AUGUST 1989 LIDS-P-1899

ON THE GENERATION OF A VARIABLE STRUCTURE AIRPORT SURFACE TRAFFIC CONTROL SYSTEM*

Jacques J. Demaël
Alexander H. Levis

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

ABSTRACT

A quantitative approach for modeling variable structure decision making organizations is presented. In these organizations, the interactions between the decisionmakers can change, depending on the task being processed. Using Colored Petri Nets as the appropriate mathematical formalism, an algorithm is presented for generating such variable structures. The approach is illustrated through the modeling and design of a hypothetical Airport Surface Traffic Control System.

* This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the U.S. Office of Naval Research under contract no. N00014-84-K-0519.

INTRODUCTION

To meet ever increasing requirements of reliability and reconfigurability from users, systems that adapt the interactions between components to the task being processed have been proposed. As some patterns of interactions may be more suitable for the processing of a given input than others, a properly designed variable structure system can be expected to achieve a higher overall performance. Theoretical and practical evidence indicates that the control problem can be more easily solved if some issues are addressed at the plant design level. A good plant design, for example, may eliminate difficult control problems. However, little is known on how to generate analytically in some orderly manner system designs that are not just variants of a single structure.

This paper presents a quantitative framework for representing variable structure systems and a methodology for generating variable structure decisionmaking organizations perceived as distributed intelligence systems. Teams of human decisionmakers are organized when the task that must be processed exceeds the cognitive limits of a single individual [1]. Through specialization, individuals acquire the capacity to apply relatively complex cognitive strategies to a narrowly-defined task environment. Through division of labor, substantial cognitive resources can be brought to bear concurrently on many tasks.

An appropriate mathematical framework is defined based on the theory of Colored Petri Nets [2], an extension of Petri Net theory [3]. As in Ordinary Petri Nets, a Colored Petri Net is a graph with two sets of nodes, places and transitions. A transition is denoted by a bar node and describes a process, while a place is denoted by a circle node and models a buffer between two processes. The precedence relations between places and transitions are described by the links of the net. In CPN theory, the tokens represent messages and have an identity, denoted by a set of attributes, collectively referred to as color. The variable interactions between processes are described by annotating the links of the net, so thet some tokens are allowed to be carried by an link, while some others cannot pass. The annotation of the links is based on the language of Linear Algebra.

In the next sections, the key elements of the methodology are presented. First, a mathematical model of interactions in decisionmaking organizations is presented. A framework is described to represent variable structure organizations as well as fixed structure organizations. The third section describes properties of variable structures in the language of Colored Petri Nets. The fourth section describes the constraints that must be verified by a variable structure to make physical sense. In the fifth section, the generation of variable structure systems is illustrated for a hypothetical design of an Airport Surface Traffic Control system. Finally, the sixth section characterizes the set of solutions using a partial ordering of variable structures.

MATHEMATICAL MODEL

A Decision Making Organization (DMO) is seen as an information processing system that must perform several functions to accomplish its mission [4],[5]. The functions are divided into individual tasks, the *roles*. Each role is a series of repetitive procedures that are prescribed by the requirements of the mission, so that each object's activity contributes a little to each of the several functions. The inputs to the system are the observations carried out by the *sensors*. These items of information are transmitted to the proper destinations within the organization, they are analyzed, and the selected response is finally implemented by the effectors. The model is restricted to observations that are *temporally consistent,* i.e., observations that refer to the same event with a specific time of occurrence [6]. It is further assumed that the processing of one set of simultaneous observations is *deterministic*, in the sense that there is no uncertainty concerning the sequence of processesthta are activated to perform the task.

A DMO has a *variable structure* if the interactions between the individual decisionmakers can vary. Conversely, a system for which the interactions cannot vary has a *fixed structure. This paper is restricted to systems whose variability is triggered by the information contained in the sensors' observations*. The goal is to create a Colored Petri Net (CPN) model of the flow of data from the sources to the roles, the exchange of information between roles, and the communication of messages to the effectors. A CPN can then be used to assess the effectiveness of a structure, using the System Effectiveness Analysis methodology as described in [7].

Sensors: A DMO processes data from N sources of information, i.e., N sensors, labeled Sensor 1 to Sensor N. Sensor n can output one signal or symbol from its associated set of possible signals, its output alphabet $Xn = \{xn_1, ..., xn_{|Xn|}\}$, which contains $|Xn|$ elements. In the Petri Net formalism, each sensor is modeled by a place (Fig. 1).

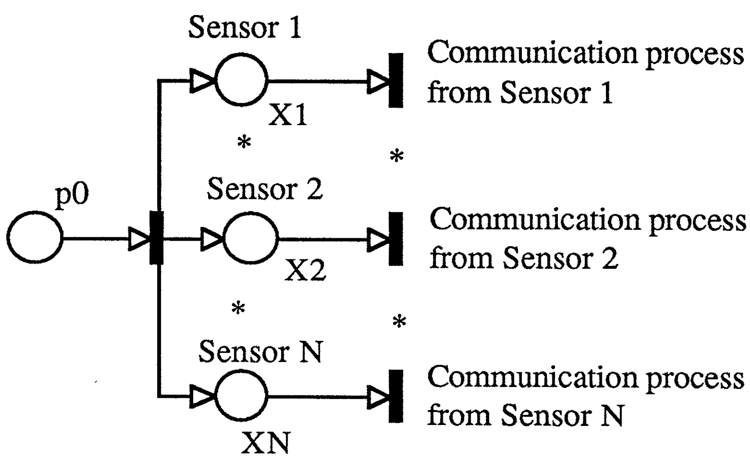

Fig. 1 Sensors

A transition models the communication of the sensor's observations. The temporal consistency of the observations is embedded in the fact that all sensors are the output of a single process. This process has a single input place p0, which is called the external place.

From the system point of view, temporal consistency also implies that the input to the system is an N-dimensional vector: x = (x1, x2, ..., xN) with components the N independent observations and where

$$x \in X = X1 \times X2 \times \ldots \times XN.$$

CPN Representation of an Interactions: An interaction is characterized by its pattern of activation over the set of inputs. Therefore, every interaction is described by a *diagonal* $|X| \times |X|$ matrix L, with $L_{ii} = 1$ if the i-th input in the lexicographic ordering of X activates the interaction, and $L_{ii} = 0$ if it does not.

In the Colored Petri Net model, an interaction is represented by a link between two transitions t1 and t2, as depicted in Fig. 2. The link indicates that the output of process t1 is an input to process t2. The place that belongs to the link is an *interactional* place, which models a communication buffer. The links are annotated by the matrix L

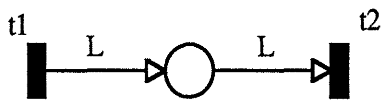

Fig. 2 Link

There are three basic types of interactions in a variable structure.

The *permanent* links. These are the links for which L is the $|X| \times |X|$ identity matrix. Every input requires this interaction to be processed. By convention, these links are depicted without annotation on a CPN model of the system.

The *inadmissible* links. These are the links for which L is the $|X| \times |X|$ null matrix. No input requires this interaction to be processed. These links are never depicted on a CPN model of the system

The *variable* links. These are the links for which L has 0s and 1s on the diagonal. Some inputs require this interaction in order to be processed, while some do not.

Inadmissible links and permanent links are of little interest as far as the coordination of roles is concerned because the existence of the interaction is not subordinated to the information content of the input. For variable links, the decision to interact or not is based on the information content of the input x, and must be simultaneously recognized by the roles that interact. The decision whether or not to interact must be based on sensors' observations. A Sensor i (an alphabet Xi) is said to be *effective*, if the decision whether or not to interact is based, partially or in whole, on the output of that sensor. More formally:

Proposition 1: Given a variable interaction described by a diagonal matrix L, the alphabet Xi is an *effective* alphabet of the interaction if and only if there exist two signals xi_1 and xi_2 in Xi, and (i) there exists an input x = (x1, x2, ..., xN) in X that activates the interaction, with xi = xi_1, (ii) there exists an input x' = (x'1, x'2, ..., x'N) in X that does not activate the interaction, with x'i = xi_2.

In the CPN model of a variable structure tokens have identity. A token's identity belongs to X, and x = <x1,...,xN> describes a message that has been generated by the set of simultaneous observations <x1,...,xN>. The matrix L attached to an interaction describes the set of tokens that can go through the link. A transition is enabled, if and only if all its input places contain messages as described by the annotation of its input links. When a transition fires, one token <x1,...,xN> is removed from each one of its input places and one token of color <x1,...,xN> is created in each output place.

