JULY 1988 LIDS-P-1789

COORDINATION IN ORGANIZATIONS WITH DECISION SUPPORT SYSTEMS*

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 model, analyze, and evaluate coordination in organizations with decision support systems 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: information consistency and 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, through an example, the dynamics of such organizations and for analyzing how a decision support system can alter the coordination in an organization.

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the Office of Naval Research under Contract No. N00014-84-K-0519 (NR 649-003).

COORDINATION IN ORGANIZATIONS WITH DECISION SUPPORT SYSTEMS*

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 model, analyze, and evaluate coordination in organizations with decision support systems 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: information consistency and 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, through an example, the dynamics of such organizations and for analyzing how a dacision support system can alter the coordination in an organization.

INTRODUCTION

The decisionmaking by organization members implementing the command and control process must be coordinated in order to improve their effectiveness. Decision aids, which are part of the C^3 systems, aim at increasing the ability of decisionmakers to perform their mission effectively. By offering faster processing capabilities as well as access to databases, they may help the organization members to achieve the requirements of the mission. However, decision aids also increase the possible alternatives among which to choose in order to process information, and in so doing, modify the nature of the decisionmakers' activities. In this context, it is important *to evaluate the extent to which decision aids,* and more particularly decision support systems (DSS), *can alter the coordination of the various decision-making processes.*

The framework used to address this problem is the quantitative methodology (Levis, 1984; 1988) for the analysis and evaluation of alternative organizational structures. 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: information consistency and synchronization. The latter measure relates to the value of information when the decisionmakers actually process it.

A generic model of a decision-maker interacting with a DSS is presented. The focus is on the architecture of the system and on the different system components that the decision-maker can access. DSS's have become an increasingly important part of the military Command, Control and Communications (C^3) systems (Waltz and Buede, 1986). In this context, the DSS's, also called battle management systems, automate the fusion of data concerning the tactical situation and the quantitative evaluation of alternative courses of action.

Decision aids are defined as any technique or procedure that restructures the methods by which problems are analyzed, alternatives developed and decisions taken. Keen and Scott Morton (1978) emphasize that decision support systems, a particular form of decision aids, have specific advantages:

"(i) *the impact is on decisions in which there is sufficient structure for computer and analytic aids to be of value, but where decisionmakers' judgment is essential.*

(ii) *the payoff is in extending the range and capability of decisionmakers' decision processes to help them improve their effectiveness.*

(iii) *the relevance for decisionmakers is the creation of a supportive tool under their own control, which does not attempt to automate the decision process, predefine objectives, or impose solutions."*

Thus, DSS's do not automate the decisionmaking process, but must facilitate it. When confronted with a particular task, the decisionmaker keeps the choice of performing it by himself or requesting information from the DSS. This selection depends on the reliability of the DSS or, more exactly, on the extent to which the organization members rely on it.

The evaluation of the effectiveness of a decisionmaking organization consisting of human decision-makers aided by a DSS is a complex issue: many interrelated factors affect the effectiveness of the overall system, e.g., the limited information processing capacities of the decision-makers, the hardware and software characteristics of the DSS, or the extent to which the organization members use and rely on the decision aid. One important question is to know whether or not the overall organization, when aided by the DSS, is more effective in fulfilling its mission.

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the Office of Naval Research under Contract No. N00014-84-K-0519 (NR 649-003).

Earlier work has assessed the impact of preprocessors (Chyen and Levis, 1985; Weingaertner and Levis, 1987) and databases (Bejjani and Levis, 1985) on the workload of the decision-makers. However, it seems necessary to measure the extent to which the DSS can affect the coordination of the various decisionmakers who use it. Indeed, the introduction of a DSS in an organization can lead either to an improvement or to a degradation of its cohesiveness, depending on the functionality and capabilities of the DSS, as well as on the perception of and access to the DSS that the decisionmakers have.

Simulation of Predicate Transition Nets is introduced to investigate the dynamics of decisionmaking processes - especially phenomena not captured by analytically tractable models, such as the use of different protocols by different decisionmakers. An example demonstrates that decision aids can degrade the coordination of decision-making organizations by affecting the dynamics of the activities and by increasing the number of alternatives for processing information.

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 Petri Nets terms, there exists a source place, p_{so}, and a sink place, p_{sk}. A resource place, p_{rs}, is introduced to model the limited organizational resources. A transition t_{par} models the partitioning of the input from the single source into inputs received by different organization members or C^3 system components. 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).

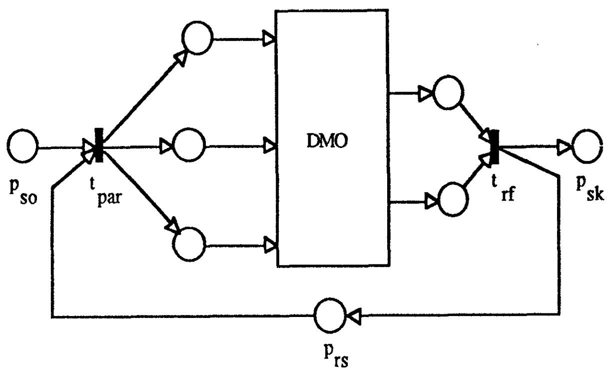

Fig. 1 Petri Net of Interactions between a Decisionmaking Organization and the Environment

The source p_{so} generates single tokens that arrive sequentially and are marked with the arrival time. The task is modeled by the alphabet $\mathcal{X} = \{x_1,..., x_n\}$ and a probability distribution prob($x = x_i$), denoted by prob(x_i) defined on $\mathcal{X}$. The set of subsets of $\mathcal{X}$ is denoted by $\Pi(\mathcal{X})$; then:

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

where $\emptyset$ denotes the empty set.

A clock is used to mark the instants T_c at which the process is observed. In accordance with the formalism of Timed Petri Nets, this clock provides 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 tokens can build in the places of 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) Since inputs originate from a single source, one at a time, the attribute T_n corresponds to the time at which the input represented by this token entered the organization.

(ii) Since in stochastic timed Petri Nets, the firing of any token takes an amount of time that depends on the processing time of the corresponding transition. One can assign to any token in a place p the time T_d at which it entered this place.

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

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.

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

The marking of PN is defined as follows: For each place p, M(p) assigns to each value of the variable associated with p a non-negative integer number which 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 triplet (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 ¢ and $M(p)[¢]$ can be denoted simply by $M(p)$.

