(2)

November 1991

GMU/ C3I-112-TH

ON THE GENERATION OF MULTILEVEL DISTRIBUTED INTELLIGENCE SYSTEMS USING PETRI NETS

S. Abbas K. Zaidi

Center of Excellence in Command, Control, Communications, and Intelligence (C^3I)

George Mason University
Fairfax, Virginia 22030

November 1991 GMU/ C3I-112-TH

ON THE GENERATION OF MULTILEVEL DISTRIBUTED INTELLIGENCE SYSTEMS USING PETRI NETS

S. Abbas K. Zaidi

Center of Excellence in Command,
Control, Communications, and Intelligence (C^3I)

George Mason University
Fairfax, Virginia 22030

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1991

3. REPORT TYPE AND DATES COVERED
Technical Report, 1/90 to 11/91

4. TITLE AND SUBTITLE
On the Generation of Multilevel Distributed Intelligence Systems Using Petri Nets

5. FUNDING NUMBERS
N00014-91-J-1584
N00014-90-J-1680
19-920090-31 (SAIC)

6. AUTHOR(S)
S. Abbas Zaidi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Center of Excellence in Command, Control, Communications and Intelligence
George Mason University
Fairfax, VA.

8. PERFORMING ORGANIZATION REPORT NUMBER
GMU/C3I-112-TH

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
and
Science Applications International Corp.

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Complex distributed intelligence systems, characterized by the hierarchical arrangement of their subsystems, are described by families of structures, with each family concerned with the behavior of the system as viewed from a different level of abstraction. A methodology to model and generate multilevel hierarchical distributed intelligence systems is presented. A mathematical framework, based on Hierarchical Petri Net theory, is developed for representing the inter-actions among the objects comprising the system at the same level and across different levels. The methodology and the resulting models also provide a structured and modular way for solving the problem of designing large-scale C3 architectures by breaking a computationally large problem into smaller subproblems. The methodology is applied to two illustrative examples.

14. SUBJECT TERMS
Organization Design, Petri Nets, Distributed Intelligence Systems, C3 Architectures

15. NUMBER OF PAGES
215

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

November 1991 GMU/C3I-112-TH

ON THE GENERATION OF MULTILEVEL DISTRIBUTED INTELLIGENCE SYSTEMS USING PETRI NETS

By

Syed Abbas Kazim Zaidi

This report is based on the unaltered thesis of Syed Abbas K. Zaidi, submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of Master of Science at George Mason University in November 1991. The research was conducted at the C3I Center of George Mason University with support provided in part by the Office of Naval Research under contract no. N00014-91-J-1584 and contract no. N00014-90-J-1680 and in part by the Basic Research Group of the Joint Directors of Laboratories (JDL) under sub-contract no. 19-920090-31 from Science Applications International Corporation (SAIC).

Center of Excellence in C3I
George Mason University
Fairfax, Virginia 22030

ON THE GENERATION OF MULTILEVEL DISTRIBUTED INTELLIGENCE SYSTEMS USING PETRI NETS

by

Syed Abbas Kazim Zaidi

A Thesis Submitted to the
Faculty of the Graduate School
of
George Mason University
in Partial Fulfillment of
the Requirements for the Degree
of
Master of Science
Electrical and Computer Engineering

Committee:

Prof. Alexander H. Levis ________ Director

Prof. Harry L. Van Trees ________

Prof. David H. Schaefer ________

Date: 11/22/91

Fall 1991
George Mason University
Fairfax, Virginia

ABSTRACT

ON THE GENERATION OF MULTILEVEL DISTRIBUTED INTELLIGENCE SYSTEMS USING PETRI NETS

Syed Abbas Kazim Zaidi

George Mason University, 1991

Thesis Director: Dr. Alexander H. Levis

Complex distributed intelligence systems, characterized by the hierarchical arrangement of their subsystems, are described by families of structures, with each family concerned with the behavior of the system as viewed from a different level of abstraction. A methodology to model and generate multilevel hierarchical distributed intelligence systems is presented. The objects that comprise such a system and the generic interactions among them are defined. A mathematical framework, based on Hierarchical Petri Net theory, is developed for representing the interactions among these objects at the same level and across different levels. The methodology and the resulting models also provide a structured and modular way for solving the problem of designing large-scale distributed intelligence systems by breaking a computationaly large problem into smaller subproblems, thus reducing the computational effort required to generate the feasible solutions. The methodology is applied to two illustrative examples.

ACKNOWLEDGEMENTS

I wish to express my gratitude to:

Dr. Alexander Levis, for his dedicated parental supervision, and for his impeccable professional example. His high requirements and commitment to excellence - though they brought a number of sleepless nights, finally ended in a pleasant professional satisfaction, and that was surely worth waiting for. Working with Dr. Alexander Levis has been a broadening experience, professionally and above all personally.

Raza and Zeba Baji, who helped me make my early days in the United States possible and their encouragement all along the way.

C3I Center, for their continous support throughout this endevour. I started working as a Student Assistant in the administrative office of the C3I Center before becoming a Research Assistant.

The comrades of C3 Architectures Lab - Mike, David, Ali and Diwakar for their comradeship and advice.

Diwakar for his moral support I desperately needed a number of times, and the endless discussions we had over all topics ranging from politics to religion.

My friends, Zafar, Moid, Mujahid, Maqbool, and Zahid who assisted me in discovering how interactions are defined in a 3-level (townhouse) distributed intelligence system.

Khadija, my daughter, who was born in the same month I started working on my thesis.

My family, and finally to Zehra, for their continuous and devoted support.

This research was conducted at the Center of Excellence in C3I of George Mason University, with support provided in part by the Office of Naval Research under contract no. N00014-91-J-1584 and contract no. N00014-90-J-1680 and in part by the Basic Research Group of the Joint Directors of Laboratories (JDL) under sub-contract no. 19-920090-31 from Science Applications International Corporation (SAIC).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 MOTIVATION

A methodology for generating distributed intelligence system designs has been developed by Remy (1986). This methodology results in flat organizational structures, where the decision making unit is a human decision maker (DM). It generates organizational structures by determining the interactional structure of the organization, i.e., the interactions among human decision makers. The methodology is based on the concept of allowable interactions among organization members - decision makers -, and on the development of a mathematical framework to represent these interactions. The methodology was extended by Demaël (1989) to the design of variable structure distributed intelligence systems. There is a growing need for a methodology to generate in some orderly manner, either by using the existing algorithms iteratively or by some new algorithm, organizational structures where the decision making units could be either decision makers or suborganizations with different internal structures. Such a methodology should also be able to generate organizational structures at an arbitrary level of abstraction, and be capable of describing the system's architecture at different degrees of detail.

1.2 PROBLEM DEFINITION

Three main problems need to be addressed to implement properly such a methodology.

(a) A mathematical framework that is appropriate for the formulation of the design problem should be identified.

(b) The concept of multilevel hierarchical organizational structures needs to be formulated analytically.

(c) Sets of constraints, structural and user-defined, have to be identified for different levels in the organization to keep the problem of generating organizational structures computationaly feasible.

This effort will fill another gap between availability of analytic tools and the real-world design issues for large-scale distributed intelligence systems. The designer will have to specify the entire organization in terms of its subsystems, defined at a given degree of abstraction; then all subsystems are defined, if possible, in terms of their subsystems and so on. The lowest, as well as the highest, degrees of detail that are desired to be used in describing organizational structure need to be specified. Requirements will then be specified for each suborganizational structures in terms of the interactions among the subsystems of the suborganizational structure at a given level in the organization. Once the requirements for an organizational or suborganizational structure at a given *stratum* (level) in the organization are specified, the designer will be able to choose a structure from a number of candidate structures, all fulfilling the requirements. Once all the structures for all the subsystems of the organization are determined, the entire organizational structure can be described at an arbitrary level of detail - the latter being bounded by the lowest and the highest levels of abstraction used in the design.

