MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

AD-A191 952

COORDINATION IN DECISION-MAKING ORGANIZATIONS*

by

Jean-Louis M. Grevet

Alexander H. Levis **

Laboratory for Information and Decision Systems

Massachusetts Institute of Technology

Cambridge, MA 02139

USA

ABSTRACT

A methodology to analyze, model, and evaluate decision-making processes that require coordination is presented. The issues of inconsistency of information and synchronization are emphasized. Predicate Transition Nets are used as the basic technique for representing organizational structures and for characterizing the coordination of processes. Protocols of interaction are modeled by transitions for which the rule of enablement is that the decisionmakers, when interacting, must refer to the same state of the environment. Two measures of coordination are then introduced: the degree of information consistency and the measure of synchronization. These measures are defined on the basis of the attributes of the tokens belonging to the input places of transitions modeling interactions. A recently developed simulation system for Predicate Transition Nets is used for investigating the dynamics of decisionmaking processes requiring coordination.

**Please addrress all correspondence regarding this paper to Dr. Alexander H. Levis, MIT, Room 35-410/LIDS, Cambridge, MA 02139, USA.

1   88 2 03 015

INTRODUCTION

The complexity of today's industrial environment - the tempo is fast and the data to be gathered and analyzed are numerous - imposes severe constraints on the managers who must respond accurately and in a timely manner to the problems that arise. The distribution of the information processing tasks among the managers is a necessity in order to reduce each organization member's workload. Their activities are synergistic; therefore, the decision-making processes must be coordinated in order to improve the effectiveness of the organization. In this context, it is important to analyze and evaluate the concept of coordination in decision-making organizations.

The framework used to address this problem is the quantitative methodology (Levis, 1984) for the analysis and evaluation of alternative organizational structures. Petri Nets have been used (a) to show explicitly the interactive structure between decisionmakers and the sequence of operations within an organizational structure (Tabak and Levis, 1985); and (b) to model asynchronous and concurrent processes.

In order to provide some insight on the cohesiveness of organizations carrying out well-defined tasks, a mathematical description of coordination is developed as it relates to decision-making processes. The Predicate Transition Net formalism (Genrich and Lautenbach, 1981) used in this paper builds on Petri Net theory (Brams, 1983), but allows the modeling of coordination based on the attributes of symbolic information carriers in the net. In this model, when decisionmakers interact, they must have some protocol to recognize that they are exchanging information pertaining to the same event. Two measures for evaluating coordination are introduced: the degree of information consistency and the measure of synchronization. The latter measure relates to the value of information when the decisionmakers actually process it.

PREDICATE TRANSITION NET MODEL OF COORDINATION

The organizations under consideration consist of groups of decisionmakers processing information originating from a single source and who interact to produce a unique organizational response for each input that is processed. In terms of Petri Nets, it implies that there exists a source place, p_{so}, and a sink place, p_{sk}: the source place models the source from where the inputs originate; the sink place models the place where the organizational response appears at the end of the process.

A resource place, p_{rs}, is introduced to model the limited organizational resources. A transition t_{par} models the partitioning of the inputs. Furthermore, if several decisionmakers provide responses that must be fused in order to obtain the organizational response, this stage of response fusion is modeled by the transition t_{rf} (see Figure 1).
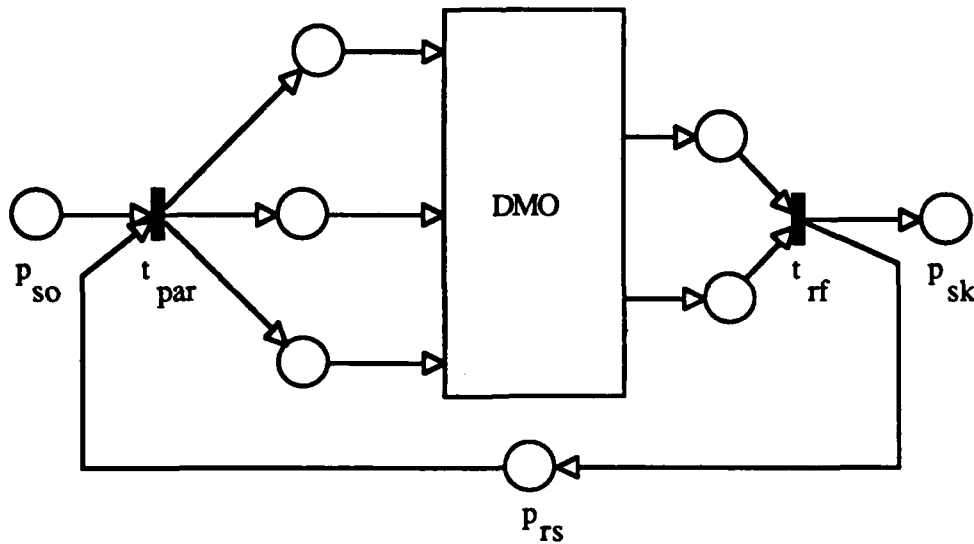

Fig. 1 Petri Net of Interactions between a Decisionmaking Organization and the Environment

The source generates inputs that arrive sequentially and one at a time: it corresponds to the time of appearance of a token in the source place p_{so}. Furthermore, it is assumed that the different decisionmakers have some means of recognizing that, when they interact, they are referring to the same input; for example, their protocols of interaction can require that they communicate the time at which the input they are processing entered the organization.

The task is modeled by the alphabet $X = \{x_1,...,x_n\}$. A probability distribution is defined on X. The probability that the input x is equal to x_i, i.e., $prob(x=x_i)$, is denoted by $prob(x_i)$. The set of subsets of X is denoted by $\Pi(X)$; then:

$$\Pi^*(X) = \Pi(X) - \{\emptyset\} \tag{1}$$

where $\emptyset$ denotes the empty set.

A clock is assumed to exist. This clock is used to follow the instants at which the process is observed. In accordance with the formalism of Timed Petri Nets, this clock provides

3

non-negative rational numbers. Therefore, the current time is denoted by T_c, such that $T_c \in Q^+$ where Q^+ denotes the set of non-negative rational numbers.

Distinguishability of Tokens

The fundamental assumption of the model is that a decisionmaker can process only one input at a time in any of his internal stages: it follows that any other input that is ready to be processed by the same stage waits in memory. Therefore, queues of inputs can build in the system.

At any internal stage of the decision-making process, a decisionmaker can discriminate between different items of information on the basis of three characteristics:

- the time T_n at which the inputs that these items of information represent entered the organization.
- the time T_d at which the item of information entered the internal stage where it is currently located.
- the class C associated with any item of information by the previous processing stage.

The definition of the attributes T_n, T_d and C derives from the following considerations:

(i) inputs originate from a single source, one at a time. For each token, the attribute T_n corresponds to the time at which the input represented by this token entered the organization. In accordance with the formalism of Timed Petri Nets, the first attribute T_n is a non-negative rational number, i.e., an element of Q^+.

(ii) the firing of any token in the net takes an amount of time that is known since it characterizes the processing time of the corresponding transition. It is thus possible to assign to any token in a place p with the time T_d at which it entered this place. This second attribute T_d is also an element of Q^+.

(iii) the task is modeled by the alphabet $X = \{x_1,...,x_n\}$. It is assumed that each place p is associated with a partitioning $D(p)$ of this alphabet. The number of elements of this partitioning is denoted by $e(p)$. This partitioning is such that $D(p) = \{D(p,1),..., D(p,e(p))\}$ where $D(p,i)$ denotes an element of $\Pi^*(X)$. Thus, the third attribute C of each token belongs to a certain partitioning $D(p)$ of X, this partitioning depending on the place p where the token is located.