Interactions: Each role in a fixed structure has been modeled by a subnet with four transitions and three internal places (Fig. 3) [5]. The four stage decisionmaking process consists of four algorithms SA, IF, CI, and RS. In Figure 3, x represents an input signal from an external source of information or another role. The *Situation Assessment (SA)* algorithm processes the incoming signal to obtain an assessment of the situation. The assessed situation z may be transmitted to other roles. Concurrently, the role may incorporate one or several signals z" from other parts of the system. The signals z and z" are fused together in the *Information Fusion* stage *(IF)* to produce the final situation assessment z'. The next transition, the *Command Interpretation* algorithm *(CI)* receives and interprets possible commands (v') from other roles, which restrict the set of options that can be consiered and responses that can be generated. The CI stage outputs a command v, which is used in the *Response Selection* algorithm *(RS)* to produce the response of the role, the output y. This output can be sent to the effectors and/or to other roles in the system. The input stage of a role may be SA, IF or CI, i.e., all the stages that accept external inputs. The final output stage, however, must be RS, the stage at which the role selects its response.

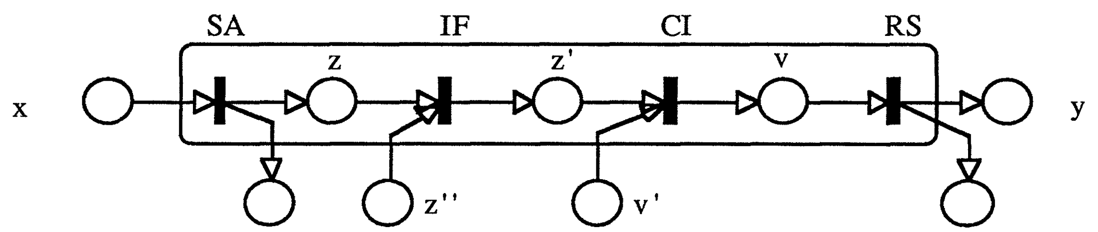

Fig. 3 Four Stage Model of a Decisionmaker or Intelligent Node

Every role in the system might not have access to all the sensors' observations. It might base its situation assessment on a restricted number of observations. Fig. 4 depicts schematically the interactions between sensors and roles and S_{ij} models a link between Sensor i and Role j, Role j incorporates the output of Sensor i to make his situation assessment.

Only certain types of interactions between roles make physical sense in the context of the model [8]. They are depicted in Fig. 5. For clarity, only the links from the i-th role to the j-th role have been represented. The symmetrical links from i to j are valid interactions as well.

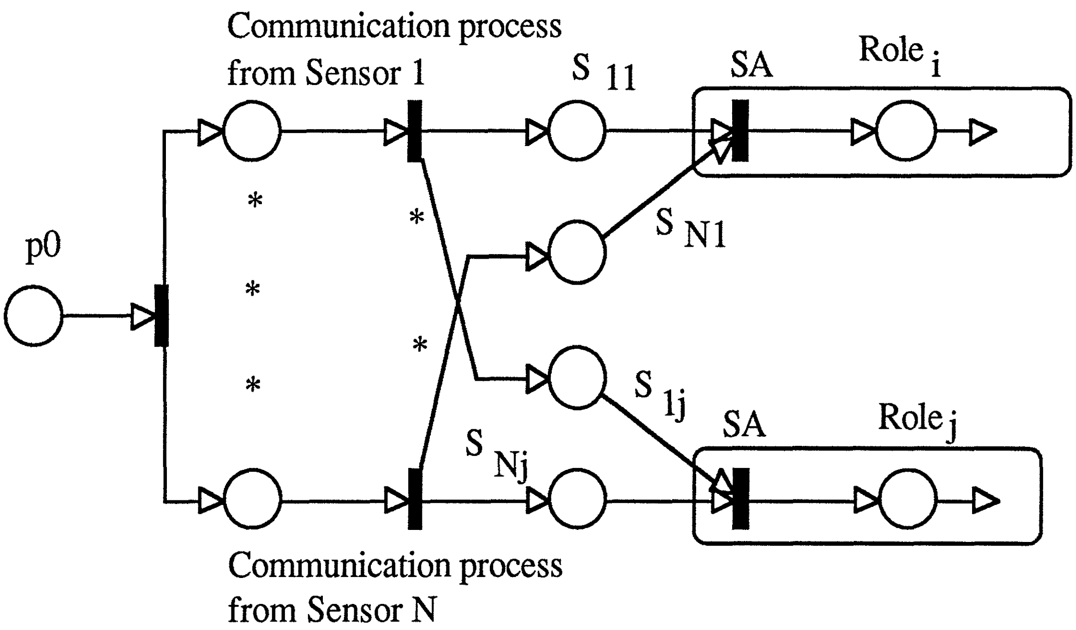

Fig. 4 Interactions Roles-Sensors

The parameter s_i models the case in which the i-th role communicates the response it has selected to the external environment through the effectors which are represented by a single transition with a unique output place, called the sink. F_{ij} is the interaction that occurs when the situation assessment which is produced as an output of the SA stage is sent to the j-th role to be fused with the assessment of the j-th role, and/or assessments from other roles. G_{ij} depicts the case where the response selected by the i-th role is the input of the j-th role. H_{ij} shows the sharing of a result. The i-th role informs the j-th role of its final decision. The j-th role may or may not take this information into account. Finally, C_{ij} has been introduced to model hierarchical relationships between roles. It describes the possibility of role i sending a command to role j.

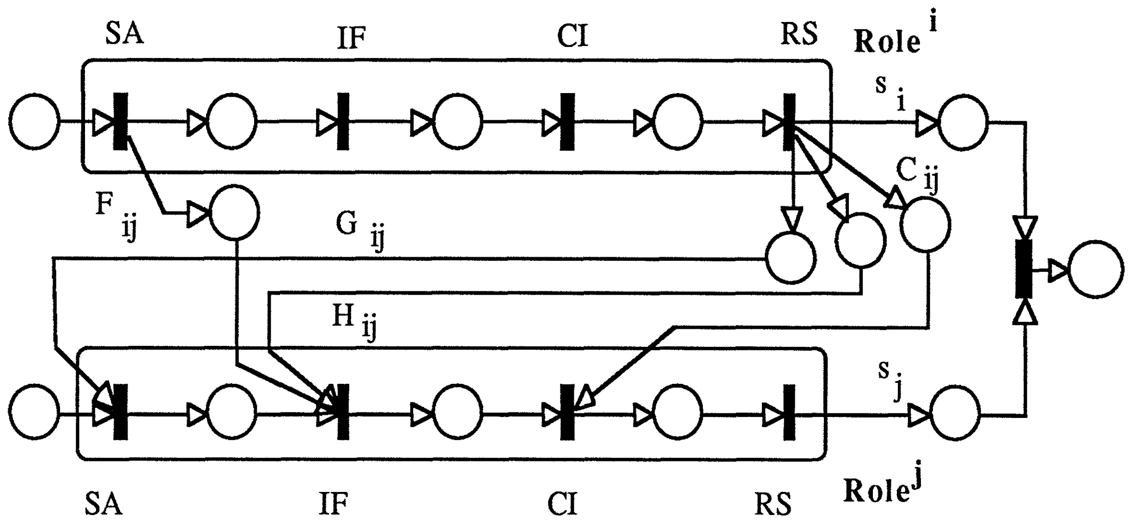

Fig. 5 Allowable Interactions between two Roles

Matrix representation of interactions: The model of interactions in a variable structure DMO leads to a unified representation in matrix form. Suppose that a variable structure contains R roles, N sensors, and that its input alphabet is $X = X_1 \times X_2 \times \ldots \times X_N$. Then, a variable structure is completely determined by the six-tuple:

$$\Pi = (S,\ s,\ F,\ G,\ H,\ C)\ .$$

- S is a $N \times R$ block array, which depicts the flow of information from the sensors to the organization.
- s is a $1 \times N$ block array that depicts the flow of information from the organization to the effectors.
- F, G, H, C are four $R \times R$ block arrays which model the interactions between roles within the organization. F_{ij} models the link from the SA stage of Role i to the IF stage of Role j. G_{ij} models the link from the RS stage of Role i to the SA stage of Role j. H_{ij} models the link from the RS stage of Role i to the IF stage of Role j. Cij models the link from the RS stage of Role i to the CI stage of Role j.
- Every block in (S, s, F, G, H, C) is a $|X| \times |X|$ diagonal matrix L with $L_{ii} = 1$ if the i-th input in the lexicographic ordering activates the link and $L_{ii} = 0$ if it does not.