1.3 THEORETICAL BACKGROUND

A quantitative methodology for modeling, designing and evaluating fixed structure distributed systems has been developed at the MIT Laboratory for Information and Decision Systems by Remy (1986), Andreadakis (1988), and Demaël (1989). In this work an organization is considered as a system performing a task; the system is modeled as an interconnection of organization members. Each organization member is represented by a multi-stage model. Each stage represents a well defined procedure or algorithm that a decision maker can perform.

In Remy (1986), a framework was presented which allows designers to express their design problems in mathematical terms. Then, an algorithm was developed that makes it possible to characterize and generate all feasible organization structures in terms of partially ordered sets of fixed structures that satisfy both structural and designers' requirements.

Monguillet (1986) formalized the notion of variable structure decision making organizations and introduced the use of High Level Nets - Colored Petri Nets -, to model certain types of variability.

Demaël (1989) extended the earlier work by Monguillet (1986) and developed a methodology for modeling and generating variable structure distributed intelligence systems. He presnted a mathematical framework for modeling systems that adapt their structure of interactions to the input they process. The methodology used the language of Colored Petri Nets to describe the architectures.

1.4 GOALS AND CONTRIBUTION

This research will present a major extension of the earlier work by addressing the problem of designing *multilevel* hierarchical organizational forms. The work will be a direct extension of the work done by Remy (1986).

In this thesis, a mathematical model of interactions among suborganizations at different levels is defined. This model is an extended version of the existing model by Remy (1986). The model allows the designer to first determine the levels of organization being considered. The subsystems of the organization at different levels are specified in terms of their subsystems. At the lowest level, *stratum 'N'*, the decision making unit is a human decision maker with a five stage structure, (Levis, 1991). At all other levels, the decision making units are suborganizations. Figure 1.1 presents a block diagram representation of a multi-level organization. Depending on the particular level chosen, the designer is required to characterize with an arbitrary degree of precision the class of interactions among the decision making units comprising a system or a subsystem. The specificity of the designer's requirements determines the degrees of freedom left. Lattice theoretic results are used to define a partial order among all allowable organizational structures belonging to a system or a subsystem; then the set of all allowable organizational structures of the given system or subsystem is characterized by its boundaries.

The mathematical formulation of the problem is based on Petri Net theory. All the allowable structures will be translated into Petri Net representations. The set of all allowable organizational structures can then be analyzed and a particular organizational structure can be chosen as a result of a comparison of performance with respect to some designer-defined criteria.

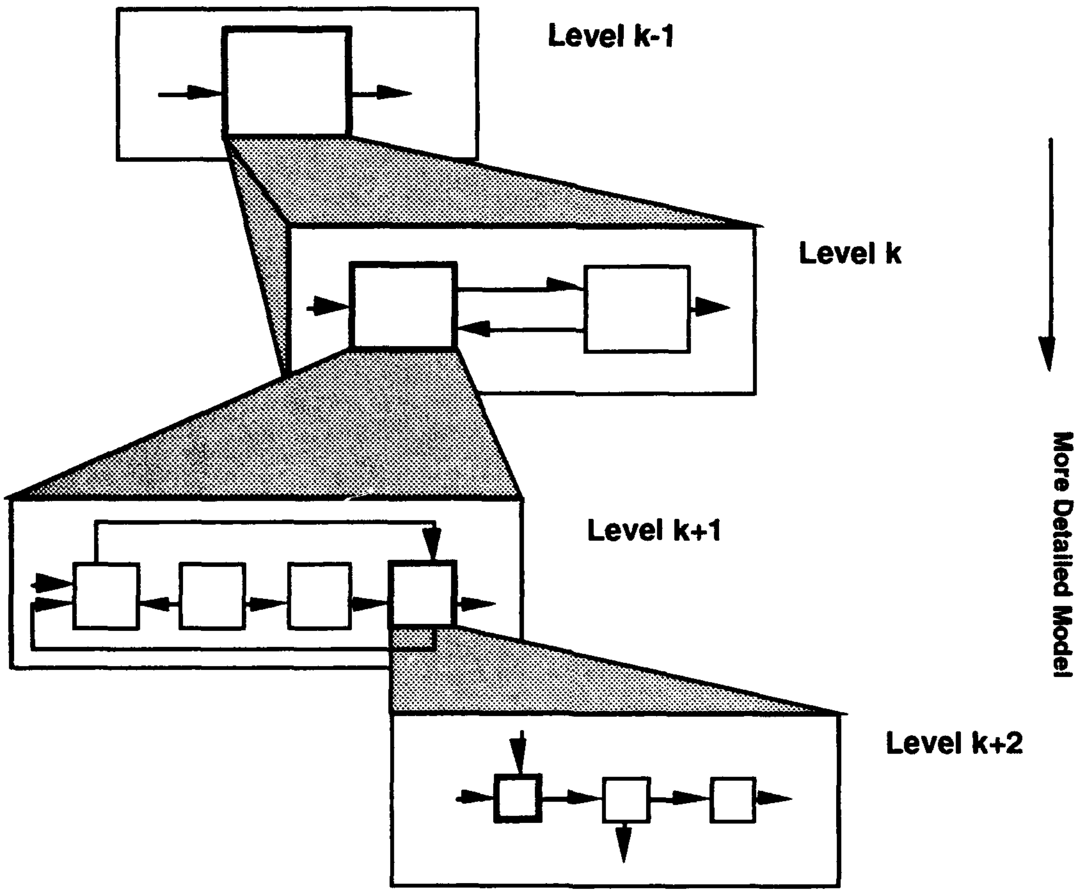


Figure 1.1 A Multi-level Organization

The entire organization is described in terms of its subsystems. The organizational structures associated with the subsystems of the organization are *folded* or *unfolded* to represent the system's architecture at different levels. A set of connectivity rules are formulated to translate interactions among subsystems of the organization defined at a given level to their lower level representations. The interactions that exist at a higher level of abstraction are translated to their more detailed description whenever an organization is *unfolded* to a more detailed representation. The connectivity rules are based on the concept of a multiechelon hierarchy; the hierarchical relationships are formulated on the basis of messages that flow to and from the decision making units.

1.5 THE THESIS IN OUTLINE

The thesis is organized as follows. Chapter II is a review of Petri Net theory: the basic notions are reviewed together with some some advanced topics that will be used throughout the thesis. Chapter III is a review of Lattice Theory: it presents the formalism used in subsequent chapters. In Chapter IV, an introduction to the methodology for generating multilevel organizational structures is presented. Chapter V describes the translation of this methodology into the language of Hierarchical Petri Nets. In Chapter VI, additional constraints are introduced that will define the concept of valid organizational form. Chapter VII addresses the problem of representing connectivity of a higher level interaction when it is defined at a lower level. In Chapter VIII, a review of the Lattice algorithm is presented. The algorithmic implementation of the overall methodology is presented in Chapter IX where an application is also given. Finally, Chapter X contains conclusions and suggestions for further research.

CHAPTER II

PETRI NET THEORY

This chapter is an introduction to Petri Net theory. In this chapter the basic formalism of Ordinary Petri Nets is presented. One extension of the theory that overcomes some limitations of ordinary Petri Nets is High Level Nets. Two models have been developed within that approach, Predicate Transition Nets and Colored Petri Nets. The concepts of High Level Nets are not presented in this chapter as the theory and results developed in this thesis do not require them. However, the concept of Hierarchical Ordinary Petri Nets is presented as it is the key concept used throughout the thesis. More introductory material can be found in Peterson (1981), Brams (1983), and Reisig (1985). High Level Nets have been described in Genrich and Lautenbach (1981). Advanced materials on Predicate Transition Nets are provided in Genrich (1987) and Monguillet (1988). Similarly, advanced material of Colored Petri Nets can be found in Jensen (1987) and in Demaël (1989).