4

The different resources that the organization has are assumed to be indistinguishable; this might not be the case when organizational resources are allocated to different inputs in accordance with some doctrine. In the same way, the resources that represent the decisionmakers' processing capacities are not distinguishable.

Consequently, three types of places are defined: *Memory places* carry information internally processed by each decisionmaker; *structural places* carry information exchanged between a decisionmaker and the environment or other organization members; and *resource places* that model the limitation of resources that constrains the processing of information by individual DMs. Memory and structural places contain tokens that have an identity since they model information carriers, while resource places contain tokens with no identity since they model resources.

Each place is associated with one of the variables χ or ϕ. The variable ϕ takes its values in the set Φ such that $\Phi = \{\phi\}$. All the tokens with no identity are denoted by the color ϕ. The variable χ takes its values in the set X such that $X = Q^+ \times Q^+ \times \Pi^*(X)$. Each element of X is a color that is represented by (T_n, T_d, C). A token with an identity is an individual that is assigned a color.

The marking of PN is defined as follows: For each place p, $Z(p)$ designates the set of applications from $\underline{v}(p)$ into N where $\underline{v}(p)$ designates the set of values where the variable associated to the place p takes its values and N denotes the set of non-negative integer numbers. Z_n is the set of $Z(p)$ for all p. The marking M of the Petri Net PN is defined as an application from the set of places, P, into the set Z_n:

$$M : P \rightarrow Z_n$$
$$p \rightarrow M(p) \qquad\qquad (2)$$

For each place p, $M(p)$ is an application from $\underline{v}(p)$ into N. It assigns to each value of the variable associated with p a non-negative integer number. It represents the number of tokens in the place that have the corresponding color. If m designates a certain color, $M(p)[m]$ will denote this number. Since each color m corresponds to a 3-tuple (T_n, T_d, C), this number will be also denoted by $M(p)[(T_n, T_d, C)]$. In the case of a resource place, the tokens can have only the color ϕ. $M(p)[\phi]$ will be denoted by $M(p)$.

5

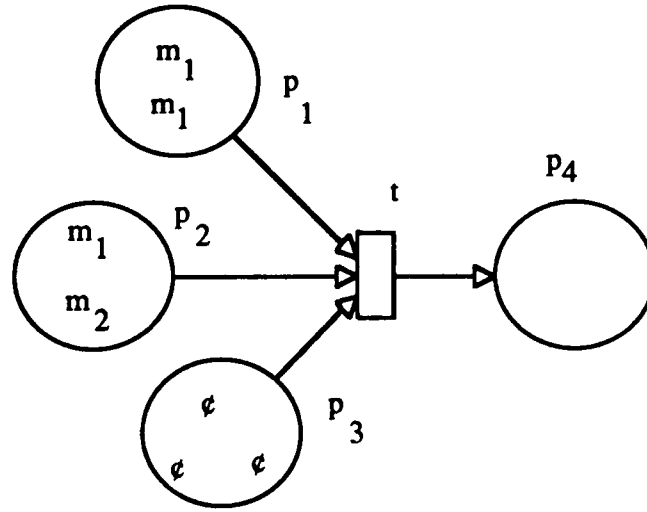The following example (Figure 2) illustrates these definitions:

Fig. 2 Example of Marking

In this example, the following relations hold:

- $m_1 \in X$, $m_2 \in X$.
- $M(p_1)[m_1] = 2$; $\forall\ m \in X-\{m_1\}$, $M(p_1)[m] = 0$.
- $M(p_2)[m_1] = 1$; $M(p_2)[m_2] = 1$; $\forall\ m \in X-\{m_1, m_2\}$, $M(p_2)[m] = 0$.
- $M(p_3) = 3$.
- $\forall\ m \in X$, $M(p_4)[m] = 0$.

The firing of a transition t, as illustrated in Figure 3, is characterized by the following:

- if p' is a resource place, m' is the color ¢.
- if p and p' are memory or structural places, m and m' are elements of X.

The attribute T_n characterizes one and only one input since the source generates one input at a time. Furthermore, two representatives of the same input cannot stand in the same place. Indeed, the net is an Event-Graph and, so, each place has only one input transition which produces in each of its output place only one token per firing.

Proposition 1: A place cannot contain two tokens which have the same attribute T_n.

6
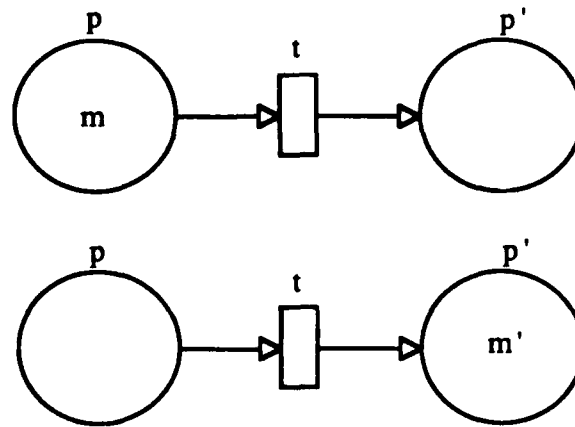
Fig. 3 Firing of a Transition

Protocols of Interaction

One must recall that the set of input places of any transition t can contain a resource place. The rule according to which the resource place must contain at least a token in order for t to be enabled will apply. However, since resource places do not constrain the rule of enablement of a transition, but by requiring the presence of a token, the discussion on enablement that follows focuses on structural and memory places. The Petri Net model of transitions where fusion of data is done is shown in Figure 4.

Fig. 4 Petri Net Model of Interaction with Fusion of Data

When the fusion of data is performed by a decisionmaker, only one of the places p_1,..., p_r is a memory place. We denote it by p_k. Any rule of enablement can be introduced at this point. It can

7

take explicitly into account the attributes of the tokens in the input places, but this is not necessary. For example, here are two possible rules. M denotes the marking of the net:

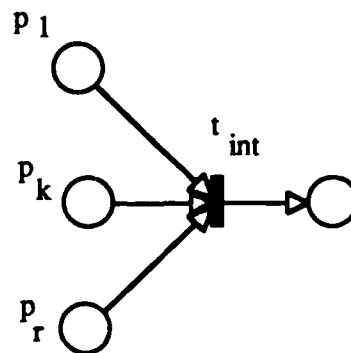**rule 1:** t_{int} is enabled, if and only if all its input places contain a token with the same value of the attribute T_n

$$\forall \, p \in Pre(t_{int}) \, , \exists \, (T_n, T_d, C) \in Q^+ \times Q^+ \times \Pi^*(X) \, , \, M(p)[(T_n, T_d, C)] \neq 0,$$

$$\forall \, p' \in Pre(t_{int}) \, , \, p' \neq p \, , \exists \, (T_n', T_d', C') \in Q^+ \times Q^+ \times \Pi^*(X) \, , M(p')[(T_n', T_d', C')] \neq 0,$$