If X, N, R are fixed, then the set of all six-tuples Π of dimensions X, N, R, is called V, the set of Well Defined Variable Structures (WDVS).

Fixed structures have been studied in Remy [8]. A fixed structure is determined by two parameters, the number of roles, R, and the number of sensors, N. A fixed structure of dimensions N, R, can be represented by the six-tuple:

$$\Sigma = (S', s', F', G', H', C')$$

- S' is an $N \times R$ array
- s' is a $1 \times N$ array,
- F', G', H', C' are four $R \times R$ arrays.

Their entries are in {0,1} with 1 if the interaction is present and 0 if the interaction is not present. Here again, if N and R are fixed, the set of all six-tuples S of dimensions N, R is called W, the set of Well Defined Fixed Structures (WDFS).

PROPERTIES

The mathematical framework is now related to the language of Petri Net theory, which is used as the formalism within which the design problem is articulated. Some properties of the sets V and W are stated below:

Proposition 2: Each element of V can be equivalently described in matrix form Π or by a Colored Petri Net.

There exists a one to one relationship between the representation of a variable structure in matrix form and a CPN model of the structure. One can thus work with the language that is most appropriate for one's needs. The proof is as follows. One transition is created for each process in the system, and a link is drawn between any two transitions that interact. This information is provided by the matrix form Π. Note, however, that a DMO contains also internal links, which describe the continuous flow of information within one role, and are not embedded in the matrix form Π. The key proposition is that the internal links are completely determined by the activation of interactional links: a link between two internal processes t1 and t2 within a role exists if and only if t1 has at least one input link - internal or interactional.

Proposition 3: Each element of W can be equivalently described in matrix form Σ or by an Ordinary Petri Net.

The proof for fixed structures is similar to the proof for variable structures.

Finally, the theory of variable structures is related to the theory of fixed structures through Proposition 4.

Proposition 4: Any variable structure corresponds to a mapping

$$\Pi: \quad X \dashrightarrow W \qquad x \dashrightarrow \Pi(x),$$

which associates with each input in X one and only one fixed structure.

Therefore, the Colored Petri Net model of a variable structure can be represented as a mapping from X into the set of Ordinary Petri Nets W. In the Petri Net literature, the decomposition of a CPN into a mapping is referred to as "unfolding" [9], and the translation of a mapping into a CPN is called "folding", because it yields a more compact representation. Finally, the sets V and W can be investigated using some partial orderings, which allow one to sort elements in a set.

Proposition 5: The set V of variable structures is ordered by the binary relation ACT, where Π ACT Π' means that very input that activates an interaction in Π activates the same interaction in Π'.

The elements of V can be ordered from the ones with the least activation to the ones with the most activation.

Proposition 6: The set W is ordered by the binary relation SUB, where Σ SUB Σ' means that every interaction in Σ is present in Σ', i.e. the Ordinary Petri Net that represents Σ is a subnet of the Ordinary Petri Net that represents Σ'.

The elements of W can be ordered from the ones that are least connected to the ones that are maximally connected.

CONSTRAINTS

All the variable structures that belong to V might not model DMOs that make physical sense. Some generic constraints must be defined on V, to restrict the class of variable structures to those that are admissible. The generic constraints on V can be divided into two classes. The first class relates the properties of variable structures to the properties of fixed structures, as described in Remy [8]. The second class is specific to variable structures [10]. The set of variable structures that satisfy the generic constraints is called AV, the set of Admissible Variable Structures (AVS).

Let Π be a variable structure. For any x in X, the fixed structure $\Pi(x)$ must satisfy:

- (R1) (a) The Ordinary Petri Net that corresponds to $\Pi(x)$ should be connected, i.e., there should be at least one (undirected) path between any two nodes in the Net; (b) A directed path should exist from the external place to every node of the PN and from every node to the sink.
- (R2) The Ordinary Petri Net that corresponds to $\Pi(x)$ should have no loops, i.e., the structure must be acyclic.
- (R3) In the Ordinary Petri Net that corresponds to $\Pi(x)$, there can be at most one link from the RS stage of a role i to another role j, i.e., for each i and j, only one element of the triplet $\{G(x)_{ij}, H(x)_{ij}, C(x)_{ij}\}$ can be non-zero.
- (R4) Information fusion can take place only at the IF and CI stages. Consequently, the SA stage of a role can either receive observations from sensors, or receive one and only one response sent by some other role.
- (R5) There cannot be one link from the SA stage of role i to the IF stage of role j and a link from the RS stage of role i to the SA stage of role j. Such an arrangement cannot deadlock.

Constraint R1(a) eliminates data flow structures that do not represent a single structure. Constraint R1(b) insures that the flow of information is continuous within the organization. Constraint R2 eliminates fixed data flow structures that contain cycles. This restriction is imposed to avoid deadlocks and infinite circulation of messages within the organization. Constraint R3 indicates that it does not make sense to send the same output to the same role at several stages. It is assumed that once the output has been received by a role, this output is stored in its internal memory and can be accessed at later stages. Constraints R4 and R5 ensures that the IF stage is indeed a stage at which items of information coming from different sources are fused.

- (R6) If the first stage of a role is SA, then each input link in S and G is permanent
- (R7) If the first stage of a role i is IF, then each input link F_{ji}, H_{ji} for j in [1..R] is permanent.
- (R8) If the first stage of a role i is CI, then each input link C_{ji} for j in [1..R] is permanent.
- (R9) Let L be a variable link between two stages t1 and t2, and let us suppose that Xi is an effective alphabet of the variable interaction.
- If t1 is a SA stage and t2 is a IF stage, then there must be in every $\Pi(x)$ a directed path from the place Sensor i to t1, and a directed path from the place Sensor i to the SA stage of the role that contains t2.
- If t1 is a RS stage and t2 is a IF stage, then there must exist in every $\Pi(x)$ a directed path from the place Sensor i to t1, and a directed path from the place Sensor i to the SA stage of the role that contains t2.
- If t1 is a RS stage and t2 is CI stage, then there must exist in every $\Pi(x)$ a directed path from the

place Sensor i to t1, and a directed path from the place Sensor i to the IF stage of the role that contains t2.

Constraints R6, R7, and R8 proceed from a common rationale. They state that a role at its input stage does not have any knowledge about the input to the system, and cannot exhibit a variable interaction. Thus, at the SA stage, any link between the sensors and the roles must be fixed. Similarly, if a role receives the response from another role, the latter must always communicate its response (R6). Constraints R7 and R8 incorporate the fact that the input stage of a role can be the Information Fusion or Command Interpretation stages. Constraint R9 states that a variable interaction between two stages t1 and t2 must be based on sources of information that are accessed jointly by the roles that interact. The stage t1 must determine, based on some information it has accessed, whether or not it has to send a message to t2. Similarly, the role that contains t2 must infer from some of the information it has already received, whether or not it must wait for a message from t1 before initiating process t2. The condition that the information can be accessed is formulated by checking that there is a flow of information (a directed path) from a source to the stages at which the information contained in the sensor observation is needed. In other words, an *effective* source of information must be *accessible.*

A system designer can introduce constraints which reflect the requirements of the specific application. Some links might be ruled in or ruled out, a certain pattern of variability might be prescribed, or some hierarchical relationships between roles need be preserved. These requirements can be expressed in a matrix form Π or on a Colored Petri Net.

AIRPORT SURFACE TRAFFIC CONTROL

This section illustrates the specification of requirements for a hypothetical design of an Airport Surface Traffic Control system (ASTC) at Logan Airport in Boston, Massachusetts.