2.1 INTRODUCTION

Large-scale distributed systems have certain characteristics:

They exhibit *concurrency* or parallelism. Several components can work at the same time on the same task. There is thus a need to represent the precedence relations between the processing of the different components.

These systems very often offer *choices*. One process may be done by several components, or several combinations of components. Conversely, a particular component is usually able to perform different types of processes.

A choice may create a *conflict*.

The operations executed by the various components are *asynchronous*. There are no global mechanisms that coordinate the scheduling of the processes. Each component usually starts its processing as soon as it has received all the information it needs. If several tasks are requested, a queuing discipline (First In First Out (FIFO), Last In First Out (LIFO), etc...) is enforced to schedule the individual requests.

Complex systems need *representations* that are easy to use and review. Graphical models address some of these issues. Complex systems also demand the quality of *verifiability* in models. That is, system model should be capable of revealing their logic when analyzed and allow performance analysis and simulation. Petri Nets have been introduced in the modeling of Distributed Systems because they give a graph-theoretic representation of the communication and control patterns, and a mathematical framework for analysis and validation. Petri Net modeling is appealing for the following reasons:

- Petri Nets provide an integrated methodology, with well developed theoretical and analytical foundations, for modeling physical systems together with complex (cognitive) decision processes.

- Petri Nets capture the precedence relations and structural interactions of concurrent and asynchronous events. Deadlocks and conflicts can be easily identified on a Petri Net .

- The graphical nature of Petri Nets helps to visualize easily the complexity of the system. They are thus appealing both to the layman and to the analyst.

- Various extensions of the basic theory allow for quantitative analysis of resource utilization, throughput rate, effect of failures, and real time implementation.

- The additional property of *executability* makes the Petri Net a powerful modeling language.

2.2 ORDINARY PETRI NETS

2.2.1 Definitions

Definition 1.1

An Ordinary Petri Net is a bipartite directed graph: (P, T, I, O).
There are two sets of nodes:

- P = {p1, ..., pn} a finite set of *places*.
  A place is depicted by a circle node.

  A place models a resource, a buffer, or a condition.
- T = {t1, ..., tm} a finite set of *transitions*.
  A transition is represented by a bar node.

  A transition stands for a process, an event, or an algorithm.

- The *arcs* or connectors that connect these nodes are directed and fixed. They can only connect a place to a transition, or a transition to a place. They are given by:

- I : P × T -> {0,1}

  I is an input function that defines the set of directed arcs from P to T.
  I(p,t) = 1 if the arc exists, I(p,t) = 0 otherwise.
  An arc from a place p to a transition t indicates that the process t requires the availability of the resource p, the fulfillment of the condition p, or the availability of information in the buffer p, in order to occur.

- $O : P \times T \rightarrow \{0,1\}$

  O is an output function that defines the set of directed arcs from T to P. O(p,t) = 1 if the arc exists, O(p,t) = 0 otherwise.
  An arc from a transition t to a place p indicates that when the process t is finished, it either enables the condition p, makes the resource p available, or sends an item of information to the buffer p.

Example 2.1: Consider the Ordinary Petri Net shown in Figure 2.1

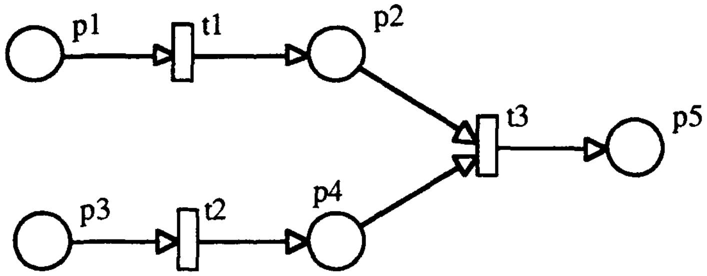


Figure 2.1 Ordinary Petri Net

The set of places P, the set of transitions T, and the input and output functions that define the arcs for this net are:

P = { p1, .. p5} T = {t1, t2, t3}
I(p1, t1) = I(p2, t3) = I(p3, t2) = I(p4, t3) = 1 I(p, t) = 0 otherwise.
O(p2, t1) = O(p4, t 2) = O(p5, t3) = 1 O(p, t) = 0 otherwise.

Definition 2.2

A Petri Net is *pure* if and only if it has no self loop, i.e., no place that can be both an input and an output of the same transition.

The net of Fig. 2.1 is pure. All Petri Nets that are considered in this thesis are pure. For an extensive discussion of this modeling issue, see Hillion and Levis (1986).

Definition 2.3

A *path* is a set of k nodes and k - 1 connectors, for some integer k, such that the i-th connector either connects the i-th node to the i+1-th node or the (i + 1)-th node to the i-th node. The path is directed if the i-th connector connects the i-th node to the (i + 1)-th node for all i = 1,..k..

Example 2.2: In Figure 2.1

p3 - t2 - p4 - t3 - p5 is a directed path,
p5 - t3 - p2 is not a directed path.

If a Petri Net has sources and sinks, then any path from a source to the sink is called an information flow path. If an information flow path is a set of k nodes such that the k nodes are distinct, then the information flow path is said to be simple. The path p1 - t1 - p2 - t3 - p5 is, for example, a simple information flow path of the Petri Net of Figure 2.1.

Definition 2.4

A Petri Net is *connected* if and only if there exists a path - not necessarily directed - from any node to any other node.

Fig. 2.1 depicts a connected net. Intuitively, this definition formalizes the idea that a Petri Net models a whole system. There are no partitions of the set of nodes into disjoint subsets, such that the nodes in one subset are not connected to the other subsets.

Definition 2.5

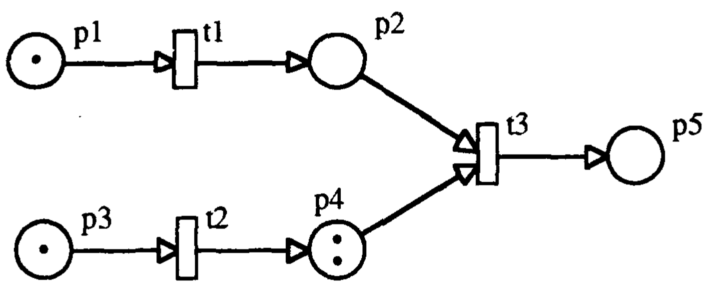
A Petri Net is *strongly connected* if and only if there exists a directed path from any node to any other node.

The net of Fig. 2.1 is not strongly connected because, for example, there is no directed path from p1 to p2.

2.2.2 Petri Nets with Markings

A Petri Net can contain tokens. Tokens are depicted graphically by indistinguishable dots (•), and reside in places. The existence of one or more tokens represents either the availability of the resource, or the fulfillment of the condition, or the number of items of information in the buffer. The travel of tokens through the net is controlled by the transitions. A marking of a Petri Net is a mapping M that assigns a non negative integer (the number of tokens) to each place.

Example 2.3: Consider the Petri Net in Fig. 2.2 with the indicated marking.



M(p1) = M(p3) = 1; M(p4) = 2; M(p2) = M(p5) = 0.

Figure 2.2 Petri Net with Marking

It is the same net shown in Fig. 2.1. The single token in the place p1 indicates the availability of a resource for the process modeled by transition t2. Similarly, the two tokens in place p4 represent the availability of two resources or input conditions for the execution of process modeled by t3.

Definition 2.6

A transition is *enabled* by a marking, if and only if all of its input places contain at least one token provided each input arc represent a single connection between the place and the transition.

In Example 2.3, t1 and t2 are enabled. All the conditions to be satisfied are fulfilled.

Definition 2.7

An enabled transition can *fire*. The firing of the transition corresponds to the execution of the process or the algorithm. The dynamical behavior of the system is embedded in the changes of the markings, when the firing takes place, a new marking is obtained by removing a token from each input place and adding a token to each output place.

Example 2.4: In Fig. 2.2, if t1 fires, then the resulting marking is shown in Figure 2.3.