$$T_n = T_n' \tag{3}$$

Rule 1 means that the transition t_{int} is enabled if and only if all the places of its preset contain at least a representation of the same input. Indeed, it results from the fact that memory and structural places contain only tokens of the (T_n, T_1, C) type, and that tokens having the same attribute T_n represent the same input. From the organizational standpoint, it means that, when decisionmakers interact, they must refer to the same input. Rule 1 will hold in the remainder of the paper. The representation of rule 1 is illustrated in Figure 5.
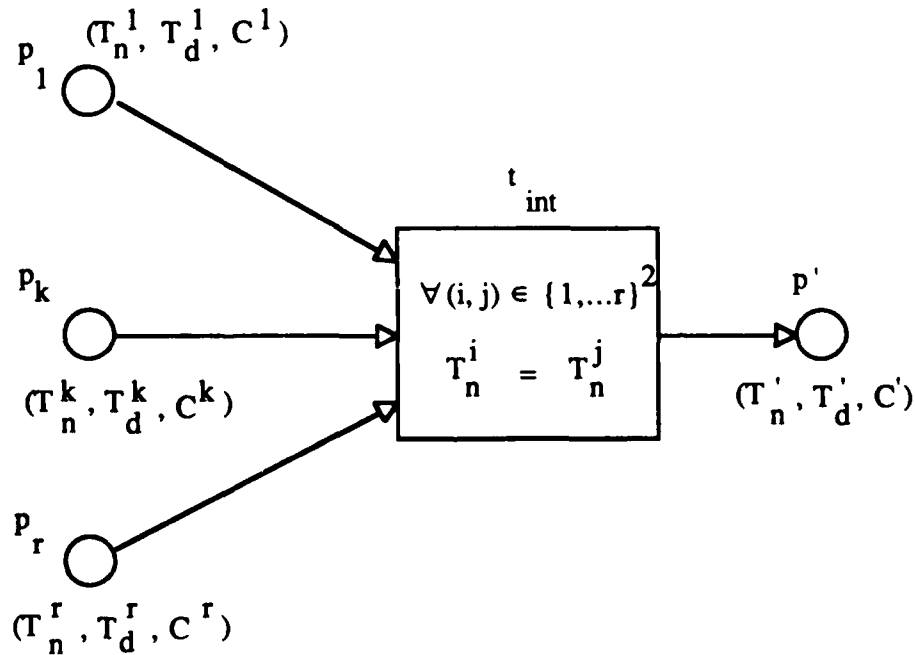

Fig. 5 Predicate Transition Net Model of Interaction

rule 2: t_{int} is enabled if and only if rule 2 applies or there exists a token in the memory place p_k which has been in it for more than d units of time.

(rule 1) or

$$(\exists \, (T_n, T_d, C) \in \mathbf{Q}^+ \times \mathbf{Q}^+ \times \Pi^*(\mathbf{X}) , M(p_k)[(T_n, T_d, C)] \neq 0, (T_c - T_d) \leq d) \qquad (4)$$

Rule 2 means that rule 1 applies when a token has been in the memory place for less than d units of time. This rule models the interactions where decisionmakers wait for information from other parts of the organization only for a certain amount of time.

In this paper, a transition will be enabled if and only if rule 1 is verified. In the case of internal transitions, rule 1 is always verified when all its input places have a token since the preset contains only one place that is not a resource place. Therefore, the rule of enablement can be expressed formally as:

$$\forall \, (i, j) \in \{1, ..., r\} \times \{1, ..., r\}, \qquad T_n^i = T_n^j \qquad (5)$$

It means that the attributes T_n^i of the colors $m_1, ..., m_r$ must have the same value.

Token Selection

The problem of token selection arises since the tokens are distinguishable. Rules of selection must be applied to select the tokens that will be fired for any firing of a transition t. These rules operate on the tokens of the input places that enable the transition t. This is illustrated by the example of Figure 6 where rule 2 of enablement applies.

In this case, we suppose that:

- $m_1 = (T_n^1, T_d^1, C^1)$; $m_1' = (T_n^1, T_d'^1, C'^1)$; $m_1'' = (T_n^1, T_d''^1, C''^1)$.
- $m_2 = (T_n^2, T_d^2, C^2)$; $m_2' = (T_n^2, T_d'^2, C'^2)$; $m_2'' = (T_n^2, T_d''^2, C''^2)$.

9
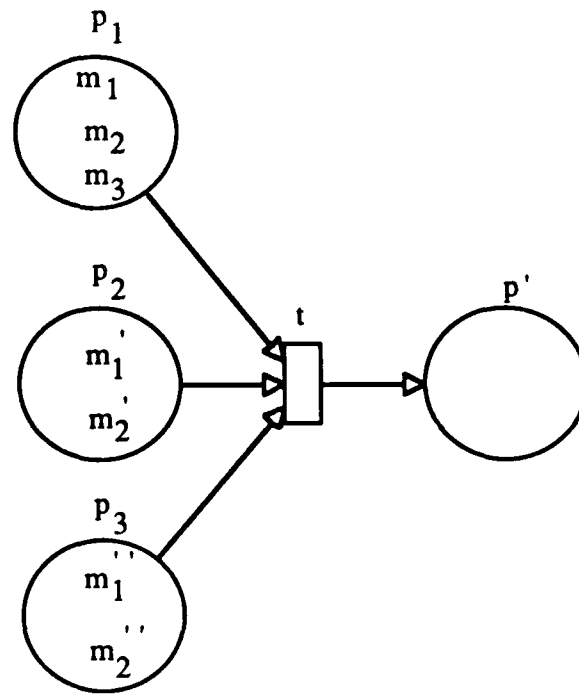
Fig. 6 Token Selection

Since the enabling condition is (5), it follows that the transition t is enabled by the set $\{m_1, m_1', m_1''\}$ and by the set $\{m_2, m_2', m_2''\}$. Therefore, a rule must decide what tokens will be removed by the next firing of transition t.

It is assumed that this rule works as follows: it selects a token in a certain place p of the preset of transition t ; then the set of tokens removed is the one to which the token selected belongs. It is will be shown in Proposition 2 that a token can belong to one and only one such set. Therefore, before applying the rule to the place p, it is necessary to decide in which place p the selection will be done. One can see on the example of Figure 6 that p_1, p_2 and p_3 contain each two tokens that enable transition t. This means that the selection of the tokens that will be fired next can be done in place p_1 or place p_2 or place p_3. Thus, a choice must be made to decide if the token selection rule will apply on p_1 or p_2 or p_3.

Different strategies can be applied to choose the place on which the token selection rule will operate; for example:
- the decisionmaker considers only his own information in order to discriminate between the various items of information that he can continue to process. In such a

10

case, the token selection rule would apply to the memory place.
- the decisionmaker relies more on the information sent by another member, and, therefore, chooses the next piece of information to process on the basis of the data that he receives from this decisionmaker.

In this paper, the choice of the place on which the token selection rule will apply is done according to some well-known rule PS(t), for each transition t, given the state of the system. Once the rule PS(t) has been applied, the place p on which the token selection rule will apply is determined. Then, the selection of a token in this place determines an attribute T_n. The knowledge of this attribute allows to select the corresponding tokens in the other places. In the example above, if PS(t) selects p_1, the token selection rule must discriminate between m_1 and m_2 and m_3. If m_2 is selected, then m_2' and m_2'' are automatically selected in places p_2 and p_3.

Proposition 2: The selection in the place p of a token among the tokens that can be fired by transition t determines uniquely the tokens that will be fired in the other places. Once a token has been selected in the place p, its attribute T_n corresponds to one and only one token in any other place of the preset of the transition t.

Thus, it must be decided what will be the possible strategies that the decisionmakers will use in order to choose between the several pieces of information that they can continue to process in a given stage. Four types of rules of selection can be thought of:

(i) rules that discriminate with respect to the attribute T_n.
(ii) rules that discriminate with respect to the attribute T_d.
(iii) rules that discriminate with respect to the attribute C.
(iv) rules that combine different rules of the previous types.