The following example (Figure 2) illustrates these definitions:

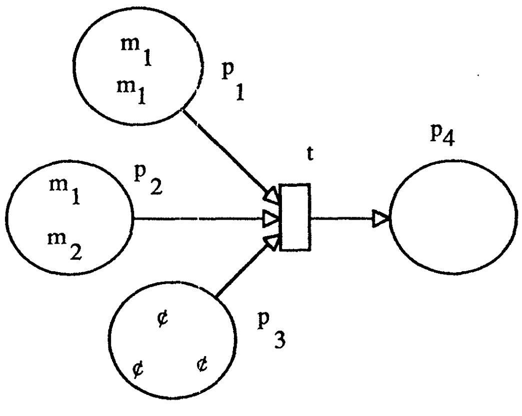

Fig. 2 Example of Marking

In this example, the following relations hold:

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

The firing of a transition t 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 Marked Graph and, so, each place has only one input transition which produces in each of its output places only one token per firing.

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

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 3.

When the fusion of data is performed by a decisionmaker, *only one of the places* $p_1, \ldots, p_r$ is a memory place. We denote it by p_k. Any rule of enablement can be introduced at this point. Let M denotethe marking of the net. Then, two possible rules are:

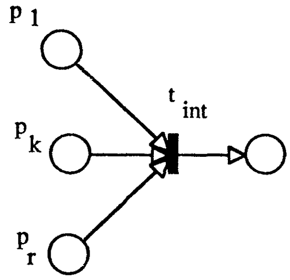

Fig. 3 Petri Net Model of Interaction with Fusion of Data

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

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. This results from the fact that memory and structural places contain only tokens of the (T_n, T_l, 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 2: t_{int} is enabled if and only if rule 1 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 2 models the interactions where decisionmakers wait for information from other parts of the organization but 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. It means that the attributes T_n^i of the colors $m_1, \ldots, m_r$ must have the same value.

Token Selection

The problem of token selection arises since the tokens are distinguishable. These rules operate on the tokens of the input places that enable the transition t. This is illustrated by the example of Figure 4 where rule 2 of enablement applies.

Suppose that:

- $m_1 = (T_n^1, T_d^1, C^1)$; $m_1' = (T_n^1, T'_d{}^1, C'^1)$;
  $m_1'' = (T_n^1, T''_d{}^1, C''^1)$.
- $m_2 = (T_n^2, T_d^2, C^2)$; $m_2' = (T_n^2, T'_d{}^2, C'^2)$;
  $m_2'' = (T_n^2, T''_d{}^2, C''^2)$.

Since the enabling condition is that the tokens have the same arrival time T_n, it follows that the transition t is enabled by both sets $\{m_1, m_1', m_1''\}$ and $\{m_2, m_2', m_2''\}$. Therefore, a rule must exist to decide which token set 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. Therefore, before applying the rule, it is necessary to decide in which place p the selection will be done. One can see on the example of Figure 4 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. Different strategies can be applied to choose the place. 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. 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 of Fig. 4, 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.

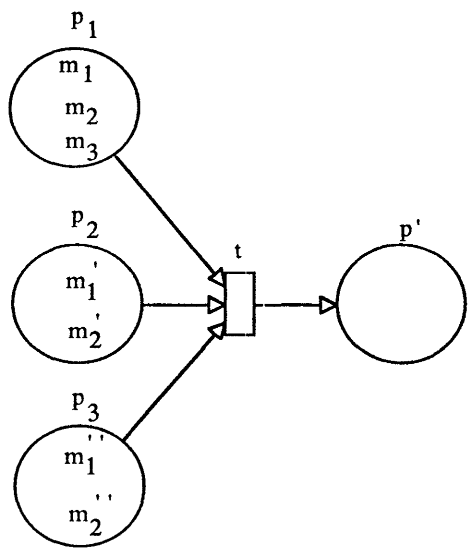

Fig. 4 Token Selection

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.

Four types of rules of selection PS(t) can be considered:

(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:

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.

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

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.

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.

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.

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.

CHARACTERIZATION OF COORDINATION

The Petri Net representation of the transitions considered in this section is shown in Figure 2. 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 relations are defined:

Ψ_1 is a binary relation defined by :

$$((x, y, z)\ \Psi_1\ (x', y', z')) \Leftrightarrow ((x = x') \text{ and } (y \leq y')) \quad (2)$$

Ψ_2 is a binary relation defined by:

$$((x, y, z)\ \Psi_2\ (x', y', z')) \Leftrightarrow ((x = x') \text{ and } (z = z')) \quad (3)$$

Ψ_3 is a binary relation definedby :

$$((x, y, z)\ \Psi_3\ (x', y', z')) \Leftrightarrow (((x, y, z)\ \Psi_1\ (x', y', z')) \text{ and } ((x, y, z)\ \Psi_2\ (x', y', z'))) \quad (4)$$

The relation Ψ_3 defines an order relation on the set X. Let $m_1, ..., m_r$ denote the elements of X which represent the colors of the r tokens removed from places $p_1, ..., p_r$, respectively, by transition t_{int}; let m_k denotes the color of the token removed from the memory place p_k. Furthermore, each color m_i corresponds to some triplet (T_n^i, T_d^i, C^i).

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

$$\forall i \in \{1, ..., r\} \quad (T_n^i, T_d^i, C^i)\ \Psi_1\ (T_n^k, T_d^k, C^k) \quad (5)$$

This definition allows to discriminate between firings that are synchronized and firings in which one or several tokens m_i arrive in their respective places later than m_k in p_k.

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

$$\forall (i, j) \in \{1, ..., r\} \times \{1, ..., r\},\ (T_n^i, T_d^i, C^i)\ \Psi_2\ (T_n^j, T_d^j, C^j) \quad (6)$$

i.e., the data fused by DM_k are consistent, if they correspond to the same class C. 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, the 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 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.