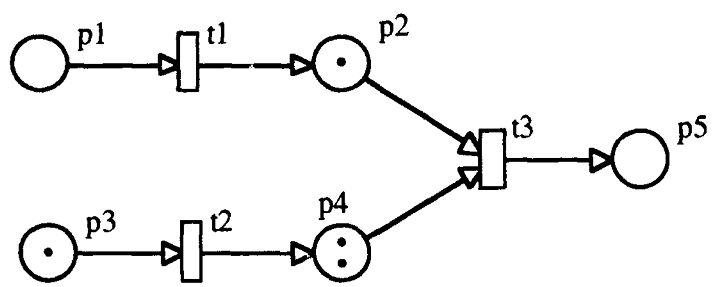


Figure 2.3 Petri Net after Firing

Transitions t3 and t2 are now enabled. If t3 fires, the new marking is shown in Figure 2.4.

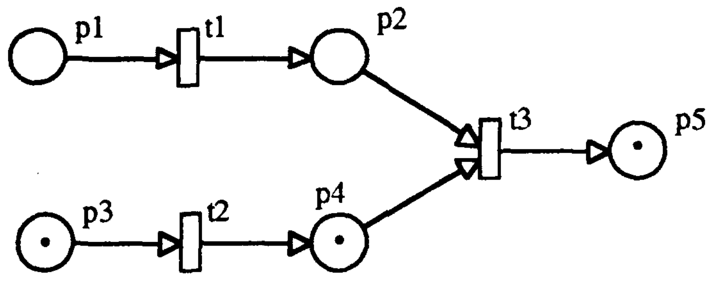


Figure 2.4 Petri Net after Second Firing

Remark: A transition may fire concurrently more than one token, i.e., a process may handle several tasks at the same time. Each firing of a transition is thus characterized by an integer k, the firing pattern of the transition. A transition can fire according to the firing pattern k, if and only if all of its input places have at least k tokens. When the firing takes place, k tokens are removed from each input place, and k tokens are added to each output place. The firing pattern is 0 if a transition does not fire.

2.2.3 Linear Algebraic Approach

So far, Petri Nets have been described as graphs. An alternative and equivalent approach can be developed using linear algebra with integer coefficients (Memmi and Roucairol, 1980).

Definition 2.8

A Petri Net with n places and m transitions can be represented by a $n \times m$ matrix C, the *Incidence Matrix*. The rows correspond to places, the columns correspond to transitions.

- Cij = 1 if there is a directed arc from the j-th transition to the i-th place. "1" indicates that the firing of the j-th transition adds one token to the i-th place.

- $C_{ij} = -1$ if there is a directed arc from the i-th place to the j-th transition. "-1" indicates that the firing of the j-th transition removes one token from the i-th place.

- $C_{ij} = 0$ if there is no arc from the j-th transition to the i-th place.

Example 2.5: The incidence matrix of the net on Fig. 2.1 is

$$C = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{matrix} \quad \text{(columns } t1 \;\; t2 \;\; t3\text{)}$$

Properties

- The marking of a net can be represented by a $n \times 1$ vector M, where $M_i = M(p_i)$. *The i-th entry corresponds to the number of tokens in the i-th place.*
- The firing pattern of the net can be represented by an $m \times 1$ firing vector F, where F_j is the firing pattern of the i-th transition.

- Given an incidence matrix C, an initial marking M, and a firing pattern F, the new marking M' is

$$M' = M + C * F. \qquad (2.1)$$

Example 2.6: The matrix equation that corresponds to the firing of Fig. 2.3 is

$$M' = \begin{bmatrix}1\\0\\1\\2\\0\end{bmatrix} + \begin{bmatrix}-1 & 0 & 0\\1 & 0 & -1\\0 & -1 & 0\\0 & 1 & -1\\0 & 0 & 1\end{bmatrix} * \begin{bmatrix}1\\0\\0\end{bmatrix} = \begin{bmatrix}0\\1\\1\\2\\0\end{bmatrix} \qquad (2.2)$$

2.2.4 Invariants

An incidence matrix makes it possible to use results from linear algebra to infer properties of the net. Much of the literature is devoted to the study of S- invariants.

Definition 2.9

Given an incidence matrix C, an S-invariant is a $n \times 1$ non-negative integer vector X of the kernel of C^T, i.e.,

$$C^T * X = 0 \qquad (2.3)$$

Remark: One must pay particular attention to the fact that X must have non-negative integer components. The rationale for this constraint results from Theorem 2.1, which gives a physical interpretation to S-invariants.

Theorem 2.1

Let M_0 be any initial marking, and M be any marking that is reachable from M_0 after a sequence of firings. X is an S-invariant if and only if

$$X^T * M = X^T * M_0. \qquad (2.4)$$

This relation is interpreted as a weighted conservation of tokens. A marking is by definition a vector of non-negative integers. Conservation of tokens must thus be expressed with non-negative integers.

Definition 2.10

If X is an S-invariant, the set of places whose corresponding components in X are strictly positive is the *support* of the invariant, noted <X>.
The support of an S-invariant is said to be *minimal* if and only if it does not contain the support of another S-invariant but itself and the empty set.

Theorem 2.2

If X^1 and X^2 are two S-invariants with the same non-empty minimal support, then X^1 and X^2 are linearly dependent.

Proof.

Consider $X^1 = [x^1{}_i]$ and $X^2 = [x^2{}_i]$, $i = 1,..,n$. By assumption, X^1 and X^2 are non null vectors. Nothing is changed if it is assumed that the support is made out of the first p, $0 < p \leq n$, places.
Define $r = \min_{i=1..k} (x^1{}_i / x^2{}_i)$ and m an integer large enough so that for every i $m*r*x^2{}_i$ is an integer.
Then $m * (X^1 - r * X^2)$ is an S-invariant whose support is strictly included in the support of X^1 and X^2.
Indeed, $C^T * m * (X^1 - r * X^2) = m * C^T * X^1 - m * r * C^T * X^2 = 0 - 0 = 0$. For every i, $m * x^1{}_i - m * r * x^2{}_i$ is an integer (Definition of m), and $x^1{}_i - r * x^2{}_i$ is non negative (Definition of r). Finally, by definition of r there exists some i_0 such that $r = x^1{}_{i_0} / x^2{}_{i_0}$, hence $m.(X^1 - rX^2)_{i_0} = 0$.
Consequently, the support of $m * (X1 - r * X2)$ is Ø. Thus $m * (X1 - r * X2)$ is zero. The vectors are linearly dependent.

Definition 2.11

A *minimal support S-invariant* X is an S-invariant whose support <X> is minimal.

The following important result, due to Memmi and Roucairol (1979), highlights the importance of minimal support S-invariants. Valraud (1989) presents an application of this result to analyze structural properties of a net.

Theorem 2.3

Consider a net P. The set of minimal supports of the net P is finite.

If <X>1, ..., <X>k are the k finite supports, andX1,..., Xk is a family of S-invariants, with <Xi> = <X>i, then the family X1,..., Xk constitutes a minimal generating family of the S-invariants, i.e., every S-invariant can be written as a linear combination of X1,..., Xk with rational coefficients.

Definition 2.12

The *S-component* associated with an S-invariant X of a Petri Net is the subnet whose places are the places of <X> and whose transitions are the input and output transitions of the places of <X>.

By extension, a *minimal S-component* is the S-component of a minimal support S-invariant.

Example 2.7: Consider the Petri Net PN of Figure 2.5.

The incidence matrix of PN is

$$C = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

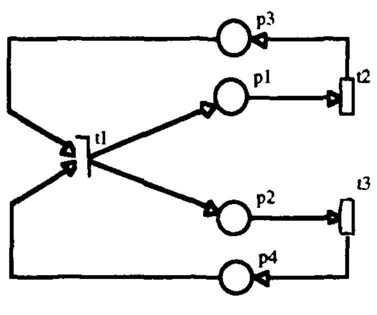


Figure 2.5 Petri Net PN

$X = [x1, x2, x3, x4]$ is an S-invariant if and only if $C^T * X = 0$.
This yields $x1 = x3$ and $x2 = x4$. There are two minimal supports <X1> = {p1, p3} and<X2> = {p2, p4}. The S-components associated with <X1> and <X2> are depicted in Figures 2.6 and 2.7.