Some example of possible rules are the following:

1/ FIFO: the decisionmaker can decide to process first the inputs that entered the organization first. In this case, the token with the lowest T_n is selected.

2/ LIFO: the decisionmaker decides to process first the inputs that entered the organization last. Then, the token with the highest T_n is selected.

3/ LOCAL FIFO (LFIFO): the decisionmaker decides to process first the inputs that entered the internal stage where they currently are first. The token with the lowest T_d is selected.

11

4/ LOCAL LIFO (LLIFO): the decisionmaker processes first the inputs that entered the internal stage where they currently are last. The token with the highest T_d is selected.

5/ PRIORITY: the decisionmaker can assign priorities to certain classes of inputs, i.e., can set priorities on the basis of the attribute C. He selects first the items of information with the highest priority.

6/ MIXED: if several pieces of information have the same highest priority, the decisionmaker can then decide to apply some rule of the type (i) to (iv) to discriminate between them.

PREDICATE TRANSITION NET CHARACTERIZATION OF COORDINATION

In accordance with the considerations developed in the previous section, the coordination of different decisionmakers that shows the extent to which their activities are synchronized and the information that they exchange is consistent can be evaluated.

The Petri Net representation of the transitions considered in this section is shown in Figure 4. The characterization of the coordination for an interaction t_{int}, using the Predicate Transition Net model introduced in the previous section, derives from the definition of an order relation on the set of tokens fired by transition t_{int}. The following binary relations are defined:

Ψ_1 is a binary relation defined on $Q^+ \times Q^+ \times \Pi^*(X)$ by :

$$((x, y, z) \, \Psi_1 \, (x', y', z')) \Leftrightarrow ((x = x') \text{ and } (y \leq y')) \qquad (6)$$

Ψ_2 is a binary relation defined on $Q^+ \times Q^+ \times \Pi^*(X)$ by :

$$((x, y, z) \, \Psi_2 \, (x', y', z')) \Leftrightarrow ((x = x') \text{ and } (z = z')) \qquad (7)$$

Ψ_3 is a binary relation defined on $Q^+ \times Q^+ \times \Pi^*(X)$ by :

$$((x, y, z) \, \Psi_3 \, (x', y', z')) \Leftrightarrow (((x, y, z) \, \Psi_1 \, (x', y', z')) \text{ and } ((x, y, z) \, \Psi_2 \, (x', y', z')))$$

$$(8)$$

The relation Ψ_3 defines an order relation on the set $X = Q^+ \times Q^+ \times \Pi^*(X)$. It derives from the fact that the relation $\leq$ defines an order relation on the set Q^+. We denote by $m_1, ..., m_r$ the elements of X which represent the colors of the r tokens removed respectively from places

12

$p_1, ..., p_r$ by transition t_{int}. m_k denotes the color of the token removed from the memory place p_k. Furthermore, each color m_i corresponds to (T_n^i, T_d^i, C^i), element of $Q^+ \times Q^+ \times \Pi^*(X)$.

The firing of t_{int} is **synchronized** if, and only if:

$$\forall i \in \{1, ..., r\} \qquad (T_n^i, T_d^i, C^i) \, \Psi_1 \, (T_n^k, T_d^k, C^k) \qquad (9)$$

This definition allows to discriminate between firings that are synchronized and firings where one or several tokens m_i arrive in their respective places later than m_k in p_k. The measure of the degradation of synchronization in the latter cases will be evaluated in the next section. In the same way, the data fused by DM_k are consistent, if they correspond to the same class C. It leads to the following definition:

The firing of t_{int} is **consistent** if, and only if:

$$\forall (i, j) \in \{1, ..., r\} \times \{1, ..., r\}, (T_n^i, T_d^i, C^i) \, \Psi_2 \, (T_n^j, T_d^j, C^j) \qquad (10)$$

On this basis, the following definition for the coordination of an interaction is obtained: The firing of t_{int} is **coordinated** if, and only if, it is synchronized and consistent.

It is possible now to characterize a coordinated transition firing by the order of arrival of the tokens in the places of its preset.

Proposition 3: When the firing of $m_1, ..., m_r$ by t_{int} is coordinated, the relation Ψ_3 induces an order relation on the set $\{m_1, ..., m_r\}$ for which m_k, token of the memory place, is the unique greatest element.

The definition of coordination applies to a single interaction. The definitions of the coordination of a single task, i.e., for a sequence of interactions concerning the same input, as well as for all tasks executed are derived as follows.

The execution of **a task is coordinated** if, and only if, it is coordinated for all interactions that occur during the task. The execution of **a Petri Net PN is coordinated** if, and only if, it is coordinated for all the tasks performed.

13

DEGREE OF INFORMATION CONSISTENCY

Given an interaction stage, t_{int} denotes the interactional transition that models this stage in the Petri Net representation, as shown in Figure 4. At each transition t_{int}, the decisionmaker DM_h associates a class C^h to each input x_i. In this section, this class is denoted by $C^h(x_i, t_{int})$. This class belongs to $D(p_h)$, a partition of the alphabet X, that the designer defines a priori.

In order to achieve a higher consistency, the designer has to ensure that the r decisionmakers who interact in the stage are provided with the same set of classes; therefore, it is assumed that:

$$\forall (i, j) \in \{1, ..., r\} \times \{1, ..., r\}, \quad D(p_i) = D(p_j) \tag{11}$$

If $m_1, ..., m_r$ designate the colors of the tokens in the preset of t_{int} that correspond to input x_i and that are fired by t_{int}, then $C^1(x_i, t_{int}), ..., C^r(x_i, t_{int})$ denote their attribute C. Let $V(x_i, t_{int})$ designate the vector $(C^1(x_i, t_{int}), ..., C^r(x_i, t_{int}))$, element of $[\Pi^*(X)]^r$. Let $prob(C^1(x_i, t_{int}), ..., C^r(x_i, t_{int}))$ denote the probability of having tokens with attribute $C^1(x_i, t_{int}), ..., C^r(x_i, t_{int})$ for the input x_i at the stage t_{int} in places $p_1, ..., p_r$. It will be written as $prob(V(x_i, t_{int}))$. If $z(V(x_i, t_{int}))$ is the number of subsets of two elements $\{C^a(x_i, t_{int}), C^b(x_i, t_{int})\}$ of $\{C^1(x_i, t_{int}), ..., C^r(x_i, t_{int})\}$, we have:

$$z(V(x_i, t_{int})) = \binom{r}{2} = \frac{r!}{2! (r-2)!} \tag{12}$$

where $n(V(x_i, t_{int}))$ is the number of subsets of two elements $\{C^a(x_i, t_{int}), C^b(x_i, t_{int})\}$ of $\{C^1(x_i, t_{int}), ..., C^r(x_i, t_{int})\}$ such that $C^a(x_i, t_{int}) = C^b(x_i, t_{int})$. Finally, we have:

The **degree of information consistency** for stage t_{int} and input x_i is:

$$d(x_i, t_{int}) = \sum_{V(x_i, t_{int})} prob(V(x_i, t_{int})) \frac{n(V(x_i, t_{int})}{z(V(x_i, t_{int}))} \tag{13}$$