The ASTC System: ASTC is broadly defined as the portion of the Air Traffic Control system that is responsible for traffic on the runways and taxiways of an airport. The system consists of two control positions, Local Control and Ground Control, which are stationed in the tower cab using visual surveillance, voice radio, and ground surveillance radars wherever available. Local Control handles the traffic on the runways and in the airspace in the immediate vicinity of the airport, while Ground Control handles the traffic on the taxiways, and, at some airports, issues advisories regarding airplane movement at the ramps. This example deals with the design of a variable ASTC system that encompasses one Local Controller (LC) and two Ground Controllers (GC), i.e., a system with three roles. The plan of Logan Airport has been simplified, as depicted in Fig. 6, in order to focus on the issues of the design.

The airport has three terminals and two runways, A and B. Planes land and take off on runways, and move to and from the terminals on the taxiways. The utilization of the runways depends on the direction of the wind, the guiding principle being that landings and takeoffs are done "against the wind." If the wind blows from the North or from the South, Runway A is used. If the wind blows from the East or the West, planes land and take off on Runway B. Finally, both runways can be used if wind speed is below a certain limit. Under low wind conditions the runways are not used in the same way however. Because of noise abatement concerns in the communities around Boston Harbor, commercial aircraft land and take off from Runway A only. Runway B is used exclusively for general aviation.

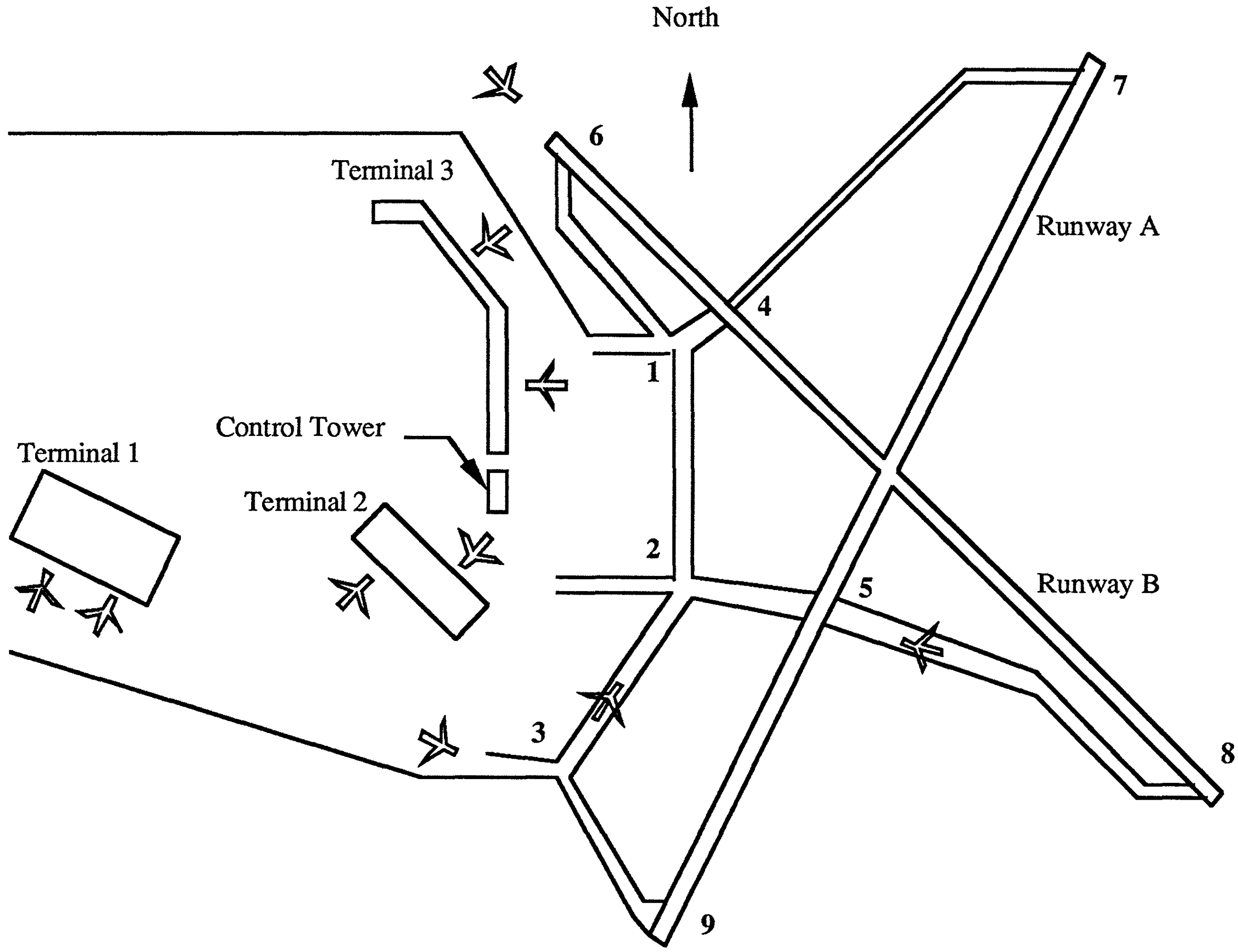

Fig. 6 Plan of Logan Airport

It is assumed that the division of Ground Control between two Ground Controllers is done on a geographic basis. One GC, hereafter called GC 1, is responsible for the southern sector of the airport, while the other one, GC 2, monitors the northern sector. Terminals 1 and 2 are monitored by GC 1, while Terminal 3 is monitored by GC 2. Nine areas have been labeled on Figure 6, which indicate critical spots of the ASTC system. Intersections 1, 2, and 3 designate dangerous crossings between taxiways. Intersection 1 lies within the jurisdiction of GC 2 while intersection 3 is under GC 1's. Intersection 2, however, stands on the boundary between the northern and southern sectors, and is monitored by both GCs. Intersections 4 and 5 indicate crossings between a taxiway and a runway. Intersection 4 is monitored by GC 2, while Intersection 5 is monitored by GC 1. If a plane on a taxiway approaches 4 or 5, a Ground Controller must interact with the Local Controller to get the status of the runway, and authorize or deny the crossing. Finally, intersections 6, 7, 8, and 9 designate the ends of the runways, the point at which a plane changes jurisdiction between Ground Control and Local Control.

Five sources of information can trigger variable patterns of interaction in the system:

- Sensor 1 : Wind. Wind is a parameter that influences the direction of landings and takeoffs. This source can take five values: X1 = {0, E, N, S, W}. N (S, E, W respectively) indicates that the wind comes from the North (South, East, West, respectively). 0 models low wind conditions in which both runways can be used.
- Sensor 2 : Runway Status. This source of information models the movements on the runways. Its output alphabet is X2 = {LAA, LAB, TOA, TOB, CL}, where LAA indicates that a plane is landing on Runway A, LAB indicates that a plane is landing on Runway B, TOA the fact that a plane is taking off from runway A, TOB, the fact that a plane is taking off from Runway B, and CL the possibility of the runways being clear of any movement. Note that one and only one runway is used if the output of Sensor 1 is E, N, S or W. Either runway can be used under low wind conditions, but safety standards require that there cannot be simultaneous use of both runways.
- Sensor 3 : Terminals 1 and 2. This source models a plane leaving Terminal 1 or Terminal 2. Its output alphabet is X3 = {DP1, NDP1} where DP1 models a departure from Terminal 1 or Terminal 2, and NDP1 indicates that no plane is departing. Note that a departing plane must be directed to a runway. If either runway is available, Ground Controller 1 must ask the Local Controller where to direct the plane.
- Sensor 4 : Terminal 3. Similarly, this source models a plane leaving Terminal 3. Its output alphabet is X4 = {DP3, NDP3}, where DP3 models the case in which one plane is leaving, and NDP3 the case in which no plane leaves. Note that GC2 must ask the Local Controller where to direct the plane in case of low wind conditions.
- Sensor 5 : Conflicts. This source of information models the existence of conflicts at the boundaries of the northern and the southern sectors. Its output alphabet is X5 = {C, NC} where C indicates that there is some conflict in the area that is monitored by both Ground

Controllers, whereas NC models the absence of conflict.

The set of inputs to the system is

$$X = X1 \times X2 \times X3 \times X4 \times X5.$$

The set of inputs contains 5 * 5 * 2 * 2 * 2 = 200 elements. Recall that these inputs model only the parameters that necessitate or influence coordination between roles. They do not describe completely the parameters that are processed by one role only.

System Requirements: Constraints are expressed to restrict the set of solutions to the design problem from AV to those AVSs that are relevant for ASTC. These constraints are translated into the language of Colored Petri nets on Fig. 7. To ease the readability of the net, a link that has been ruled out has not been represented, a link that is permanent is drawn with a bold line without annotations, a link whose variability is imposed has been drawn with a bold line, and a degree of freedom of the design is depicted by a plain line.