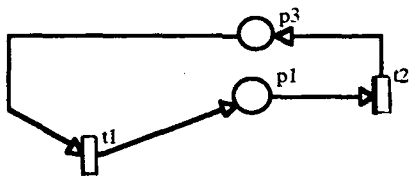


Figure 2.6 S-component associated with <X1>

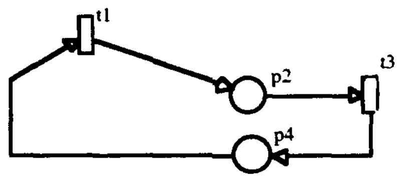


Figure 2.7 S-component associated with <X2>

2.2.5 Marked Graphs

Definition 2.13

A *marked graph* is a connected Petri Net in which each place has exactly one input and one output transition.

Throughout this thesis, marked graphs play an important role. One crucial result about marked graphs is Theorem 2.4 (Hillion, 1986). This result has been applied extensively in Remy (1986) to characterize the Petri Net model of fixed structure systems; it is used here in Chapter VIII.

Example 2.8: The net in Figure 2.1 is not a marked graph., because this net has two sources, i.e. two places without input arcs, and one sink, i.e., a place without an output arc. Figure 2.8 shows a marked graph.

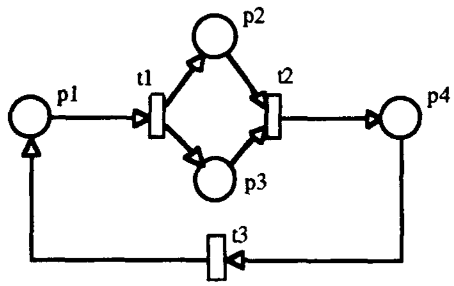


Figure 2.8 Marked Graph

Theorem 2.4 is stated after the introduction of two new terms.

Definition 2.14

A *directed circuit* is a directed path from one node back to itself. In Fig. 2.8 p1-t1-p3-t2-p4-t3-p1-t1-p2-t2-p4-t3-p1 is a directed circuit.
A *directed elementary circuit* is a directed circuit in which only one node appears more than once. In Fig. 2.8, p1-t1-p3-t2-p4-t3-p1 is a directed elementary circuit. The place p1 is the node that appears more than once.

Theorem 2.4 (Hillion, 1986)

The minimal S-components of a marked graph are exactly its directed elementary circuits.

Theorem 2.4 is important, because it indicates that the computation of the minimal S-components can be done by an efficient algorithm based on Linear Algebra, such as the algorithm of Alaiwan and Toudic (1985).

In this thesis, a particular type of nets are of importance. In these nets, all the places but two have exactly one input and one output transition. There is one place with only one output transition (the source or the external place) and one place with one and only one input transition (the sink). These nets can be transformed into marked graphs by merging the external place and the sink into a single place p_0, (Hillion, 1986). Under those circumstances, the simple information flow paths from the source to the sink are exactly the directed elementary circuits that contain the place p_0. The simple information flow paths can be computed in that case using the algorithm of Alaiwan and Toudic. See Valraud (1989) for an extensive treatment.

2.2.6 Petri Nets with Switches

A switch is a node with multiple output places. As with any transition, a switch is enabled whenever there is at least one token in each of its input places. When a switch fires, a token is put in *only one* of its output places. This place is chosen according to some decision rule.

The decision rules associated with the switch can be anything. They can be deterministic or stochastic. They can take the information that is contained in the inputs into account, etc. It is thus possible to model distributed variable structures with switches.

Example 2.9: Figure 2.9 represents a Petri Net with a switch.

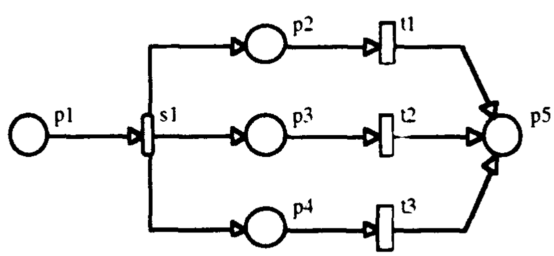


Figure. 2.9 Petri Net with a Switch

At the stage modeled by the switch s1 there are three alternative courses of action. According to some rule, only one is chosen. In each case, the course of action that is chosen will satisfy the condition modeled by p5.

2.3 HIERARCHICAL PETRI NETS

Hierarchical Petri Nets allow the designer to create a large model composed of many submodels, and isolate a segment to study its details without disturbing or altering the entire structure. They also provide a modular approach towards modeling a complex system. This feature is vital for designing complex systems that require frequent study of alternative structures during the development process. The hierarchical nature of the Petri Nets provides the designer an abstraction mechanism that

- provides an overview and an adequate representation of system structure, absent in single level system models;
- hides details in a consistent way;
- separates into well-defined and reusable components;
- supports top-down and bottom-up design strategies.

2.3.1 Compound Transition

If a subnet of a Petri Net model is replaced by a single transition, the single transition is termed *compound transition*. It represents the aggregated effect of the processes represented by the transitions of the subnet. The system with compound transitions describes the system at a higher degree of abstraction than the one without them.

Figure 2.10 shows a Petri Net model of a system in which the system's functionality is described at the most detailed level. The dotted box contains the processes that are to be aggregated. In Figure 2.11 the outlined subnet is shown replaced by a single transition - a compound transition denoted by the label "HS". The subnet that represents the compound transition at a *subpage* is shown in Figure 2.12. The term subpage is used in *Design /CPN™*, a commercially available software package for Hierarchical Petri Nets, to denote pages which contain the subnets replaced by compound transitions and compound places.

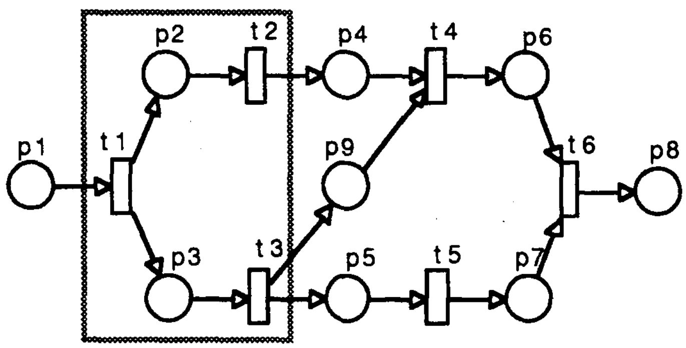


Figure 2.10 Detailed Description of a System

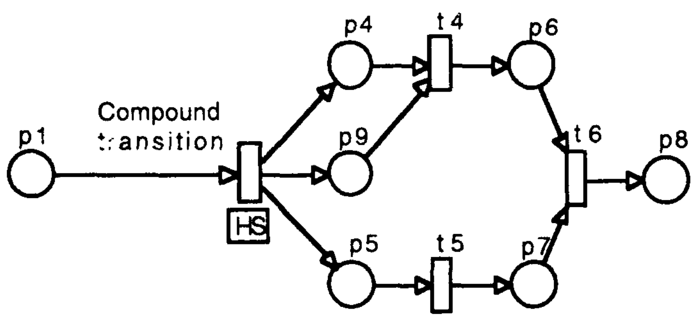


Figure 2.11 System's Description with a Compound Transition

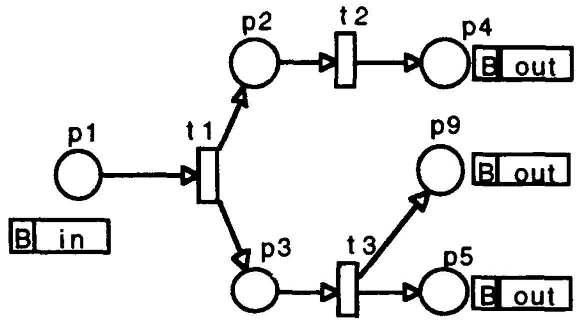


Figure 2.12 Subpage Representation of the Compound Transition

The places, in Figure 2.12, with label "B in" or "B out" represent the *port nodes*. Port nodes are defined to be the input and output places of the subnet; its connections with the uncompound net. On the other hand, all those places whose input and output transitions are defined within the subnet are not port nodes. Port nodes are used to preserve the connectivity of the original net. They model the sockets for the places that exist in the preset and postset of the compound transition in the system's net. The places p1, p4, p5, and p9 in Figure 2.11 are defined as port nodes in Figure 2.12.