For example, consider the case depicted in Figure 6:
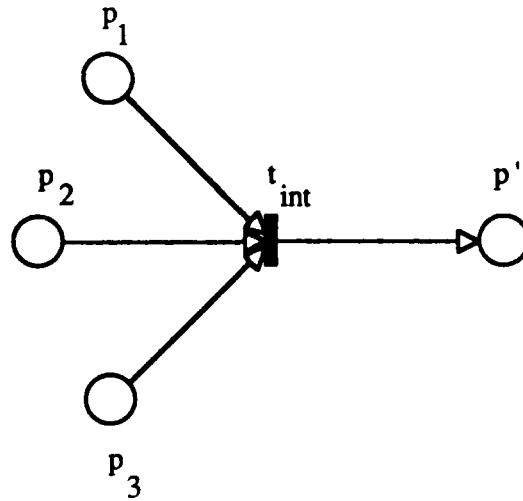
Fig. 6 Example of Degree of Information Consistency

We assume that the set of classes is $\{C^1, C^2, C^3\}$ for all three places, i.e. :

- $\mathcal{D}(p_1) = \{C^1, C^2, C^3\}$
- $\mathcal{D}(p_2) = \{C^1, C^2, C^3\}$
- $\mathcal{D}(p_1) = \{C^1, C^2, C^3\}$

Furthermore, consider the five colors, elements of X:

- $m_1 = (T_n, T_d^1, C^1)$
- $m_2 = (T_n, T_d^2, C^2)$
- $m_3 = (T_n, T_d^3, C^3)$
- $m_4 = (T_n, T_d^4, C^1)$
- $m_5 = (T_n, T_d^5, C^1)$

If M designates the marking, the three following cases are considered:

- case 1: $M(p_1)[m_1] = 1; M(p_2)[m_2] = 1; M(p_3)[m_3] = 1.$
- case 2: $M(p_1)[m_1] = 1; M(p_2)[m_4] = 1; M(p_3)[m_3] = 1.$
- case 3: $M(p_1)[m_1] = 1; M(p_2)[m_4] = 1; M(p_3)[m_5] = 1.$

Following relation (13), in case 1, the degree of information consistency is zero. In case 2, it is 1/3. In case 3, the degree of information consistency is 1. By adding the degrees of information

consistency $d(x_i, t_{int})$ for each organizational interaction t_{int} and each input x_i and weighing by the probability of having the input, one can measure the organizational degree of information consistency for the task at hand.

The **organizational degree of information consistency**, D, is defined by:

$$D = \sum_{x_i} prob(x_i) \sum_{t_{int}} d(x_i, t_{int}) \qquad (14)$$

This measure varies between 0 and 1, the value 1 corresponding to ideal information consistency of all interactions for the whole task. The next section introduces a measure of performance for synchronization.

A MEASURE OF SYNCHRONIZATION

When a decisionmaker processes an item of information, the total processing time of this item for decisionmaker DM_i consists of two distinct parts: (i) the total time $T_i{}^t$ during which the decisionmaker actually operates on the information, i.e., the total time spent by the information in the decisionmaker's algorithms; and (ii) the total time $T_i{}^p$ spent by the information in memory without being processed.

The time $T_i{}^p$ is due to two factors: (i) Information can remain in the memory of the decisionmaker until he decides to process it with the relevant algorithm. Since an algorithm cannot process two inputs at the same time, some inputs will have to remain unprocessed in memory for a certain amount of time until the relevant algorithm is available. (ii) Information can also remain in memory because the decisionmaker waits to receive data from another organization member.

An organization is not well synchronized when the decisionmakers have to wait for long periods before receiving the information that they need in order to continue their processing. Conversely, the organization is well synchronized when these lags are small.

The **sojourn time** $T_s{}^h(x_i, t_{int})$ of the token m_h, representing the input x_i in the place p_h of the preset of transition t_{int}, measures the amount of time spent by the token in the place before it is fired:

$$T_s^h(x_i, t_{int}) = T_c - T_d^h \qquad (15)$$

This quantity is zero when the firing occurs at the same time the token enters the place. Conversely, it differs from zero when the firing cannot be initiated at the same time the token enters the place. The following quantity can now be introduced:

$$S_L^{hj}(x_i, t_{int}) = T_s^h(x_i, t_{int}) - T_s^j(x_i, t_{int}) \qquad (16)$$

The quantity $S_L^{hj}(x_i, t_{int})$ measures the difference between the sojourn times of the tokens representing x_i in p_h and p_j, i.e., the difference between the lengths of time that the information sent by DM_h and DM_j to DM_k remained inactive before being processed.

When p_k represents the memory place, $S_L^{kj}(x_i, t_{int})$ will be computed for each structural place p_j. If it is positive, it implies that the token m_k has spent more time in p_k than the token m_j in p_j. If it is negative, the opposite is true. In the latter case, there is no degradation of synchronization, because DM_k is not ready to process the next task.

Let $F(x)$ denote the function defined on the set of rational numbers, Q, by:

$$\forall x \in Q, (x \geq 0) \Rightarrow (F(x) = x)$$
$$(x < 0) \Rightarrow (F(x) = 0) \qquad (17)$$

Let $INT(t_{int})$ denote the set of indices h for the structural places p_h of $Pre(t_{int})$. Then the total lag for the transition t_{int} in processing input x_i can now be defined as follows.

The total lag for transition t_{int} and the input x_i, $S(x_i, t_{int})$, is such that:

$$S(x_i, t_{int}) = \max_{h \in INT(t_{int})} (F[S_L^{kh}(x_i, t_{int})]) \qquad (18)$$

or, from (16),

$$S(x_i, t_{int}) = \max_{h \in INT(t_{int})} (F[T_s^k(x_i, t_{int}) - T_s^h(x_i, t_{int})]) \qquad (19)$$

17

Thus, $S(x_i, t_{int})$ measures the maximum of all the lags during which the decisionmaker has to wait before having all the information he needs to continue his processing. The measure S does not take into consideration the items of information for which the decisionmaker does not wait.

For instance, in the example illustrated in Figure 6, if p_2 denotes the memory place, we have:

- case 1: $S(x_i, t_{int}) = max(F(T_d^2 - T_d^1), F(T_d^2 - T_d^3))$
- case 2: $S(x_i, t_{int}) = max(F(T_d^4 - T_d^1), F(T_d^4 - T_d^3))$
- case 3: $S(x_i, t_{int}) = max(F(T_d^4 - T_d^1), F(T_d^4 - T_d^5))$

On the basis of this definition, it is possible to define two measures. First, we denote by $\mathcal{A}$ the set of all interactional transitions. Furthermore, $\mathcal{A}(k)$ denotes the set of all interactional transitions executed by decisionmaker DM_k.

The measure of synchronization between decisionmaker DM_k and the rest of the organization, S_k, is defined as:

$$S_k = \sum_{x_i} prob(x_i) \sum_{t_{int} \in \mathcal{A}(k)} S(x_i, t_{int}) \qquad (20)$$

It is the expected value of the sum of the maximum lags for the interaction stages executed by decisionmaker DM_k for the inputs x_i.

The measure of synchronization for the organization, S_T, is given by:

$$S_T = \sum_{x_i} prob(x_i) \sum_{t_{int} \in \mathcal{A}} S(x_i, t_{int}) \qquad (21)$$

It is the expected value of the sum of the maximum lags over the overall decision-making process for the inputs x_i.

On the one hand, the measures S_k, for each k, and S_T achieve their best values when they are zero. On the other hand, there is no upper bound on the values taken by these measures; they grow to infinity if a deadlock occurs. Since each interactional transition t_{int} belongs to one decisionmaker, and one only, the following relation holds:

$$S_T = \sum_k S_k \qquad (22)$$

Thus, one can compute the contribution of each individual decisionmaker DM_k to the total synchronization measure S_T for the organization by taking the ratio S_k/S_T.