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 3. At each transition t_{int}, the decisionmaker DM_h associates a class C^h with each input x_i; this class is denoted by $C^h(x_i, t_{int})$ and belongs to $\mathfrak{D}(p_h)$, a partition of the alphabet $\mathbf{X}$, 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 a particular stage are provided with the same set of classes; therefore, it is assumed that:

$$\forall\ (i, j) \in \{1, ..., r\} \times \{1, ..., r\},\quad \mathfrak{D}(p_i) = \mathfrak{D}(p_j) \qquad (7)$$

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 the quantities $C^1(x_i,t_{int}),..., C^r(x_i, t_{int})$ 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 $[\Pi^*(\mathbf{X})]^r$. 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:

$$z(V(x_i, t_{int})) = \binom{r}{2} = \frac{r!}{2!\,(r-2)!} \qquad (8)$$

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:

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

$$d(x_i, t_{int}) = \sum_{V(x_i, t_{int})} prob(V(x_i, t_{int}))\, \frac{n(V(x_i, t_{int}))}{z(V(x_i, t_{int}))} \qquad (9)$$

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 that input, one can measure the **organizational degree of information consistency, D**, for the task at hand:

$$D = \sum_{x_i} prob(x_i) \sum_{t_{int}} d(x_i, t_{int}) \qquad (10)$$

This measure varies between 0 and 1, with 1 being the ideal information consistency of all interactions for the whole task.

SYNCHRONIZATION

The total processing time of an item of information for decisionmaker DM_i consists of two parts: (i) the total time T_i^t during which the decisionmaker actually operates on the information; and (ii) the total time T_i^p spent by the information in memory prior to 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 has to wait 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:

$$T_s^h(x_i, t_{int}) = T_c - T_d^h \qquad (11)$$

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:

$$S_L^{hj}(x_i, t_{int}) = T_s^h(x_i, t_{int}) - T_s^j(x_i, t_{int}) \qquad (12)$$

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:

$$\forall\ x \in \mathbf{Q},\ (x \geq 0) \Rightarrow (F(x) = x)$$
$$(x < 0) \Rightarrow (F(x) = 0) \qquad (13)$$

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, $S(x_i, t_{int})$, can now be defined as follows.

$$S(x_i, t_{int}) = \max_{h \in INT(t_{int})} \left(F[S_L^{kh}(x_i, t_{int})] \right) \quad (14)$$

or, from (12),

$$S(x_i, t_{int}) = \max_{h \in INT(t_{int})} \left(F [T_s^k(x_i, t_{int}) - T_s^h(x_i, t_{int})] \right) \quad (15)$$

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.

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

$$S_k = \sum_{x_i} prob(x_i) \sum_{t_{int} \in \mathcal{A}(k)} S(x_i, t_{int}) \quad (16)$$

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:

$$S_T = \sum_{x_i} prob(x_i) \sum_{t_{int} \in \mathcal{A}} S(x_i, t_{int}) \quad (17)$$

It is the expected value of the sum of the maximum lags over the overall decisionmaking 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:

$$S_T = \sum_k S_k \quad (18)$$

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.

A MODEL OF A DECISION SUPPORT SYSTEM

The amount of data that must be handled by C^3 systems for a typical mission is very large. For example, the antisubmarine warfare (ASW) mission requires the surveillance of a vast area where multiple sensors gather information on the environment. The typical information requirements (Waltz and Buede, 1986) are the following:

- the surveillance area covers 2000 × 2000 km.
- the sensor systems consist of 4 surveillance aircraft, 12 ASW ships, and 2 ASW submarines.
- the number of targets can be as high as 200.
- the number of reports per minute ranges from 1000 to 5000.

In this context, there is clear need for a computerized decision aiding system for the coordination of the activities of the various decisionmakers. Such a decision support system can modify the activities of a decision-maker because the latter has to consider the possibility of querying the system (Weingaertner and Levis, 1987). For each input and each stage of his internal decisionmaking process, the decisionmaker must make **meta-decisions** concerning the use of the DSS. These meta-decisions are of three types:

- the DM does not query the DSS and performs all processing by himself.
- the DM sends a query to some component of the system and relies totally on the response.
- the DM sends a query to some component of the system, but compares its response with his own assessment.

When several decisionmakers use a DSS for a common task, the DSS can increase or decrease the coordination of the group.

It is not possible to define a generic type of decision support system because DSS's are, in general, application-oriented and, therefore, quite specific to the organizations which use them and to the task that must be performed. The following model takes into account several capabilities and characteristics which are common to most of the real systems. In particular, it takes into consideration the fact that most real DSS's have facilities shared by several users and facilities accessed individually. From a physical standpoint, the DSS consists of a mainframe shared by the organization and which is accessed by the decision-makers through remote intelligent terminals and a communication network. The terminals are called "intelligent" to the extent that they provide the users with the opportunity to do local processing without querying the central system.

The DSS provides a multiple-access capability to the decisionmakers who can query it in parallel. Several databases are stored in the mainframe so that a decisionmaker can get information concerning the state of the environment as well as the possible responses that he can give to any input; it implies that the decisionmaker can query the database both in his Situation Asessment stage and in his Response Selection stage.

The applications implemented on the system do not embody any heuristic and do not develop alternative solutions. They implement models and doctrines well known to the decision-makers. Consequently, the processing of any particular task by DM_i involves some or all of the four essential components described in Figure 5: the decisionmaker DM_i, the intelligent terminal i that he uses, the communication network, and the mainframe.

For each of the three paths illustrated above, the amount of time that it takes to process the input for each internal stage of DM_i depends on several factors:

(i) in path 1, the decision-maker processes the information by himself; this takes an amount of time equal to the processing time of the corresponding protocol.

(ii) in path 2, the decision-maker uses only the intelligent terminal. The total amount of time taken by this operation corresponds to the sum of the following delays:

- time spent by the decision-maker to query the terminal;
- time spent by the terminal to process and display the information;
- time spent by the decision-maker to assess the response.

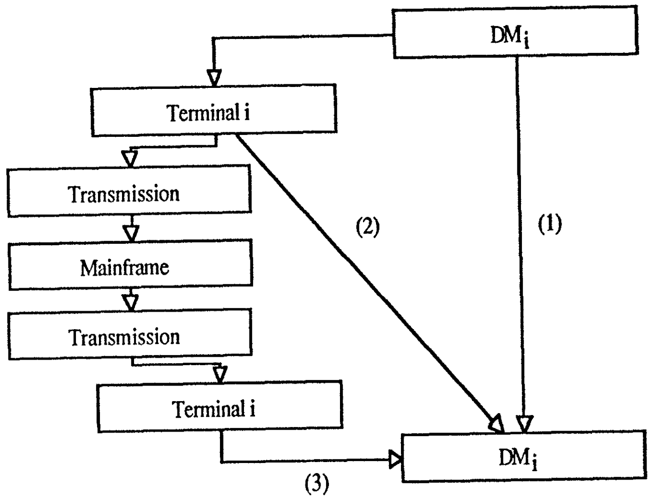

Fig. 5 DM_i Interacting with the DSS

(iii) in path 3, the decision-maker uses the terminal as a dumb terminal to query the mainframe. The total delay of this operation is the sum of the following delays:

- time spent by the decision-maker to query the mainframe;
- time spent by the terminal to access the network;
- time of transmission to the mainframe;
- time spent by the mainframe to recognize the query and initiate the processing;
- time spent by the mainframe to process the information;
- time spent by the mainframe to access the network;
- time of transmission to the terminal;
- time spent by the terminal to display the information;
- time spent by the decision-maker to assess the response.