- Constraints on S: It is assumed that each role knows the direction of the wind, the output of Sensor 1. The output of Sensor 2, the status of the runway(s), is known by LC only. The output of Sensor 3, the status of Terminals 1 and 2, is monitored by GC 1, but not by GC 2. LC may or may not have access to that source of information, for example, by looking at departure strips on a monitor or by knowing in advance the planning of departures. Similarly, the output of Sensor 4, that is the status of Terminal 3, is monitored by GC 2, may or may not be known by LC, and is not surveyed by GC 1. Finally, the conflicts at the boundaries of the northern and southern sectors, the output of Sensor 5 are only known to the GCs.
- Constraint on s: Every role has to produce a response to be the sent to the planes under its jurisdiction, and each role must have a fixed link from its RS stage to the effectors.
- Constraints on F: No constraints are imposed on sharing information between Ground Controllers or from a GC to LC. Some constraints are imposed on the exchange of information from LC to the GCs, because LC must inform the GCs of the Runway Status. There is no need for fixed interactions because the utilization of the runways does not always create conflicts or dangers in both the northern and the southern sectors. If the wind is N or W, the landing planes leave the runway at 6 or 7, under the jurisdiction of GC 2, and head for the terminals through intersections 1 and 4, all monitored by GC 2. The departing planes take off from 8 or 9. The way they follow on the taxiways to intersections 8 and 9 is mainly under the jurisdiction of GC 1, and they cannot cross the runway that is used. Therefore, GC 2 only needs to be informed by LC. Symmetrically, if the wind is E or S, GC 1 needs to be informed only by LC. If Runways A and B are in use, however, both GCs must be informed, because each GC monitors an intersection (4 or 5) between a runway and a taxiway.

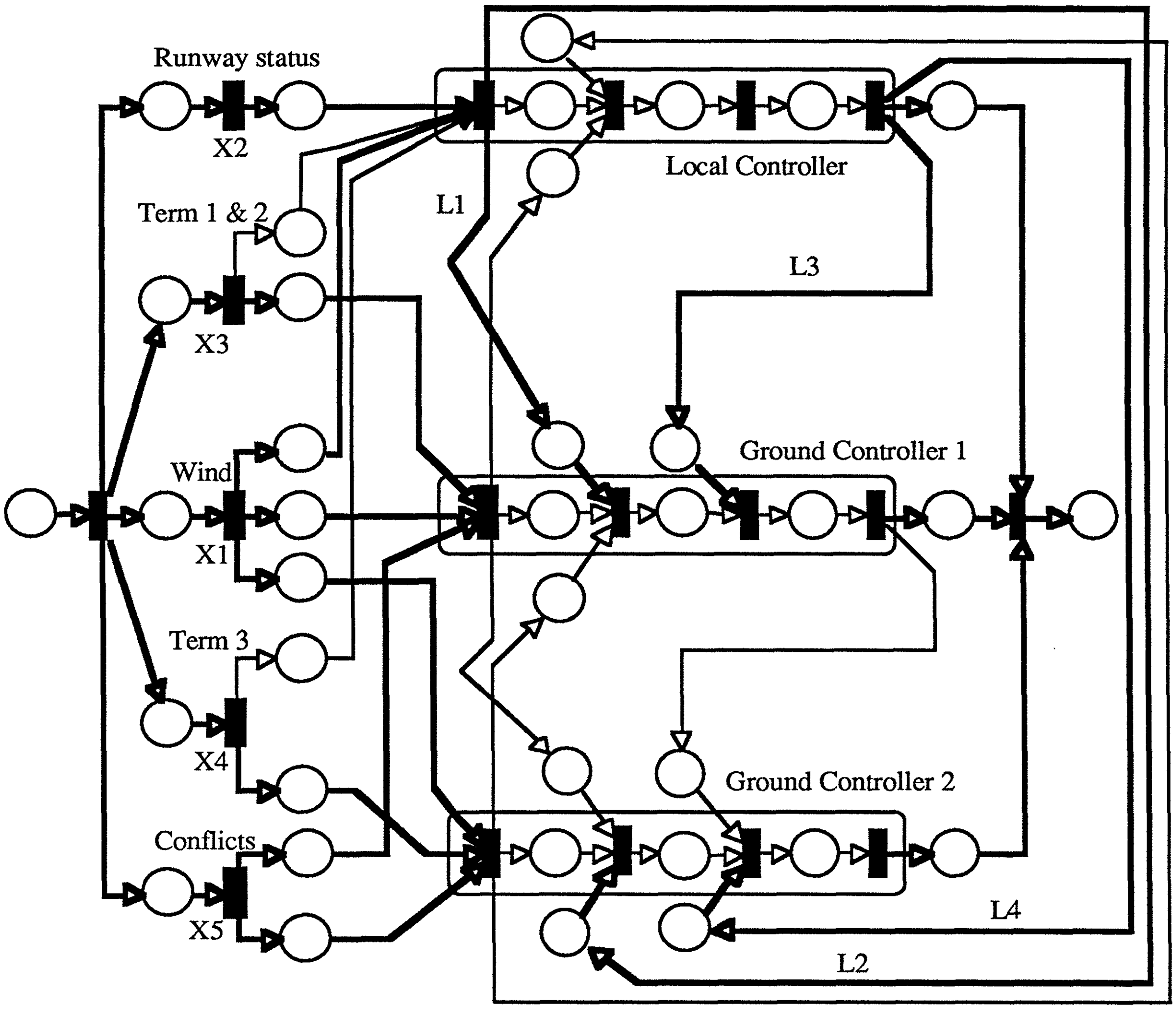

Fig. 7 Colored Petri Net model of the Constraints

These constraints can be summed up by stating that the interaction between the SA stage of LC to the IF stage of GC1 is activated by a matrix L1. The non null entries of L1 correspond to the case x1= E or S or 0. Similarly, the interaction between the SA stage of LC to the IF stage of GC 2 correspond to the the case x1 = N or W or 0.

- Constraint on G: Every SA stage receives some sensors' observations, therefore all links between RS stages and SA stages must be ruled out.
- Constraint on H: All links from the RS stages to the IF stages have been ruled out. The rationale is that a output of a role is a command to be executed, a resolution of a conflict or an answer to a request. Therefore, the communications of responses to roles are most appropriately modeled by the links from the RS stages to the CI stages.
- Constraint on C: It is assumed that GCs cannot issue commands to LC because the movements

on the runways have priority as far as safety is concerned. Second, GC 2 cannot issue a command to GC1, whereas GC 1 may issue one to GC 2. GC 1 covers a larger geographical area than GC 2, and may need to restrict the courses of action of GC2 to solve the conflicts in its own jurisdiction. Finally, in the case of low wind conditions, LC must inform the GCs when a plane leaves a terminal. The link from the RS stage of LC to the CI stage of GC 1 is thus activated for x1 = 0 and x4 = DP3. This variability has been indicated by the matrix L3. Similarly, the link from the RS stage of LC to the CI stage of GC 2 is activated for x1 = 0 and x3 = DP1.This variability has been represented by matrix L4.

CHARACTERIZATION OF THE SET OF SOLUTIONS

A solution to the design problem is any AVS that satisfies the user-defined requirements. Application of the methodology proves (a) that the task of determining the set of solutions can be carried out quantitatively using the formalism of the CPN theory and (b) that the set of solutions can be characterized without having to do a computationally expensive, and practically infeasible, exhaustive enumeration. The set can be determined from its boundaries, i.e., its minimal and maximal solutions. A solution Π_0 is *minimal* in V if it is not possible to have a variable structure Π, with Π ACT Π_0, without violating one of the constraints R1 to R9. A solution Π^0 is *maximal* in V if it is not possible to have a variable structure Π, with Π^0 ACT Π, without violating one of the constraints R1 to R9. The next propositions lead step by step to a characterization of the set of solutions.

Proposition 7: Consider the fixed structure, called the Universal Net, which contains all the interactions that have not been ruled out at the design specification stage. Then, for every x, $\Pi(x)$ must be a subnet of the Universal Net.

The Universal Net in the ASTC example is depicted on Fig. 8.