Example: Consider the two-person hierarchical organization shown in Figure 7. We assume that the processing times of the various stages are independent of the input x_i. Therefore, $S_L(x_i, t_{int})$ is written $S_L(t_{int})$. The parameters T_s^4, T_s^5, T_s^6, and T_s^8 denote the sojourn times of tokens in places p_4, p_5, p_6 and p_8, respectively. The internal transitions IF_2 and CI_1 have only two input places. It follows that:
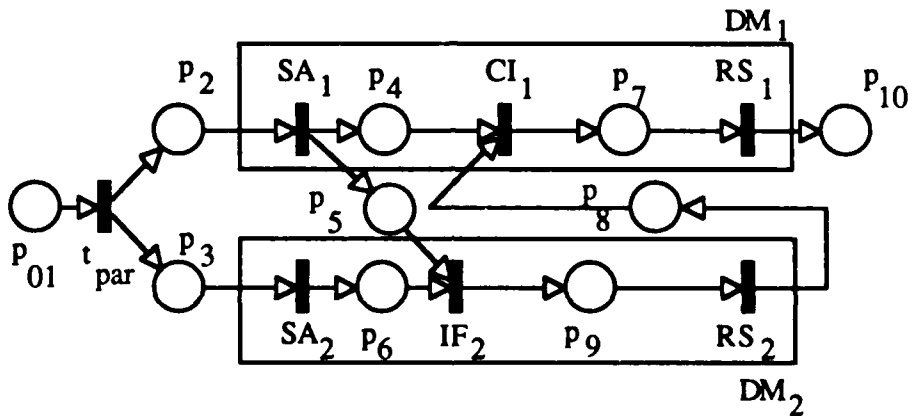
Fig. 7 Synchronization of Two-Person Organization

- $S_L(t_{IF2}) = F(T_s^6 - T_s^5)$
- $S_L(t_{CI1}) = F(T_s^4 - T_s^8)$

Then, the measures of synchronization S_T, S_1 and S_2 are:

- $S_1 = S_L(t_{CI1}) = F(T_s^4 - T_s^8)$
- $S_2 = S_L(t_{IF2}) = F(T_s^6 - T_s^5)$
- $S_T = S_L(t_{IF2}) + S_L(t_{CI1}) = F(T_s^6 - T_s^5) + F(T_s^4 - T_s^8)$

Let $f(t)$ denote the firing time of transition t for all inputs. Then two cases are considered:

19

case 1: $f(t_{SA1}) = 2, f(t_{SA2}) = 1, f(t_{IF2}) = 2, f(t_{RS2}) = 2, f(t_{CI1}) = 2, f(t_{RS1}) = 2$
case 2: $f(t_{SA1}) = 2, f(t_{SA2}) = 2, f(t_{IF2}) = 2, f(t_{RS2}) = 2, f(t_{CI1}) = 2, f(t_{RS1}) = 2$

The results for the measures S_1, S_2 and S_T are:
- case 1, $S_T = 5$; $S_1 = 1$; $S_2 = 4$.
- case 2, $S_T = 4$; $S_1 = 0$; $S_2 = 4$.

When the processing times of SA_1 and SA_2 are not equal (case 1), the synchronization measure S_T is worse than when they have equal processing times (case 2). It should also be noted that the processing delays in the two cases are equal, i.e., $T = 10$, where T is the total delay. It follows that if DM_2 takes more time to process an input, synchronization is improved without affecting the total processing delay.

In this example the measure of synchronization can be zero if, and only if, SA_1 and SA_2 have the same processing times, and IF_2 and RS_2 have zero processing times. The latter conditions are not realistic, which implies that the synchronization will never be zero.

APPLICATION

In this section, an example is presented where the lack of coordination between the different decisionmakers leads to a degradation of the organizational performance. This degradation can have various causes, but they all relate to the concept of a team. A team has been defined as being a group of persons who have a common goal, have the same interests and beliefs, and have activities that must be coordinated (Grevet, 1988). The fact that different decisionmakers have distinct interests or beliefs can lead them to adopt activities that are not coordinated with respect to the task at hand.

The organization consists of two decisionmakers who receive information for a common task. The commander DM_2 assesses the data that he receives from the environment by using always the same algorithm. In the same way, the subordinate DM_1 assesses the input from the environment with one algorithm. Then, he sends some information resulting from this assessment to his commander. The latter fuses his own result with this information and, on this basis, produces a command by using always the same algorithm. In turn, this command is sent to the subordinate DM_1. Eventually, DM_1 is responsible for producing a response on the basis of the command that he receives and of the results of his own assessment. The Petri Net model of

20

this organization has already been presented in Figure 7. It is shown in Figure 8 with the resource places present.

Since both decisionmakers have only one algorithm in each stage of their process, there is only one organizational strategy. Thus, the information can follow only one flow path. Each decisionmaker needs the information sent by the other in order to complete his processing. In accordance with the model of coordination developed in this study, the Information Fusion and Command Interpretation stages require that the data fused originate from the same input. The organization will be perfectly synchronized if DM_1 and DM_2 never wait for the information from the other decisionmaker in these stages.
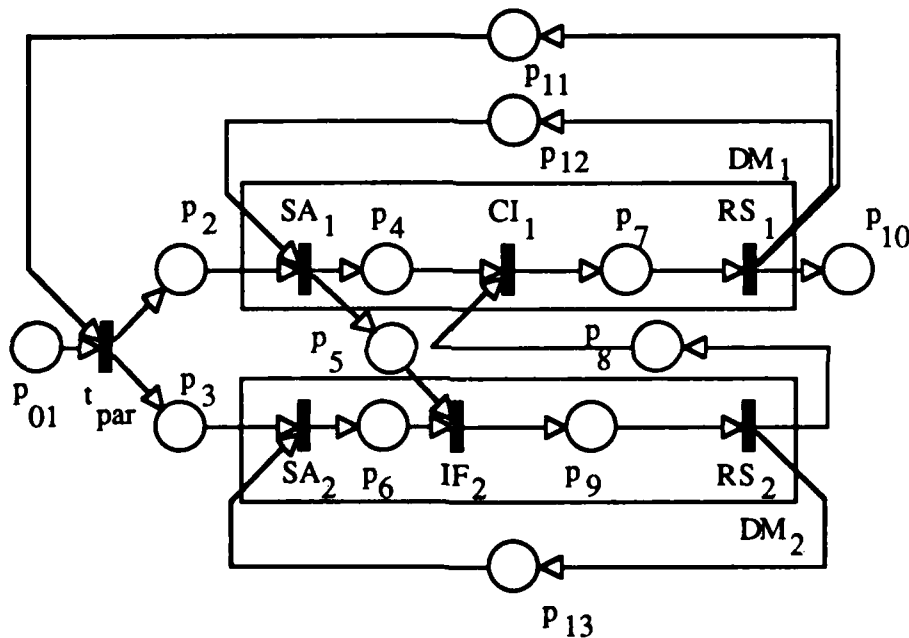
Fig.8 Petri Net Model of Two-person Organization with Resource Places

It will be assumed that, in all stages but the SA stage, the decisionmakers select the data with the LFIFO rule with priority given to the items of information that are in their memory places. Thus, for IF_2, RS_2, CI_1, and RS_1, tokens are fired in the order with which they enter the memory place of their preset. However, different conditions for the SA stages will be considered. Before having assessed any of the inputs that they have received and that they must process, the decisionmakers may have to discriminate between them because they cannot perform their assessment on all of them at the same time; only one input can be assessed at a time.