The use of the mainframe involves the execution of operations that can take an amount of time which depends to a large extent on the physical configuration of the system. In particular, the delay of transmission through the communication network can vary over a wide range according to the specific route use which depends, in turn, on the origin and the destination. Furthermore, a query to the mainframe may be much more subject to errors due to noise and the distortion in the transmission than a query to the intelligent terminal.

Petri Net Model of Decisionmaker Aided by a DSS

The Petri Net model of a decisionmaker DM aided by a DSS is given in Figure 6. This model represents the different information flow paths that exist when a DM interacts with the DSS at any internal stage of his decisionmaking process. Figure 6 illustrates the information flow paths for the case where the DM uses only one algorithm f for performing his task. The symbols in the figure are defined as follows:

- u is the decision variable for choosing between the five alternatives:

  (1) DM performs the stage by himself.
  (2) DM queries the mainframe, performs his own processing, and compares the two results.
  (3) DM queries the intelligent terminal, performs his own processing, and compares the two results.
  (4) DM queries the mainframe and relies on its response.
  (5) DM queries the intelligent terminal and relies on its response.

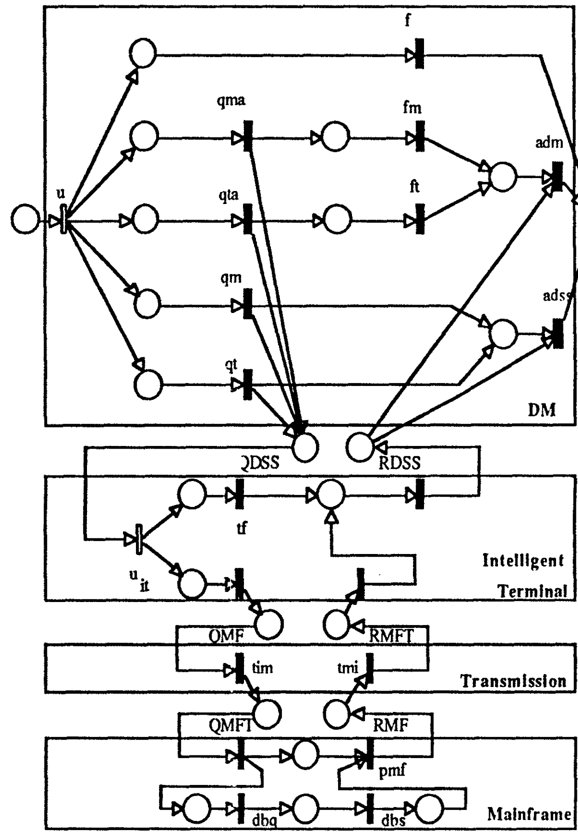

Fig. 6 Petri Net Model of DM Aided by the DSS

- qma is the algorithm used by DM to query the mainframe in alternative 2.
- qta is the algorithm used by DM to query the intelligent terminal in alternative 3.
- qm is the algorithm used by DM to query the mainframe in alternative 4.
- qt is the algorithm used by DM to query the mainframe in alternative 5.
- fm is the algorithm that DM executes when he has queried the mainframe in alternative 2.
- ft is the algorithm that DM executes when he has queried the intelligent terminal in alternative 3.
- adm is the algorithm used by DM to assess the response of the DSS and to compare it with the result of his own processing in alternatives 2 and 3.
- adss is the algorithm used by DM to assess the response of the DSS in alternatives 4 and 5.
- QDSS is the query sent by DM to the DSS.
- RDSS is the response sent by the DSS to DM.
- u_{it} is the decision variable which determines whether the intelligent terminal or the mainframe must process the query.
- tf is the algorithm performed by the intelligent terminal to process the query.
- QMF is the query sent by the intelligent terminal to the mainframe.
- RMFT is the response from the mainframe transmitted by the

network to the intelligent terminal.
- tim is the protocol of transmission from the intelligent terminal to the mainframe.
- tmi is the protocol of transmission from the mainframe to the intelligent terminal.
- QMFT is the query from the intelligent terminal transmitted by the network to the mainframe.
- RMF is the response from the mainframe.
- pmf is the algorithm performed by the mainframe for processing the query.
- dbq is the algorithm that queries the database.
- dbs is the algorithm that performs the search in the database.

This model shows that the decisionmaker interacts with the DSS by fusing the information that the latter produces. Therefore, it is possible to evaluate the synchronization between DM and the DSS.

The places labelled QDSS and RDSS represent the structural places that contain the information exchanged by DM and the DSS. In accordance with the Predicate Transition Net model, the transitions adm and adss are the only interactional ones. These transitions will fire only if the tokens in their input places have the same attribute T_n, i.e., they correspond to the same input from the environment. The measure of the synchronization between DM and the DSS evaluates, for each input and each stage, the sojourn time of the item of information in the memory place of the preset of adm or adss. Since the emphasis in this study is the coordination between DMs, for simplicity, it is assumed that the synchronization between DM and DSS is perfect. The same approach can be used to analyze the case when synchronization between DMs and the DSS is not perfect.

EXAMPLE

The impact of a decision support system on the coordination of a two-person organization is the key question addressed in this example. The degradation of the synchronization of a decisionmaking organization can result from two types of factors:

- the dynamics of the activities which lead the decisionmakers to process various inputs with different priority orders.
- the information flow paths that each decisionmaker uses to perform his task.

The impact of the first category of factors on the decisionmaking process was discussed in Grevet et al. (1988).This example assesses the second type of factors. Such a situation arises when the decisionmakers are provided with a DSS which allows them to access different local or remote computer facilities. The DSS can alter significantly the coordination of the activities, depending on the configuration of the system with respect to the organization.

The Organization and the Task

The example presented in this section aims at modeling the organizational structure and decisionmaking activities of a two-person organization in a simple ASW context. The task models a mission of surveillance that consists of listening to detect enemy submarines. In such an environment, the use of decision support systems to process the signals and discriminate between them is necessary.