When it is desired to replace a subnet by its compound transition representation care must be taken in selecting the boundaries of the subnet. In order to replace a subnet of a net by a compound transition, the boundaries of the subnet should be comprised only of transitions.

The boundary of a subnet is defined to be the set of nodes belonging to the subnet having at least one of their input and/or output be nodes of the net that do not belong to the subnet. A subnet with at least one place at the boundary of the subnet can not be replaced by a compound transition. Figure 2.13 presents such a situation with place p9 as the part of the subnet that is desired to be replaced by a compound transition.

2.3.2 Compound Place

On the other hand, if a subnet of a Petri Net model is replaced by a single place, the single place is termed *compound place*. It represents the aggregated effect of the subnet replaced by the compound place. The system with compound places describes the system at a higher degree of abstraction than the one without them.

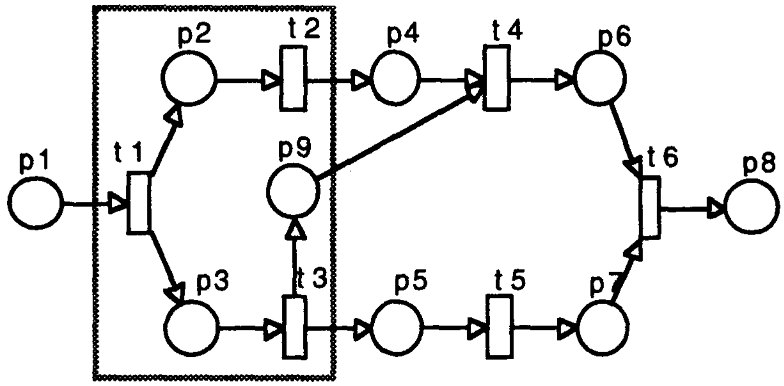


Figure 2.13 Illegal Compounding

Figure 2.14 shows the Petri Net model of a system in Figure 2.10 with the dotted box representing the subnet that is desired to be aggregated by a compound place. In Figure 2.15 the outlined subnet is shown replaced by a single place - a compound place. The subnet that represents the compound place at a subpage is shown in Figure 2.16.

The transitions, in Figure 2.16, with labels "B in" or "B out" represent the port nodes. Port nodes model the sockets for the transitions that exist in the preset and postset of the compound places in the system net. The transitions t1, t4, and t5 in Figure 2.15 are defined as port nodes in Figure 2.16.

When it is desired to replace a subnet by its compound place representation, care must be taken in selecting the boundaries of the subnet. In order to replace a subnet of a net by a compound place, the boundaries of the subnet should be comprised only of places.

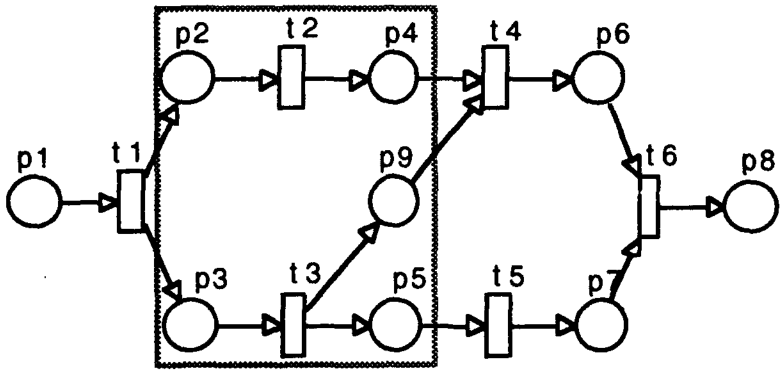


Figure 2.14 System in Figure 2.10

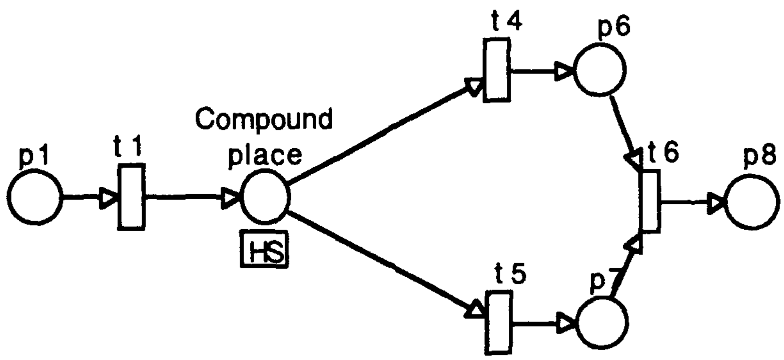


Figure 2.15 System's Description with a Compound Place

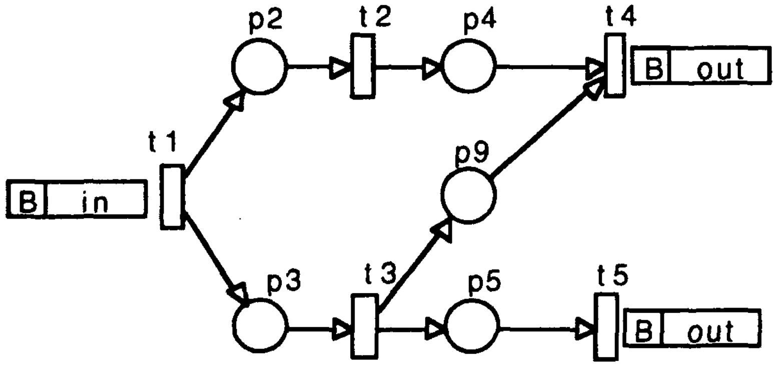


Figure 2.16 Subpage Representation of the Compound Place

The subnet outlined in Figure 2.17 can not be replaced by a compound place as it has a transition t1 at its boundary.

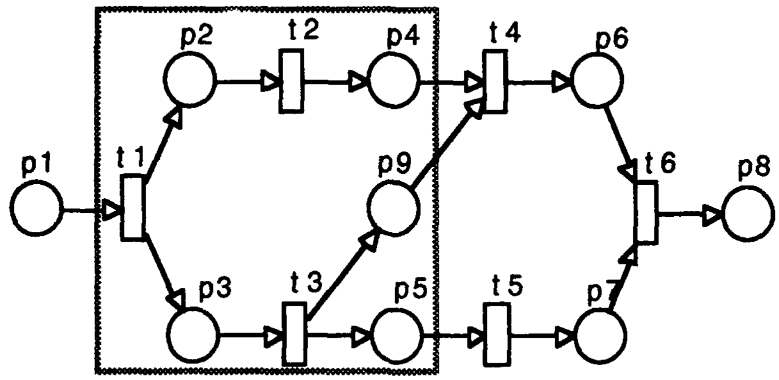


Figure 2.17 Illegal Compounding

2.3.3 Folding and Unfolding a Net

A Petri Net model of a system is said to be folded, if certain subnets of the net are aggregated by compound transitions and/or compound places. The folded net obtained as a result describes the system at a higher degree of abstraction. The subnets replaced by compound transition and/or compound places are moved to the subpages as a result of folding the net. The original detailed description of the system net can be retrieved by uncompounding the compound transitions and compound places, i.e., by moving the subnets back to their original locations. A compound transition or a compound place, therefore, represents a subnet stored at a subpage with port nodes to preserve the original connectivity of the net. The process of uncompounding all the compound transitions and compound places is termed unfolding the net. In this thesis, the organizational structures represented in terms of Petri Nets are folded by creating compound transitions representing different suborganizations. *The processes of folding and unfolding do not effect the Petri Net properties of the structures; the structures obtained as a result of folding and unfolding are legitimate, executable, Petri Nets.* The folded structures can be executed with or without the subpage structures. Figure 2.18 presents a Petri Net with two of its subnets outlined by dotted boxes. The outlined subnets are replaced by their compound transition representation in Figure 2.19. The Petri Net in Figure 2.19 is the folded version of the net in Figure 2.18. It represents the same system in Figure 2.18 but at a higher degree of abstraction.