21

Depending on the characteristics of the data that they receive from the environment, the decisionmakers can use different rules to perform this selection: two cases will be investigated:

- the inputs have not been preprocessed by any decision-aid: in this case, the decisionmakers have no information concerning the nature of the inputs and must use rules that are based on attributes independent from the characteristics of the inputs, e.g., time of entry in the organization.

- the inputs have been preprocessed by a decision-aid that aggregates them in classes, or zones of indifference (Chyen and Levis, 1985). In this case, the decisionmakers have some information concerning the nature of the inputs and can use rules that assign priorities to these different classes.

The processing times, measured in some time unit, of the various stages are presented below; t_{par} denotes the partitioning stage, as introduced in Figure 1.

	t_{par}	SA_1	SA_2	CI_1	IF_2	RS_1	RS_2
Time:	1	10	10	10	10	10	10

The scenario corresponds to the infinite queue of inputs, i.e., to the case where the organization always uses all its resources.

case 1: The initial marking of the resource places is:
$M^0(p_{11}) = 4$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.
Both SA_1 and SA_2 use the LFIFO rule.

case 2: The initial marking of the resource places is:
$M^0(p_{11}) = 4$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.
SA_1 uses the LFIFO rule whereas SA_2 uses the LLIFO rule.

case 3: The initial marking of the resource places is:
$M^0(p_{11}) = 4$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.
SA_1 uses the LLIFO rule; SA_2 uses the LFIFO rule.

case 4: The initial marking of the resource places is:
$M^0(p_{11}) = 4$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.
Both SA_1 and SA_2 use the LLIFO rule.

The time of entry of an input in the organization, T_i, is the time at which the sensors begin to process it, i.e., in Petri Net formalism, the time at which the transition t_{par} fires. The time of leaving from the organization, T_o, is the time at which the organizational response is obtained, i.e., in Petri Net formalism, the time at which a token appears in the sink place. The delay, T, is the difference T_o-T_i. The measure of synchronization S is obtained for each token; it is the sum of the values of the measure for transitions IF_2 and CI_1. The values were obtained using the simulation system MIT/SIM (Grevet et al., 1988). The results for T_i, T_o, T as well as for the synchronization S for the first ten inputs which enter the net in each of these four cases are shown in Table 1.

TABLE 1. Synchronization and Delay - Cases 1 to 4

input #	case 1				case 2				case 3				case 4			
	T_i	T_o	T	S	T_i	T_o	T	S	T_i	T_o	T	S	T_i	T_o	T	S
1	0	51	51	20	0	51	51	20	0	51	51	20	0	51	51	20
2	1	61	60	20	1	-	-	-	1	-	-	-	1	-	-	-
3	2	101	99	40	2	101	99	40	2	101	99	20	2	101	99	40
4	3	111	108	40	3	151	148	110	3	131	128	90	3	61	58	20
5	51	151	100	40	51	201	150	90	51	161	110	30	61	151	90	40
6	61	161	100	40	101	251	150	90	101	191	90	30	101	201	100	40
7	101	201	100	40	151	301	150	90	131	221	90	30	151	251	100	40
8	111	211	100	40	201	351	150	90	161	251	90	30	201	301	100	40
9	151	251	100	40	251	401	150	90	191	281	90	30	251	351	100	40
10	161	261	100	40	301	451	150	90	221	311	90	30	301	401	100	40

The results obtained in case 1 are the same as in the case where the tokens have no identity. The steady-state of the process is K-periodic (Hillion and Levis, 1987) with a period of one. It is reached after the sixth input and is characterized by a constant delay and synchronization. The same conclusions can be drawn in case 2, case 3 and case 4: all three processes are K-periodic with a period of one. In case 2, the steady-state is reached after the fifth input whereas it is reached after the sixth input in case 3 and case 4. one can see that the three tokens with attribute T_n = 1 are blocked in places p_3, p_4 and p_5, respectively. The processing of the corresponding input is **blocked** as shown in Table 1. This happens because there are always two tokens in the

23

input places of SA_1 and SA_2 where the LFIFO and LLIFO rules are used.

In the steady-state, the delays for each input are identical in case 1 and case 4. In case 2, this delay increases by 50 percent. In case 3, the delay is reduced by 10 percent. However, since in the situations where a LLIFO rule is used the processing of one input is blocked, the delay for this input is infinite and the organization can use only three resources out of four for the other inputs. Thus, the thoughput rates decrease . Moreover, for the organizations under consideration in this study, a response must be given to each input in a timely manner. Thus, when the input represents a threat for which a response must be provided in a certain window of opportunity (Cothier and Levis, 1986), the LLIFO rule will degrade considerably the accuracy and timeliness of the organization.

In case 1, the synchronization of the organization is equal to 40 units of time in the steady-state. In case 2, it is equal to 90 and, so, degrades considerably. In case 3, the synchronization is equal to 30 units of time in the steady-state. It represents therefore an improvement with respect to case 1. In case 4, the synchronization in the steady-state is the same as in case 1.

Nevertheless, one must consider also the individual tokens that are blocked during the processing. In case 2 and case 3, the synchronization for the second token degrades considerably with respect to case 1. Indeed, if DM_1 uses the LFIFO rule and DM_2 the LLIFO rule, the item of information for which the process is blocked is in the input place of the SA stage of the latter, whereas it has been assessed by DM_1 and is in the memory place of his CI stage. Thus, the measure S for this input is infinite. The same situation occurs when DM_1 uses the LLIFO rule and DM_2 the LFIFO rule but, in this case, the degradation of the synchronization is due to the fact that DM_2 waits indefinitely in the IF stage for the data from DM_1 to arrive. In case 4, the second input is also blocked, but the two corresponding tokens remain in the input places of SA_1 and SA_2: it implies that none of the decisionmakers will wait for the data from the other member for this input. From this standpoint, the synchronization of the activities for this input does not degrade.

The processing of the inputs in these four cases took place for a configuration in which there were four organizational resources and two resources for each decisionmaker. The following cases examine a situation in which the organizational resources are increased by one unit. The scenario still corresponds to the infinite queue of inputs and the processing times of the protocols are not changed.

case 1': The initial marking of the resource places is:

$M^0(p_{11}) = 5$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.

Both SA_1 and SA_2 use the LFIFO rule.

case 2': The initial marking of the resource places is:

$M^0(p_{11}) = 5$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.

SA_1 uses the LFIFO rule whereas SA_2 uses the LLIFO rule.

case 3': The initial marking of the resource places is:

$M^0(p_{11}) = 5$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.

SA_1 uses the LLIFO rule whereas SA_2 uses the LFIFO rule.

case 4': The initial marking of the resource places is:

$M^0(p_{11}) = 5$; $M^0(p_{12}) = 2$; $M^0(p_{13}) = 2$.

Both SA_1 and SA_2 use the LLIFO rule.

Table 2 provides the results for T_i, T_o and T as well as for the synchronization S for the first ten inputs which enter the net in each of these four cases.