The organization consists of a submarine and a surface ship which are in charge of tracking enemy submarines. It is a hierarchical organization where the submarine is the subordinate and the surface ship the commander. This example has been studied from another standpoint by Papastavrou (1986). The Petri Net model of such an organization is presented in Figure 7.

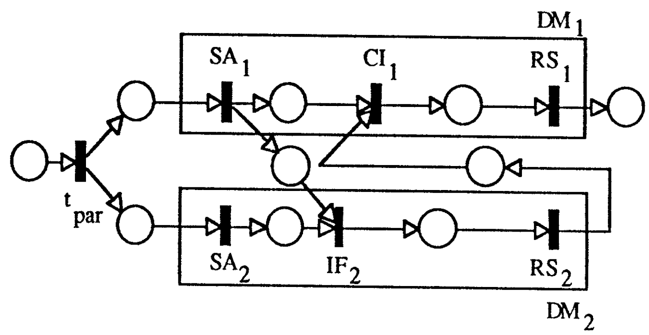

Fig. 7 Petri Net Model of Subordinate (DM_1) and Commander (DM_2)

The decision-making process of the commander and the subordinate have three stages each. In the Situation Assessment stages, they assess the signals that they receive from the environment. The subordinate sends the result of his own assessment to the commander, who fuses in the Information Fusion stage this information with his own assessment. On the basis of the result of this interaction, the commander identifies the signal and produces an order which is the sent to the subordinate. The latter interprets the order in the Command Interpretation stage and produces the organizational response.

The task is modeled as the alphabet X and the probability distribution prob(x) such that:

$$X = \{x_i = a_i b_i c_i d_i e_i f_i \mid (a_i, b_i, c_i, d_i, e_i, f_i) \in \{0,1\}^6\}$$

$$\forall (x_i, x_j) \in X \times X, \quad prob(x_i) = prob(x_j)$$

Therefore, each input consists of an ordered string of six bits. There are 64 possible inputs that represent the signals that must be identified by the organization in order to produce the response. It is assumed, furthermore, that these inputs are equiprobable, so that the probability distribution prob(x) is defined by:

$$\forall x_i \in X, \; prob(x = x_i) = \frac{1}{64} \qquad (19)$$

The organization can produce four responses, labelled R_1, R_2, R_3 and R_4:

- if the bits a_i and d_i are both equal to 0, the signal does not come from an enemy submarine and, therefore, the submarine DM_1 should not do anything. This response is R_1. The probability of having such an input is 1/4.
- if b_i and e_i are both equal to 0, the signal comes from an enemy submarine which is trying to test the capabilities of submarine DM_1. This one should deceive it by under-reacting. This response is R_2. The probability of having such an input is 3/16.
- if the bits c_i and d_i are both equal to 0, the signal comes from an enemy submarine which is moderatly threatening submarine DM_1. The latter should over-react to this threat to deter the enemy submarine. This response is R_3. The probability of having such an input is 9/64.

- otherwise, the signal comes from an enemy submarine which is threatening submarine DM_1. In this case, DM_1 should also over-react but at a higher level than previously. This response is R_4. The probability of having such an input is 27/64.

Table 1 summarizes these possibilities. The partitioning of the input is done according to the following rule:

- the submarine, DM_1, receives the first three bits $a_ib_ic_i$.
- the surface ship, DM_2, receives the last three bits $d_ie_if_i$.

The decisionmaking process takes place on the basis of this partitioning. The Table 2 presents the cost matrix that gives the costs associated with the discrepancies between the ideal responses and the actual responses provided by the organization.

TABLE 1 Organizational Responses

input	response
$(a_i, d_i) = (0, 0)$	R_1
$(a_i, d_i) \neq (0, 0)$ $(b_i, e_i) = (0, 0)$	R_2
$(a_i, d_i) \neq (0, 0)$ $(b_i, e_i) \neq (0, 0)$ $(c_i, f_i) = (0, 0)$	R_3
$(a_i, d_i) \neq (0, 0)$ $(b_i, e_i) \neq (0, 0)$ $(c_i, f_i) \neq (0, 0)$	R_4

TABLE 2 Cost Matrix

ideal R \ actual R	R_1	R_2	R_3	R_4
R_1	0	2	3	4
R_2	4	0	1	2
R_3	6	4	0	2
R_4	8	4	3	0

The accuracy, J, of the organization is computed as the expected value of the cost for the particular set of inputs. It is assumed that, when DM_1 and DM_2 assess the input by themselves, without querying the DSS, they produce correctly the first two bits of the strings of three bits. That is, for an input $x_i = a_ib_ic_id_ie_if_i$, the result of the SA of DM_1 is $a_ib_iu_i$ where u_i is the value of the third bit that DM_1 produces : it is assumed that this value is equal to c_i with probability 1/2. In the same way, the result of the SA of DM_2 is $d_ie_iv_i$ where v_i is the value of the sixth bit that DM_2 produces: this value is equal to f_i with probability 1/2.

It is furhter assumed that the decisionmakers query the DSS only during their Situation Assessment stages. Figures 8 and 9 provide the models of the organization aided by the DSS: in Figure 8, the model of DSS is aggregated; in Figure 9 the whole model is shown.

Only three of the five alternatives a DM has to perform his processing in any internal stage where he can use the DSS are considered:

(i) DM_i does not access the DSS .

(ii) DM_i queries the intelligent terminal and relies on its response.

(iii) DM_i queries the DSS and compares its response to his own assessment.

In the remainder of this section, the following notation will hold:

- SA_i represents alternative (i).
- IT_i represents alternative (ii).
- MF_i represents alternative (iii).

This model shows that multiple flow paths can be used to process the information. Since each DM has three alternatives with respect to the use of the DSS, there are nine pure organizational strategies:

- (SA_1, SA_2)
- (SA_1, IT_2)
- (SA_1, MF_2)
- (IT_1, SA_2)
- (IT_1, IT_2)
- (IT_1, MF_2)
- (MF_1, SA_2)
- (MF_1, IT_2)
- (MF_1, MF_2)

A mixed strategy $\delta_i(p_i^1, p_i^2, p_i^3)$ for DM_i corresponds to a convex combination of his three pure strategies SA_i, IT_i and MF_i weighted by the probabilities p_i^1, p_i^2, p_i^3.