Then, an analysis of the user-defined constraints is performed to determine the correlation of the links' activations, as imposed by the user-defined requirements. This yields:

- A partition of the set of inputs X into k *elementary sets of inputs* EX_i, i = 1..k.
- A characterization of k disjoint subsets in W, W^i, i = 1..k.
- The condition that each input in EX_i must be assigned to a unique structure in W^i, Σ^i.

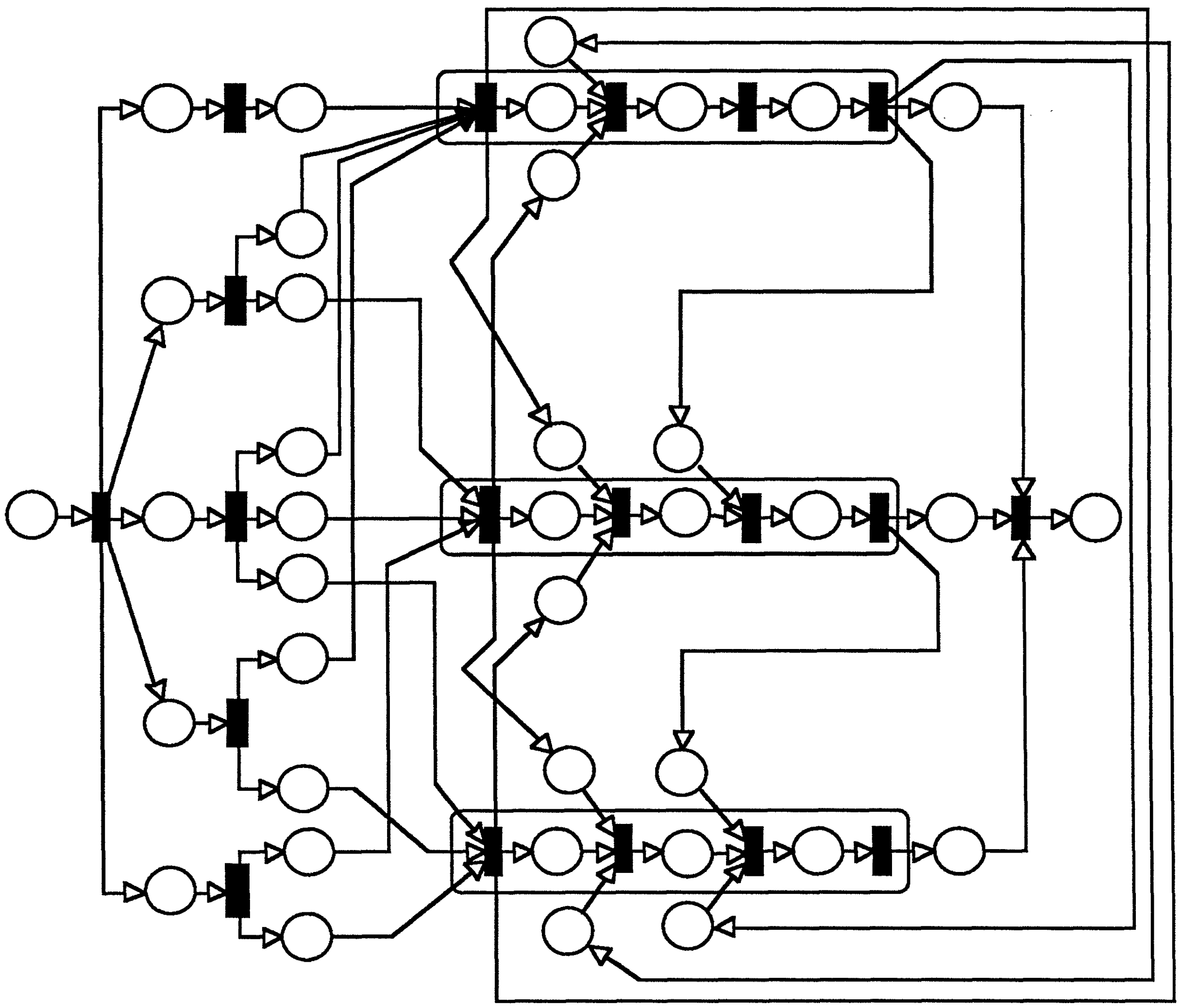

Fig. 8 Universal Net

In the ASTC example, there are six elementary sets of inputs, which correspond to different patterns of activation for Link 1, the link from SA of LC to IF of GC1; Link 2, the link from SA of LC to IF of GC2; Link 3, the link from RS of LC to CI of GC 1; and Link 4, the link from RS of LC to CI of GC2.

- EX_1 contains all the inputs with x1 = N or x1 = W. For those inputs, Link 1 must be activated and Links 2, 3, 4 must not be activated.
- EX_2 contains all the inputs with x1 = E or x1 = S. For those inputs, Link 2 must be activated, and Links 2, 3, 4 must not be activated.
- EX_3 contains all the inputs with x1 = 0, x3 = DP1, and x4 = DP3. For those inputs, Links 1, 2, 3, 4, must be activated.

- EX_4 contains all the inputs with x1 = 0, x3 = DP1, and x4 = NDP3. For those inputs, Links 1, 2, 4, must be activated, while Link 3 is not activated.
- EX_5 contains all the inputs with x1 = 0, x3 = NDP1, and x4 = DP3. For those inputs, Links 1, 2, 3, must be activated, while Link 4 is not activated.
- EX_6 contains all the inputs with x1 = 0, x3 = NDP1, and x4 = NDP3. For those inputs, Links 1, 2, must be activated, while Links 3, 4 are not activated.

The elements of every subset W^i, i = 1..k, can be characterized using *simple information flow paths*. A simple information flow path is a directed path without loops from the external place of the Universal Net to the sink.

Proposition 8: Σ is a fixed structure that belongs to W^i, i = 1..k, if and only if Σ lies between a maximal element and a minimal element in W^i:

$$\exists\ \Sigma_1 \text{ and } \Sigma_2 \quad \text{such that} \quad \Sigma_1 \text{ SUB } \Sigma \text{ SUB } \Sigma_2$$

and Σ is a union of simple information flow paths of the Universal Net.

A candidate solution is an AVS that satisfies all constraints of the design but R6 to R9. Candidate solutions are characterized by Proposition 9.

Proposition 9: An AVS verifies all constraints of the design but R6 to R9 if and only if Π lies between a maximal candidate solution and a minimal candidate solution :

$$\exists\ \Pi_1 \text{ and } \Pi_2 \quad \text{such that} \quad \Pi_1 \text{ ACT } \Pi \text{ ACT } \Pi_2$$

and Π_1 is such that for every i, i = 1..k,

$$EX_i \text{ --> } \Sigma^i, \text{ with } \Sigma^i \text{ minimal element in } W^i$$

while Π_2 is such that for every i, i = 1..k,

$$EX_i \text{ --> } \Sigma^i, \text{ with } \Sigma^i \text{ maximal element in } W^i$$

The set of AVS that satisfy all constraints of the design but R6 to R9 is completely determined by its boundaries. Unfortunately, all AVS that satisfy Proposition 9 do not fulfill constraints R6 to R9. The first reason is that candidate minimal and candidate maximal solutions may not fulfill one of the constraints R6 to R9. The second reason is that variable structures between the candidate maximal and the minimal solutions have variable links in which some effective alphabets are not accessible. However, the set of solutions is completely determined by its minimal and maximal solutions, as described by Prop. 10.

Proposition 10: The set of solutions can be divided into families of solutions. A family contains all the structures that have the same permanent input links. One family corresponds to a layer of partially ordered sets. In each layer, the AVS have the same accessible alphabets. The layers are sorted in increasing number of accessible alphabets. Within one family, Π is a solution if and only if Π fulfills Proposition 9; Π is bounded by at least one minimal and one maximal solutions that have the same accessible alphabets:

$$\Pi_1 \text{ ACT } \Pi \text{ ACT } \Pi_2.$$

and the boundaries of each layer are determined by the maximal and minimal elements of the family.

The constraints that are specific to variable structures require that the set of solutions be divided into subsets of solutions with the same input links (Constraints R6, R7, and R8). The minimal and maximal solutions determine completely the set of solutions within one family. Between a minimal solution and a maximal solution there are several layers of solutions. Each layer has boundaries, and any AVS 1) between the boundaries, 2) whose variability is based on the corresponding accessible alphabets, is a solution to the design problem. Finally, the boundaries are determined by adding selected links of the maximal solution to the minimal solution.