TABLE 2 Synchronization and Delay - Cases 1' to 4'

input #	case 1'				case 2'				case 3'				case 4'			
	T_i	T_o	T	S	T_i	T_o	T	S	T_i	T_o	T	S	T_i	T_o	T	S
1	0	51	51	20	0	51	51	20	0	51	51	20	0	51	51	20
2	1	61	60	20	1	-	-	-	1	-	-	-	1	-	-	-
3	2	101	99	40	2	-	-	-	2	-	-	-	2	-	-	-
4	3	111	108	40	3	-	-	-	3	-	-	-	3	101	98	20
5	4	151	147	40	4	-	-	-	4	-	-	-	4	61	57	40
6	51	161	110	40	51	-	-	-	51	-	-	-	61	151	90	40
7	61	201	140	40									101	201	100	40
8	101	211	110	40									151	251	100	40
9	111	251	140	40									201	301	100	40
10	151	261	110	40									251	351	100	40

In case 1', the process is K-periodic of period 2. In the steady-state, the thoughput rate and the synchronization are identical to case 1. In case 4', the process has a period equal to one. The synchronization and thoughput rate are identical to case 4. However, the second and third inputs remain blocked in the input places of the SA transitions. Thus, the organization can use only three out of its five resources to process the remaining inputs.

In case 2' and case 3', the performance of the organization is totally degraded by the fact that the whole process is blocked. The execution has reached a **deadlock**, i.e., no transition can fire. As it is shown in Table 2, five inputs remain in the organization which cannot produce a response for any of them. Because DM_1 and DM_2 do not use the same strategies, the two items of information sent by DM_1 to DM_2 after the Situation Assessment stage do not correspond to the inputs that DM_2 is processing.

Thus, since DM_2 has to wait for the information that he needs in order to proceed and since DM_1 has to wait for the commands from DM_2 to arrive, the activities of both decisionmakers are blocked. This illustrates a situation where the lack of coordination leads to a severe degradation of the effectiveness of the organization.

Figure 9 shows the Petri Net representation of the state of the organization when the deadlock occurs. Places p_5 and p_6 contain tokens that do not have the same attribute T_n, and consequently, rule 1 of enablement of transition IF_2 is not satisfied. Since the resource places p_{12} and p_{13} are empty, transitions SA_1 and SA_2 cannot fire and the tokens that have the same attribute T_n as the tokens in p_6 are blocked in p_2.

This type of situation would never occur if SA_1 and SA_2 used the LFIFO rule for the sequencing of the inputs: indeed, the interactional transitions would always fire as soon as the places of their preset contain a token since these tokens would necessarily have the same attribute T_n.

CONCLUSIONS

The purpose of this study was to investigate the concept of coordination in information processing and decision-making organizations. The framework of the research is a quantitative methodology for evaluating alternative organizational structures. The concept of coordination was defined as relating to the extent to which the decisionmakers constitute a team; to the consistency of the information exchanged by the different organization members; and the synchronization of the various activities. The latter bears directly on the dynamics of the
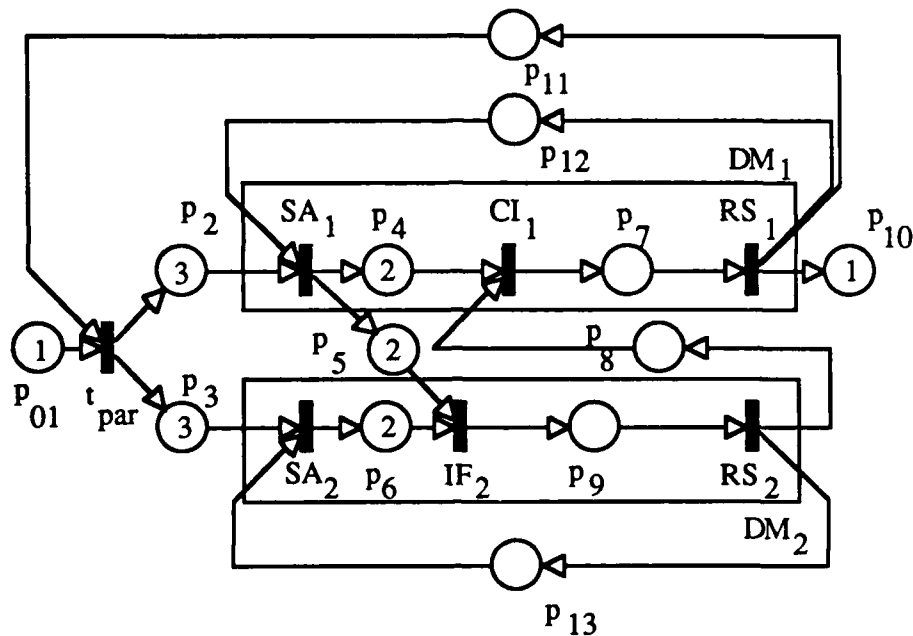
Fig.9 Two-Person Hierarchical Organization with Deadlock

decision-making process. A decision-making organization is perfectly synchronized for the task at hand if none of its members waits for the information that he needs at any stage of the process. If it is not the case, the value of information when it is actually processed may have decreased, leading to a degradation of the organizational effectiveness. The consistency of information shows the extent to which different pieces of information can be fused without contradiction.

The modeling of processes that require coordination has been developed using the basic model of the single interacting decisionmaker refined through the use of the Predicate Transition Net formalism. In particular, tokens representing symbolic information carriers have been differentiated on the basis of three attributes which account for characteristics that decisionmakers can use to discriminate between various data.

The protocols of interactions between organization members model the fact that they must refer to the same input when they fuse data. Different strategies for selecting the information to process have been introduced, e.g., FIFO or priority order between classes of data.

The evaluation of the coordination is based on a characterization of the firing of interactional transitions in the Predicate Transition Net model developed. Furthermore, two measures are

27

introduced in order to perform a quantitative evaluation of the coordination of decision-making processes, i.e., the degree of information consistency and the measure of synchronization.

A simulation system for Petri Nets has been developed. Ordinary Petri Nets, Timed Petri Nets and Predicate Transition Nets can be simulated. The dynamics of different decision-making processes can be investigated by simulating the execution of the corresponding Petri Net models. Quantitative results concerning the processing delays and the synchronization of the activities have been obtained.

REFERENCES

Brams, G.W., 1983, *Réseaux de Petri: Théorie et Pratique, T.2, Modélisation et Application*, Masson, Paris.

Chyen, G. H-L., and A.H. Levis, 1985, "Analysis of Preprocessors and Decision Aids in Organizations," *Proc. IFAC/IFIP/IFORS/IEA Conference on Analysis, Design and Evaluation of Man-Machine Systems*, Varese, Italy.

Cothier, P.H., and A.H. Levis, 1986, "Timeliness and Measures of Effectiveness in Command and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-16, No. 6.

Genrich, H.J., and K. Lautenbach, 1981, "Systems Modeling with High-Level Petri Nets," *Theoretical Computer Science*, No.13, pp.109-136.

Grevet, J.-L., 1988, "Decision Aiding and Coordination in Decision-making Organizations," S.M. Thesis, LIDS-TH-1737, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Grevet, J.-L., L. Jandura, J. Brode, and A. H. Levis, 1988, "Execution Strategies for Petri Net Simulations," *Proc. 12th IMACS World Congress on Scientific Computation*, Paris, France

Hillion, H.P., and A.H. Levis, 1987, "Timed Event-Graph and Performance Evaluation of Systems," *Proc. 8th European Workshop on Applications and Theory of Petri Nets*, Zaragoza, Spain.

Jin, V.Y., A.H. Levis, and P.A. Remy, 1986, "Delays in Acyclical Distributed Decision-Making Organizations," *Proc. IFAC Symposium on Large Scale Systems: Theory and Applications*, Zurich, Switzerland.

Levis, A.H., 1984, "Information Processing and Decision-Making Organizations: a Mathematical Description," *Large Scale Systems*, No. 7, pp. 151-163.

Tabak, D., and A.H. Levis, 1985, "Petri Net Representation of Decision Models," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 6.

END

DATE

FILMED

5-88

DTIC