An organizational behavioral strategy is the combination of the mixed strategies of DM_1 and DM_2. Therefore, it corresponds to

$$(\delta_1(p_1^1, p_1^2, p_1^3), \delta_2(p_2^1, p_2^2, p_2^3)).$$

It is assumed that the processing of information through the use of the DSS provides different results depending on whether the intelligent terminal or the mainframe is queried. When the intelligent terminal is accessed, the decision-makers can produce correctly the first bit of the strings of three bits. That is, for an input $x_i = a_ib_ic_id_ie_if_i$, the result of the SA of DM_1 for the alternative IT_1 is $a_iu_iy_i$ where u_i and y_i are the values of the second and third bits that DM_1 produces: each of these two values is equal to the actual value with probability 1/2. In the

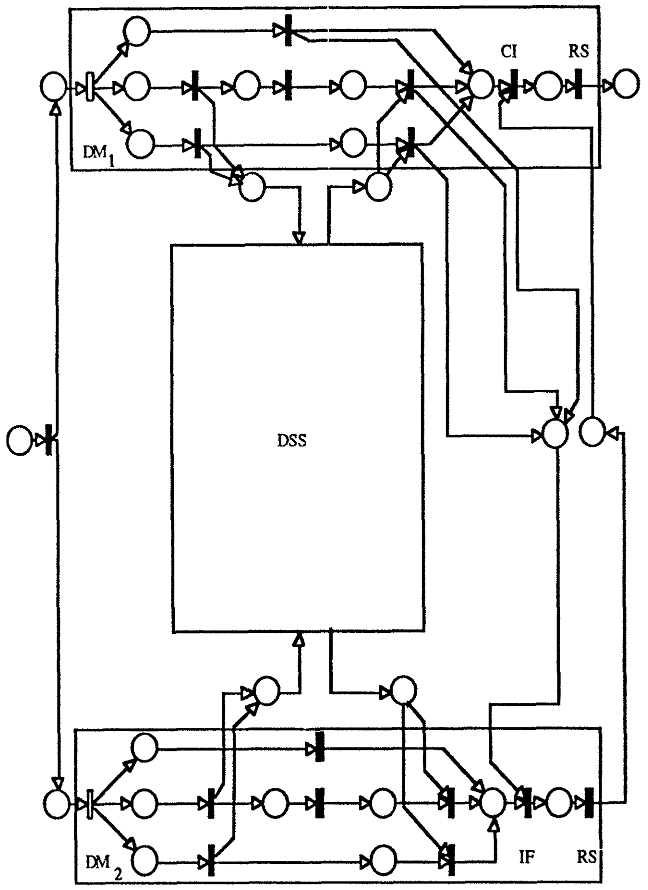

Fig. 8 Two-Person Organization Aided by DSS - Aggregated Representation

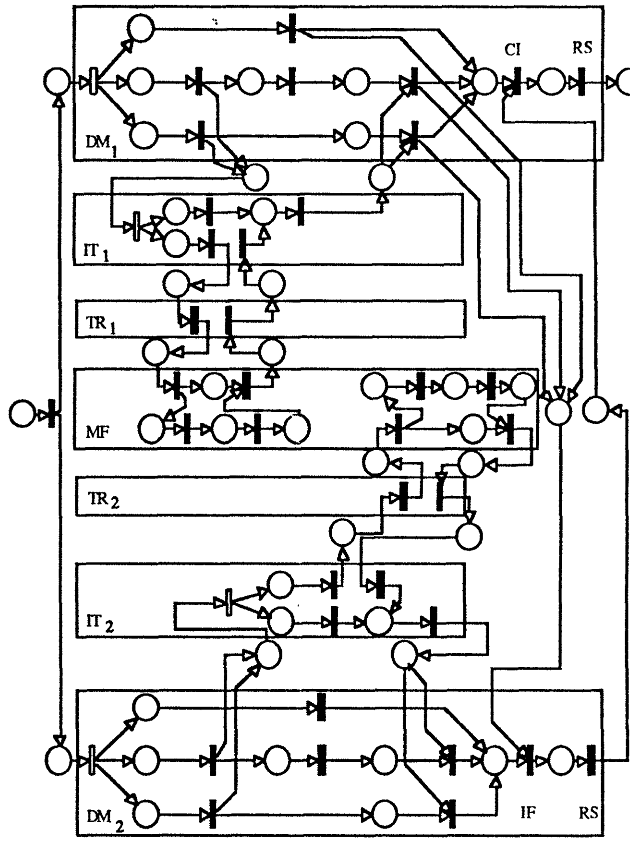

Fig. 9 Two-person Organization Aided by DSS

same way, the result of the SA of DM_2 for the alternative IT_2 is $d_i v_i z_i$ where v_i and z_i are the values of the fifth and sixth bits that DM_2 produces: each of them corresponds to the actual value with probability 1/2.

When the mainframe is accessed, the decision-makers are able to produce correctly all three bits of their respective strings. That is, for an input $x_i = a_i b_i c_i d_i e_i f_i$, the result of the SA stage of DM_1 for the alternative MF_1 is $a_i b_i c_i$. The result of the SA stage of DM_2 for the alternative MF_2 is $d_i e_i f_i$. This means that, when the organizational strategy is (MF_1, MF_2), the organization will be able to produce the correct response for all inputs. For all other strategies, the responses provided may differ from the ideal response.

The access to the intelligent terminal provides, however, a means of improving the timeliness of the decision-making process. Indeed, it will be assumed that the amount of time necessary to process the information is lower when the decisionmaker uses his intelligent terminal than when he queries the mainframe or performs his processing alone.

The amount of time taken by the decisionmakers to execute the different algorithms is equal to one unit of time, except for the Situation Assessment algorithms. Different cases have been investigated in which the processing times of these algorithms differ from unity.

These considerations account for what occurs in most situations in C^3 systems. The different decisionmakers have access to different facilities which do not have the same response time or the same accuracy. On the one hand, an intelligent terminal is likely to provide faster responses because it is co-located with the decisionmaker. However, it has no centralized database which can aggregate data from multiple sensors to get a global picture of the situation and, therefore, the responses it can provide are necessarily less accurate. On the other hand, the access to the mainframe may require the communication of data from and to remote locations through a network: the response time can be quite long.

The next section contains the results obtained for different access times to the mainframe. In each case, the performance loci have been constructed for the three measures, accuracy J, expected delay T, and synchronization S_T.

Results

The results on the accuracy of the responses produced by the organization for the nine pure organizational strategies, are listed in Table 3. Accuracy is maximal when both decision-makers query the mainframe and reaches its worst level when they both query their intelligent terminal.