There are four families for the ASTC system. As indicated by Proposition 10, each subset corresponds to a particular combination of input links. In addition, in this example, there exists a unique minimal element, a unique maximal element, and two and only two layers in each family. The first family corresponds to the AVS in which LC does not have direct knowledge of the departures at the terminals. The second family corresponds to the structures in which LC has direct knowledge of planes departing from Terminals 1 and 2, but not of the departures from Terminal 3. The third family corresponds to the structures in which LC has direct knowledge of planes departing from Terminal 3, but not of the departures from Terminals 1 and 2. The fourth and final subset corresponds to those structures in which the LC has direct knowledge of planes departing from Terminals 1, 2, and 3. Only this family of solutions is presented here, because of length requirements. The unique minimal solution is represented in Fig. 9, the maximal solution is depicted in Fig. 10, and the intermediate boundary is shown in Fig. 11.

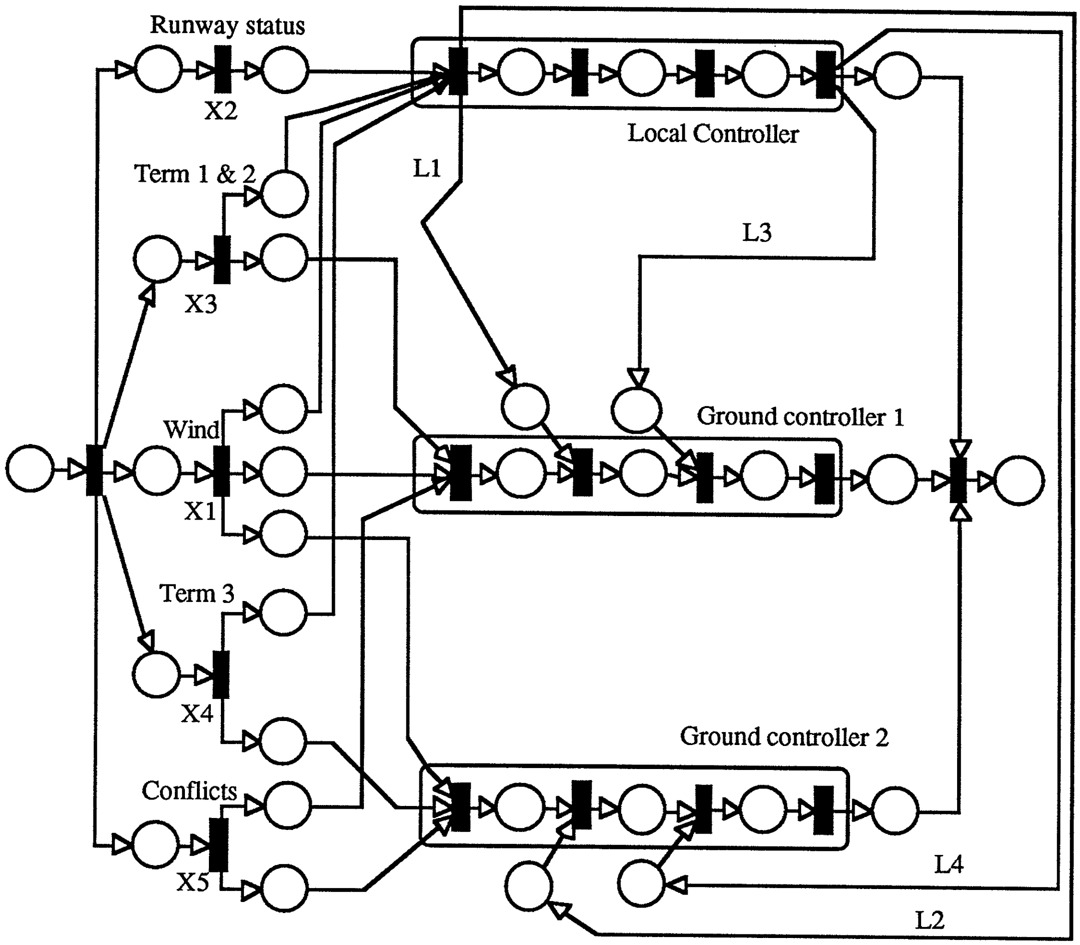

Fig. 9 Minimal Solution

Five links constitute the degrees of freedom from the minimal solution to the maximal solution:

- Link A, the link from the SA stage of GC1 to the IF stage of GC 2.
- Link B, the link from the SA stage of GC2 to the IF stage of GC1.
- Link C, the link from the RS stage of GC1, to the IF stage of GC2.
- Link D, the link from the RS stage of GC 1 to the IF stage of LC.
- Link E, the link from the RS stage of GC 2 to the IF stage of LC.

As stated, there are two layers of solutions. The higher layer is composed of those structures that are above the intermediate boundary, while the lower layer is composed of those that do not. If Π is above the intermediate boundary, then the variability of Links A and B can be based on the observations of Sensor 1 exclusively; Link C can based on the observations of Sensors 1, 3, and 4 exclusively; Link D can be based on the observations of Sensors 1 and 3 exclusively; and Link E can be based on the observations of Sensors 1 and 4 exclusively. If Π is above the minimal solution without having

Intermediate Boundary ACT Π

then Links A, B and C can be based on the observations of Sensor 1 exclusively; Link D can be based on the observations of Sensors 1 and 3 exclusively; and Link E can be based on the observations of Sensors 1 and 4 exclusively.

The difference between the two layers relates to the alphabets that can be accessed to coordinate a variable interaction at link C. The structures in the lowest layer can only coordinate a variable interaction at link C based on the direction of the wind, while the structures in the second layer, can base the variability on the planes that leave the terminals as well as the wind. The structures in the lowest layer are less flexible between ground Controllers; on the other hand, each GC processes less information, and might thus perform its role more efficiently.

