protocols).

8. ACKNOWLEDGEMENT

The authors would like to thank Ms. Victoria Jin for her contributions to this paper.

9. REFERENCES

[1] J. L. Peterson, "Petri Nets," ACM Computing Surveys, Vol. 9, No. 3, pp.223-252, September 1977.

[2] T. Agerwala, "Putting Petri Nets to Work," IEEE Computer, Vol 12, No. 12, pp. 85-94, December 1979.

[3] A. H. Levis and K. L. Boettcher, "Information Structures in Decisionmaking Organizations," Proc. of MELECON 83, Athens, Greece, May 1983; also LIDS-P-1275, Laboratory for Information and Decision Systems, MIT, February 1983.

[4] A. H. Levis, "Information Processing and Decisionmaking Organizations: A Mathematical Description," LIDS-P-1333, October 1983; to appear in Large Scale Systems, North-Holland, 1984.

[5] S. A. Hall and A. H. Levis, "Information Theoretic Models of Memory in Human Decisionmaking Models," LIDS-P-1300, Proc. of 6th MIT/ONR Workshop on C^3 Systems, Laboratory for Information and Decision Systems, MIT, July 1983.

[6] J. C. Boyce, Digital Logic and Switching Circuits, Prentice Hall, Englewood Cliffs, NY, 1975.

[7] T. R. Blakeslee, Digital Design with Standard MSI and LSI, Wiley, NY, 1975.

[8] A. H. Levis, K. L. Boettcher, Decisionmaking Organizations with Acyclical Information Structures, IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-13, No. 3, pp. 384-391, May/June 1983.