TABLE 3 Accuracy of the Organization

strategy	SA_1 SA_2	SA_1 IT_2	SA_1 MF_2	IT_1 SA_2	IT_1 IT_2	IT_1 MF_2	MF_1 SA_2	MF_1 IT_2	MF_1 MF_2
J	0.42	0.76	0.21	0.76	1.01	0.69	0.21	0.69	0.00

Two cases have been investigated as far as the processing times of the Situation Assessment stages are concerned:

(i) **case 1**: SA_1 and SA_2 take 10 units of time. IT_1 and IT_2 take 5 units of time. MF_1 and MF_2 take 15 units of time. This case corresponds to the situation where the processing times of both decision-makers are equal when they use the same strategy with respect to the use of the DSS.

(ii) **case 2**: SA_1 and SA_2 take 10 units of time. IT_1 and IT_2 take 5 units of time. MF_1 takes 15 units of time but MF_2 takes 10 units of time. This corresponds to the situation where the commander has a faster access to the mainframe than the subordinate because of a better transmission time.

In both cases, when the two DMs perform their situation assessment by themselves, they take the same amount of time to do it. This means that the Information Fusion stage of DM_2 is perfectly synchronized.

Tables 4 and 5 show the results for the expected delay, T, and the synchronization, S_T, in case 1 and case 2, for the nine pure organizational strategies.

TABLE 4 Delay and Synchronization in Case 1

strategy	SA_1 SA_2	SA_1 IT_2	SA_1 MF_2	IT_1 SA_2	IT_1 IT_2	IT_1 MF_2	MF_1 SA_2	MF_1 IT_2	MF_1 MF_2
T	15	15	20	15	10	20	20	20	20
S_T	2	7	7	7	2	12	7	12	2

TABLE 5 Delay and Synchronization in Case 2

strategy	SA_1 SA_2	SA_1 IT_2	SA_1 MF_2	IT_1 SA_2	IT_1 IT_2	IT_1 MF_2	MF_1 SA_2	MF_1 IT_2	MF_1 MF_2
T	15	15	15	15	10	15	20	20	20
S_T	2	7	2	7	2	7	7	12	7

In case 1, the maximum delay is obtained when at least one of the decisionmakers accesses the mainframe. The minimum delay is reached when both decisionmakers use their intelligent terminal. When the maximum delay is reached, the synchronization can have very different values depending on the coordination of the strategies of the decisionmakers. When they both access the mainframe, the synchronization is optimal with a delay of 20 units of time. For this same delay, this synchronization can degrade considerably, if one of them accesses his intelligent terminal as the other queries the mainframe.

In case 2, the maximum delay is reached when DM_1 accesses the mainframe. Nevertheless, when DM_2 queries the mainframe alone, the delay does not increase to this level. The optimal synchronization can no longer be obtained when the delay is maximal. Furthermore, the worst value for the synchronization is reached only for one pure organizational strategy, i.e., for (MF_1, IT_2). This value was reached for two pure organizational strategies in case 1, i.e., (MF_1, IT_2) and (IT_1, MF_2).

The interpretation of the results can be done through the consideration of the performance loci for the measures J, T and S_T. The performance loci for the two cases presented in the previous section are shown in Figures 10 and 11. They represent the values of J, T and S_T reached for each organizational strategy, pure or behavioral.

These figures show the relations and tradeoffs between the various measures of performance. It is recalled that:

- the lower the value of T, the better the delay.
- the lower the value of S_T, the better the synchronization.
- the lower the value of J, the better the accuracy.

In case 2, the part of the locus where J is the lowest, i.e., where the accuracy is the best, corresponds to higher values of S_T than in case 1. This shows that there exists a trade-off between accuracy and synchronization when the DSS does not have the same response time for the two decisionmakers.

In both cases 1 and 2, when the expected delay, T, is minimal, the synchronization, S_T, is also minimal. This is due to the fact that the intelligent terminals provide the fastest way of performing Situation Assessment, and that the delay will be minimal only if both decisionmakers query their terminal. The assumption that these terminals give the responses to both decisionmakers in the same amount of time is realistic because there is no delay due to transmission and the algorithms that they use are similar. Conversely, the fact that the synchronization is minimal does not imply that the delay will be minimal. In case 1, the synchronization reaches its lowest value for all possible values of the delay. It corresponds to the fact that, for any delay, the decisionmakers can find some way to be as well synchronized as possible.

If a constraint is imposed on the delay, the synchronization of the organization does not degrade. One can notice that the more stringent the constraint on T, the more likely the synchronization will reach a good value. In case 2, the synchronization does not reach its lowest value for all values of T; as in case 1, the best values of S_T are obtained for the lowest delays.

In case 2, the more the timeliness of the organization degrades, the more the synchronization will degrade too. When DM_1 uses the mainframe, there is no way for the organization to be well synchronized. DM_2 will have to wait for long inervals of time before receiving the data that he needs in his information fusion stage.

These facts show that the introduction of a decision support system in an organization can have different effects on the coordination of the activities. If the organization members are well coordinated when they do not use the DSS, the latter can