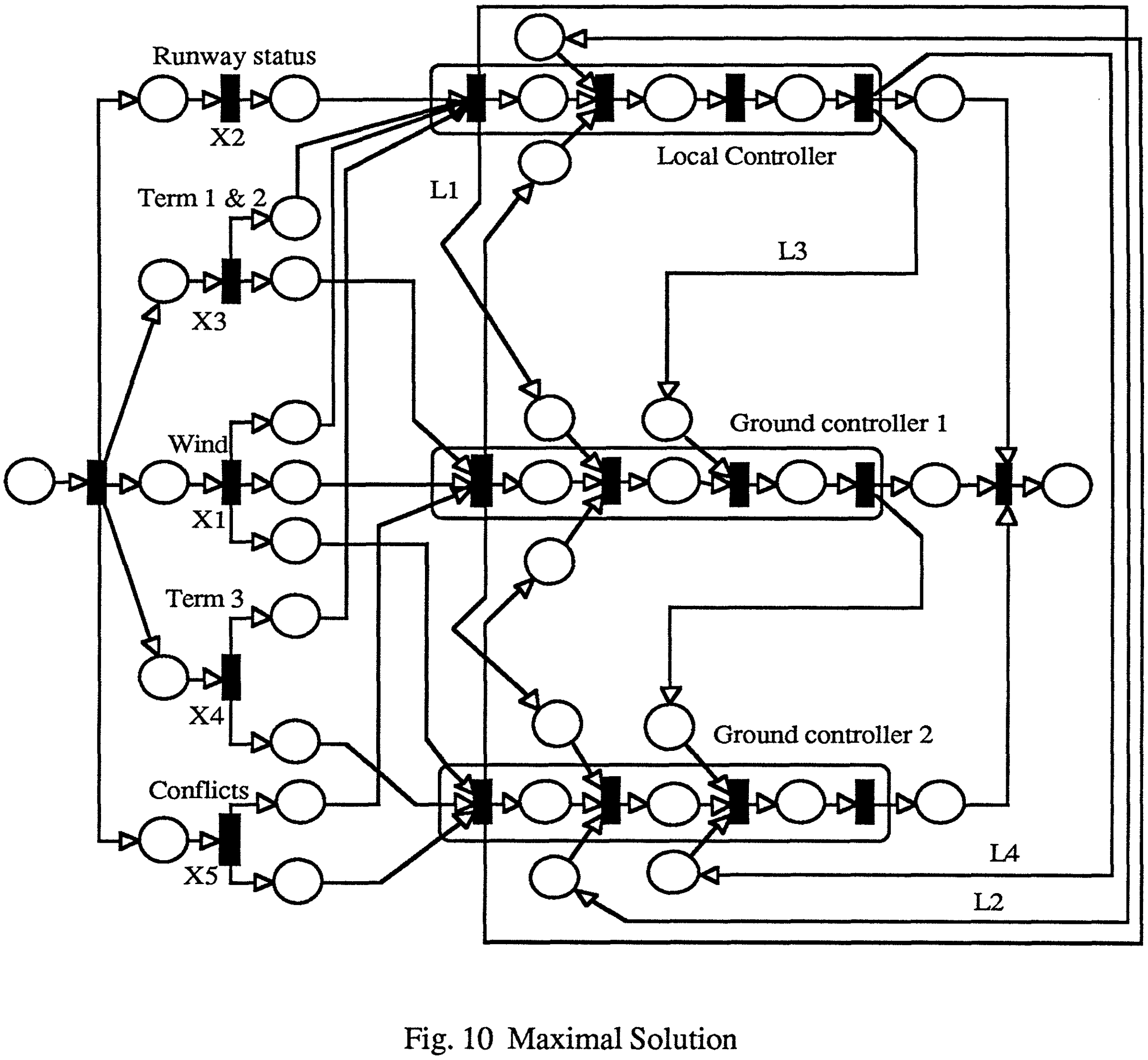

Fig. 10 Maximal Solution

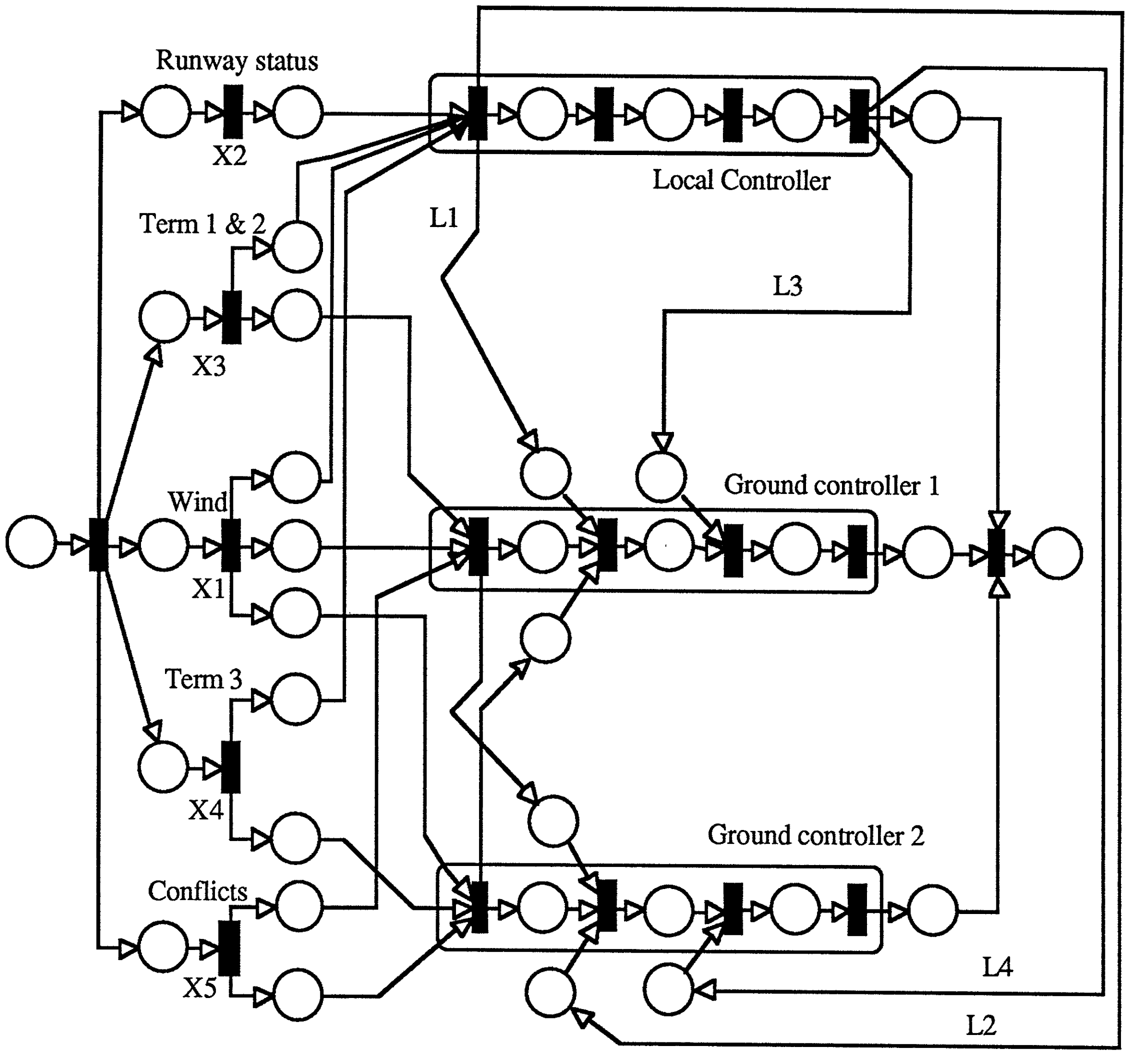

Fig. 11 Intermediate Structure

This family of solutions to the ASTC design problem can be interpreted as follows. If Local Control knows when planes leave terminals, then there are many ways to design an organization that coordinates its tasks correctly, as indicated by the degrees of freedom on the activation of Links A to E. Each Ground Controller can have a variable interaction with Local Control that is based both on the wind direction and on the departures from the terminals. Similarly, the Ground Controllers can have many interactional patterns. Furthermore, the existence of two layers shows that the issuance of an advisory regarding a command by GC1 to GC2 can be elaborated either on one sensor, or three sensors. The framework quantifies precisely, through the layers, the sources of information on which variable interactions can be based. Finally, the methodology leaves open to the designer the choice of a particular structure in this family. It describes exclusively the parameters (minimal and maximal elements, intermediate boundaries) that characterize the set of solutions.

CONCLUSION

In this paper, a quantitative methodology for modeling variable structure decisionmaking organizations and for characterizing the set of solutions to a particular organizational design problem is presented. The class of structures considered process deterministically a set of simultaneous observations. This methodology models variable structures with Colored Petri Nets, which are used as the basic mathematical framework to generate the set of structures that satisfy design requirements. The designer of a system can describe his degree of knowledge about the requirements in a matrix form that is translated into the language of Colored Petri Nets. Then, the set of variable structures that satisfy the designer's requirements as well as some generic constraints is characterized. This methodology provides a basic step towards the modeling and analysis of more realistic decisionmaking organizations.

REFERENCES

[1] Simon, H.A., *The Sciences of the Artificial*, The MIT Press, Cambridge, MA, 1981.

[2] Jensen, K., "Colored Petri Nets " in *Advanced Course on Petri Nets 1986*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, FRG, 1987.

[3] Reisig, W., *Petri Nets, An Introduction.* Springer-Verlag, Berlin, FRG, 1985.

[4] Minsky, M., *The Society of Mind.* Simon and Schuster, New York, 1986.

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

[6] Grevet, J.-L, and Levis, A. H., "Coordination in Organizations with Decision Support Systems". *Proc. 1988 Symposium on Command and Control Research.* Science Applications International Corporation, McLean, VA, 1988.

[7] Monguillet, J.-M., and Levis, A. H., "Modeling and Evaluation of Variable Structure Command and Control Organizations." *Proc. 1988 Symposium on Command and Control Research.* Science Applications International Corporation, McLean, VA, 1988.

[8] Remy, P. and Levis, A. H., " On the Generation Of Organizational Architectures Using Petri Nets," in *Advances in Petri Nets 1988*, Lecture Notes in Computer Science, G.Rozenberg Ed. Springer-Verlag, Berlin, FRG 1988.

[9] Genrich, H.J. and Lautenbach, K, "System modeling with High-Level Petri Nets," *Theoretical Computer Science*, No. 13, 1981, pp. 109-136.

[10] Demaël, J.J.,"On the Generation of Variable Structure Distributed Architectures," LIDS-TH-1869. Laboratory for Information and Decisions Systems, MIT, Cambridge, MA 1989.