The subnets that are moved to subpages as a result of folding are shown in Figures 2.20 and 2.21. Figure 2.20 represents the net replaced by compound transition **t1** along with the port nodes, while the subnet replaced by the compound transition **t2** is shown in Figure 2.21.

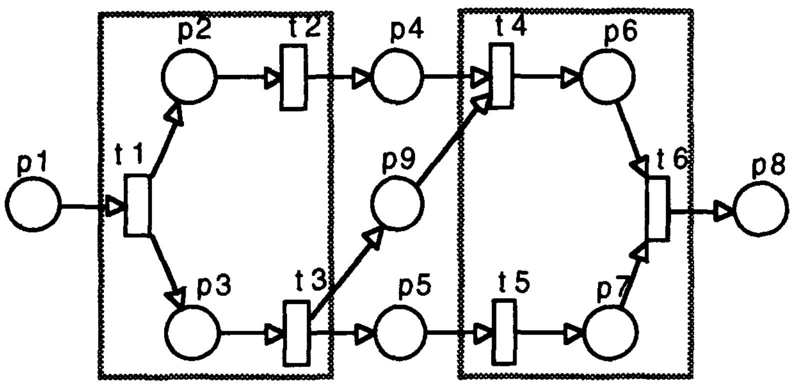


Figure 2.18 Petri Net of a System

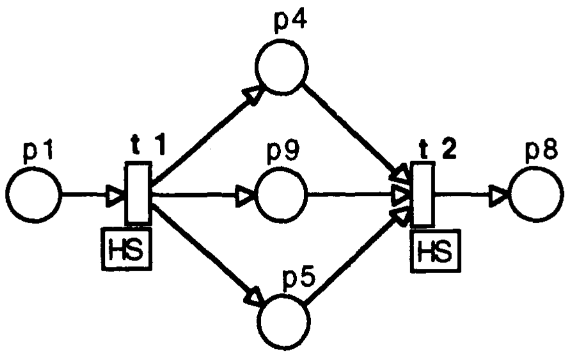


Figure 2.19 Folded Petri Net

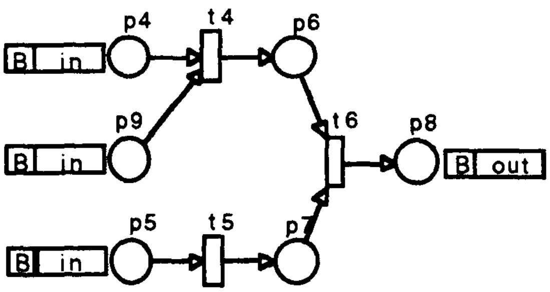


Figure 2.20 Subnet Replaced by Compound Transition **t1**

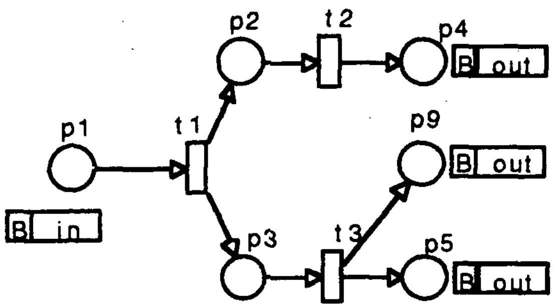


Figure 2.21 Subnet Replaced by Compound Transition **t2**

The places p4, p5, and p9 in Figure 2.19 are all the output places of the compound transition **t1** and input places of compound transition **t2**. If the system's behavior at a higher degree of abstraction is desired to be depicted, the three places p4, p5, and p9 can also be represented by an equivalent single place **p2** with input and output arcs having a weight of 3 as shown in Figure 2.22. If the single equivalent place **p2** models the flow of information from the aggregated processes represented by **t1** to aggregated processes represented by **t2** and the three places between **t1** and **t2** in Figure 2.19 represent a redundancy in the flow of information as the tokens are defined to be indistinguishable then Figure 2.23 may be used where there is no weighting on the input and output arcs of **p2**.

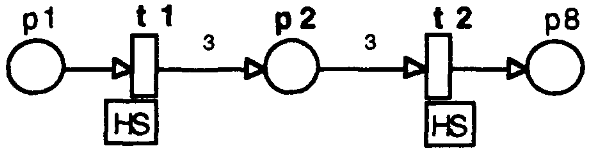


Figure 2.22 Folded Version of the Net in Figure 2.18

The net in Figure 2.23 can be unfolded to the net in Figure 2.18 by uncompounding the compound transitions **t1** and **t2**. The places that are represented by the equivalent place are defined in the subnets in Figures 2.20 and 2.21, therefore, whenever the compound transitions are uncompounded, all the places present in the original net will be retrieved from the subpages producing the original detailed description of the net in Figure 2.18.

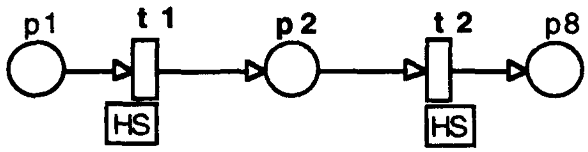


Figure 2.23 Folded Version of the Net in Figure 2.18

The folding process presented in this section will be used in the design methodology presented in this thesis.

The process of folding Petri Nets also refers to a technique used to translate the Ordinary Petri Nets to their Colored Petri Net representations. Since Colored Petri Nets are not used in this thesis, the folding process mentioned is not discussed here. Interested readers are referred to Jensen (1990).

2.4 CONCLUSION

Petri Nets were introduced to represent or model complex systems. A number of reasons were outlined that make Petri Net modeling appealing as compared to other modeling languages. A mathematical framework for Ordinary Petri Nets was presented. An extension of Petri Nets, Hierarchical Petri Nets, was presented as the mathematical and modeling framework used throughout the thesis.

CHAPTER III

ORDERING AND LATTICES

The basic concepts about orderings and lattices are presented in this chapter. Lattice theory is used extensively in Chapters IV, VI, and VII to address the generation of Stratified Decision Making Organizational (SDMO) structures. Complementary material on lattices can be found in Birkhoff (1948) and Grätzer (1971). Relationships between lattices and graphs are explained in Carré (1979). The development in this chapter follows (Remy, 1986).

3.1 DEFINITIONS

Definition 3.1

A relation R on a set A is called a binary relation if and only if
$\forall (x,y) \in A^2$ the condition x R y either does or does not hold.
In other words, for each (x, y) "x R y" is meaningful, being either true or false.

Example 3.1

Let A be the set of graduate students at the ECE department of GMU, and R be the relation "has fewer than or equal number of semester hours as". Then R is a binary relation.

Definition 3.2

A relation R on a set A is an ordering, if and only if

- R is reflexive: $\forall x \in A \quad x R x$.
- R is antisymmetric: $\forall (x, y) \in A^2 \quad (x R y) \text{ and } (y R x) \Rightarrow (x = y)$.