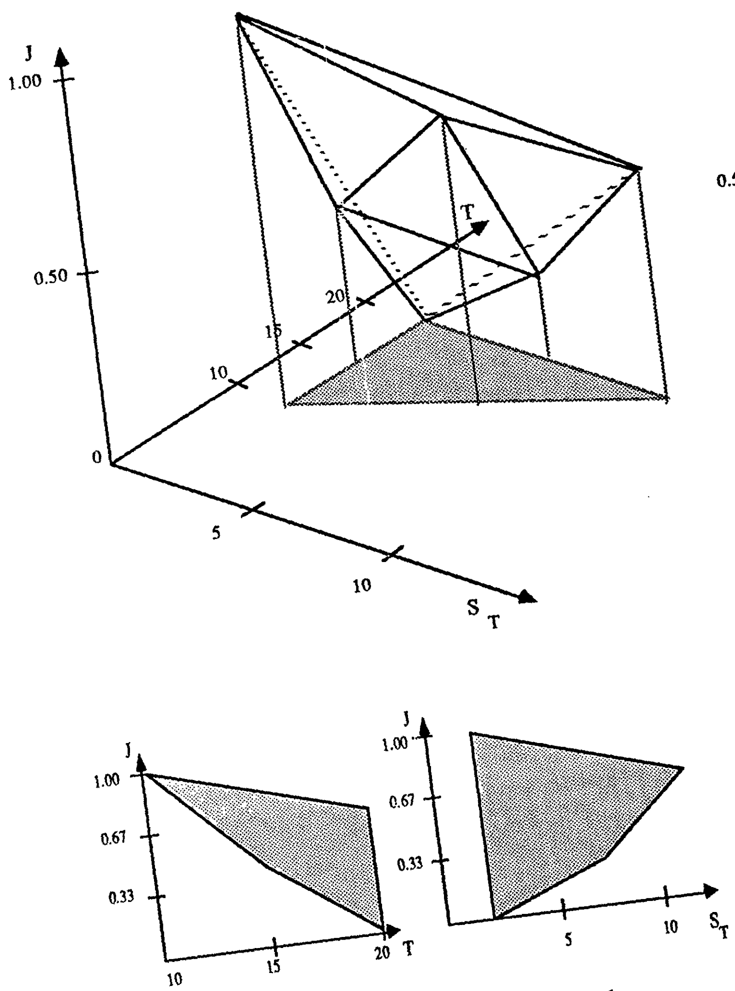

Fig. 10 Performance Locus in Case 1

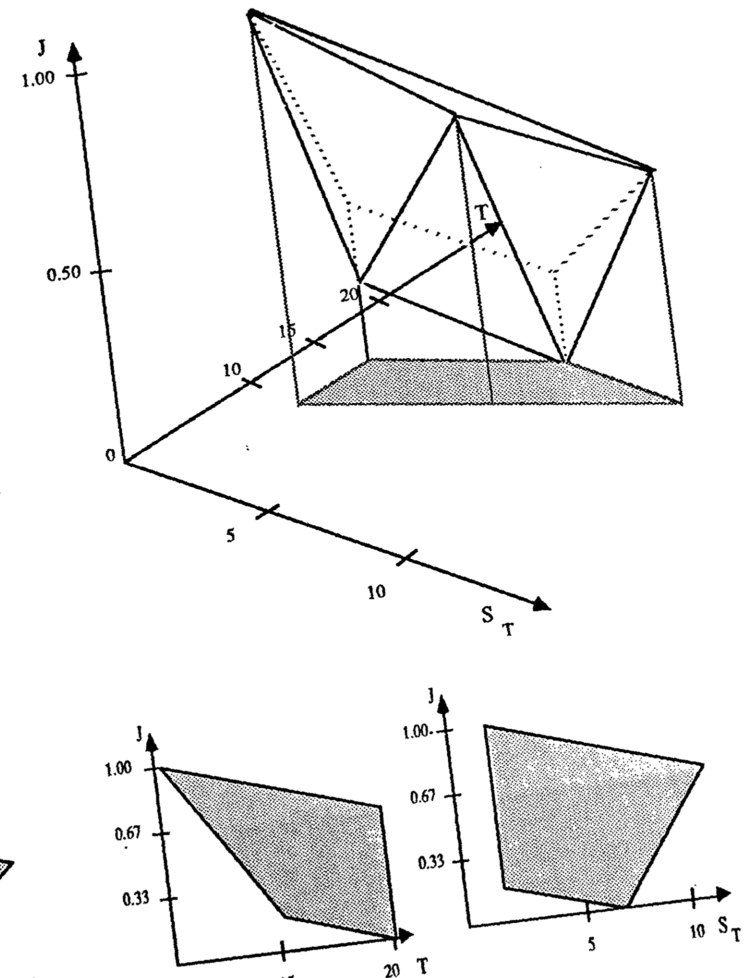

Fig. 11 Performance Locus in Case 2

degrade this coordination because of two factors:

- the decision-makers can use a larger number of facilities; it is difficult to coordinate their access.
- the quality of the responses provided by the DSS, i.e., the response time and the information produced, can be very different from one decision-maker to another.

Therefore, on the one hand, *the decision-makers have many more alternatives that they can use to perform their task; the coordination of these activities is consequently more difficult to achieve*. On the other hand, in coordinating their activities, the organization members must take into account the fact that the *DSS does not perform equally well for all of them*.

This latter consideration is illustrated by case 2 of the example: there is a tradeoff between accuracy and timeliness which is coupled with a tradeoff between accuracy and synchronization. In order to achieve good accuracy, the organization members must use strategies which lead to a degradation in timeliness and synchronization. Conversely, if the decisionmakers wants to be well synchronized, the accuracy will degrade because they cannot access the mainframe together.

Therefore, the decision support system, depending on its characteristics, leads to mixed effects on the effectiveness of the organization. As in case 2, it can lead to an improvement both in accuracy and timeliness of the organization, but the coordination then degrades. Conversely, as in case 1, it cannot produce an improvement both in accuracy and timeliness; but coordination is always highest when accuracy or timeliness are optimal.

CONCLUSIONS

The concept of coordination was defined as relating to the consistency of the information exchanged by the different organization members and to the synchronization of the various activities. The latter bears directly on the dynamics of the decisionmaking process. A decisionmaking 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 introduced in order to perform a quantitative evaluation of the coordination of decision-making processes, i.e., information consistency and synchronization.

A methodology for assessing quantitatively the impact of a decision support system on the activities of a decision-making organization has been presented. It was used to show that the introduction of a decision support system can alter considerably the synchronization of the various activities because the capabilities offered to the various decisionmakers by the system may differ. For example, a certain decisionmaker may have faster access to the central database than another one, because of different transmission times. However, the fact that some decisionmakers are provided with better capabilities can allow the organization to improve both the timeliness and the accuracy of the process.

REFERENCES

Bejjani, G.J., and A.H. Levis, 1985, "Information Storage and Access in Decision-Making Organizations," *Proc. 8th MIT/ONR Workshop on C^3 Systems*, LIDS-R-1519, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

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.

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., 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

Keen, P.G.W. and M.S. Scott Morton, 1978, *Decision Support Systems: an Organizational Perspective*, Addison-Wesley, Reading, MA.

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

Levis, A. H., 1988, "Human Organizations as Distributed Intelligence Systems," *Proc. IFAC Symposium on Distributed Intelligence Systems,"* Pergamon Press, Oxford.

Papastavrou, J.D., 1987, "Distributed Detection with Selective Communications," MS Thesis, LIDS-TH-1563, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Waltz, E.L., and D.M. Buede, 1986, "Data Fusion and Decision Support for Command and Control," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-16, No. 6, pp.865-879.

Weingaertner, S.T., and A.H. Levis, 1987, "Evaluation of Decision Aiding in Submarine Emergency Decision-Making," *Proc. IFAC Symposium on Man-Machine Systems*, Pergamon Press, Oxford.