• R is transitive: $\forall$ (x, y, z) $\in A^3$ (x R y) and (y R z) $\Rightarrow$ (x R z).

Example 3.2

- In {0, 1} the relation "is smaller than or equal to", denoted by $\leq$, is an ordering.

- Let S be the set of vectors with three entries in {0,1}: X = [x1, x2, x3] x1, x2, x3 in {0,1}. Define on S the relation « :

  For X = [x1, x2, x3] and Y = [y1, y2, y3],
  X « Y if and only if x1 $\leq$ y1 x2 $\leq$ y2 x3 $\leq$ y3 .

It is easy to conclude that « is an ordering of S.

Definition 3.3

An ordering R of a set A is a total ordering or chain if and only if given any (x, y) $\in A^2$, either x R y or y R x.
If an ordering is not a total ordering, it is called a partial ordering.

Example 3.3

- The set of real numbers is totally ordered by the binary relation "is smaller than" (<).

- The set S of example 3.2 is not totally ordered by «.
  Neither [1, 0, 0] « [0, 1, 0] nor [0, 1, 0] « [1, 0, 0] are true.

Definition 3.4

By *y covers x* is meant that x R y and that there is no element z, z $\neq$ x, y such that x R z R y.

Definition 3.5: *Connected Chain*

A chain $x_0 < x_1 < ... < x_i < ...$ will be connected if x_i covers x_{i-1} for all i.

Definition 3.6: *Dimension*

The dimension d[x] of an element x of a partially ordered set X is the maximum length d of connected chains $x_0 < x_1 < ... < x_d = x$ in X having x for greatest element - in case d is finite. Similarly, by dimension d[X] of X is meant the maximum length of a chain in X.

The notion of dimension is of particular importance when the following condition is satisfied:

Jordan-Dedekind Chain Condition

All finite connected chains between fixed end points have the same length.

Theorem 3.1

Any subset of a partially ordered set is itself partially ordered by the same binary relation (Remy, 1986).

3.2 ORDERING

An ordered set can be depicted very conveniently by a diagram, called the Hasse diagram. In this diagram, each element is represented by a point, so placed that if x R y is true then the point representing x lies below the point representing y. Lines are drawn between two points x and y if and only if y covers x. Figure 3.1 shows the Hasse diagram of the set S described in Example 3.2.

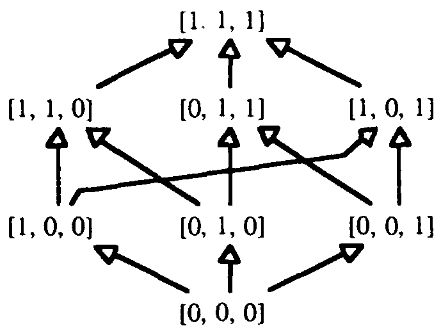


Figure 3.1 Hasse Diagram

In an ordered set, totally or partially ordered, some elements have properties that are of interest. In the next paragraphs, some of these elements are defined.

Definition 3.7

Let R be an ordering of A.

- If A contains an element ω such that ω R x for all x in A, then ω is unique and is called the *least* element of A.

- If A contains an element Ω such that x R Ω for all x in A, then Ω is unique and is called the *greatest* element of A.

Remark: These elements do not always exist. For example, in S, these elements exist. The least element is [0, 0, 0], and the greatest element is [1, 1, 1]. However, if '«' is restricted to the subset s, where s = S - {[0, 0, 0], [1, 1, 1]}, it is impossible to find a greatest and a least element.

Definition 3.8

- An element m of A is a *minimal* element if there does not exist any element in A that is strictly inferior to m, i.e.,

$$x \text{ R } m \text{ implies } x = m.$$

- An element M of A is a *maximal* element if there does not exist any element in A that is strictly superior to M, i.e.,

$$M \ R \ x \text{ implies } x = M.$$

Theorem 3.2 (Birkhoff, 1948)

Every finite ordered set A has at least one minimal and one maximal element.

Proof: Let the elements of A be x1,..., xn. Define the finite sequences m_k and M_k by:

$m_1 = M_1 = x_1$
$m_k = x_k$ if $x_k \leq m_{k-1}$, otherwise $m_k = m_{k-1}$.
$M_k = x_k$ if $M_{k-1} \leq x_k$, otherwise $M_k = M_{k-1}$.

Then m_n is by construction a minimal element of A, and M_n is by construction a maximal element.

Example 3.4: In s = S - {[0, 0, 0], [1, 1, 1]} we have three minimal elements, and three maximal elements.

The minimal elements are [1, 0, 0], [0, 1, 0], [0, 0, 1].
The maximal elements are [1, 1, 0], [1, 0, 1], [0, 1, 1].

Theorem 3.3

In a totally ordered set or chain, the notions minimal and least (respectively maximal and greatest) are equivalent.

Proof: If a is minimal then no other element of the chain X is inferior to it, i.e., x R a is not true. By Definition 3.3 we have then that a R x must hold for all $x \neq a$, a is therefore the least element of X. A similar reasoning applies to the maximal element.

The following theorem, Birkhoff (1948), gives a characterization of the Jordan-Dedekind condition.

Theorem 3.4 (Birkhoff, 1948)

Let X be a partially ordered set which has a least element (ω) and a greatest element (Ω) and in which all chains are finite. Then X satisfies the Jordan-Dedekind chain condition if and only if there exists an integer-valued function f[x] such that

$$y \text{ covers } x \Leftrightarrow y > x \text{ and } f[y] = f[x] + 1$$

3.3 LATTICES

If the set is totally ordered, its structure is particularly simple; it is a single chain. In most cases however, the ordering is not total. In order to gain some insight into the structure of the set, the concept of a lattice (Birkhoff, 1948) is needed, which is based on local properties of the set.

Definition 3.9

Let B be a subset of a partially ordered set A.

- An upper bound of B is an element a of A such that y R a for all y in B. The least upper bound (l.u.b) of B, if it exists, is the least element of the set of all upper bounds of B.

- By analogy, the greatest lower bound (g.l.b) is the greatest element, if it exists, of the set of all lower bounds of B.

Example 3.5: Let B be {[0, 0, 0], [1, 0, 0], [0, 1, 0]}. B is a subset of S.

It has a g.l.b, which happens to belong to B: [0, 0, 0].
It has a l.u.b, which does not belong to B: [1, 1, 0].

Definition 3.10

A lattice is a partially ordered set L in which any two elements x, y have
• a g.l.b or meet (denoted by $x \cap y$) that belongs to L.
• a l.u.b or join (denoted by $x \cup y$) that belongs to L.

Definition 3.11: *Sublattice*

A sublattice L' of a lattice L is a subset L' of L such that the join and meet of any two elements of L' are in L'.

Figure 3.2 illustrates the local condition. Given two elements x and y, there exists only one element in the Hasse diagram, the join, which covers both x and y. Similarly, there is only one element, the meet, which is simultaneously covered by both x and y.

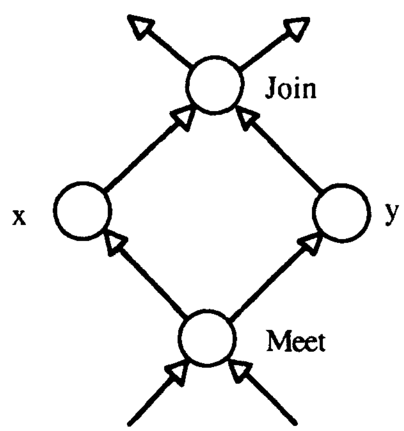


Figure 3.2 Local Condition

The lattice property ensures that the set L has some structuring patterns. It is possible to identify, for any two elements x and y, two unique boundaries in L, the join and the meet. Every element that is below x and y must be below the join. Any element that is above x and y must be above the meet.

Remark: Every Lattice L has a least and a greatest element. The least element is the meet, which belongs to L, of all the elements of L. The greatest element is the join of all the elements of L. This join belongs to L, by definition of the lattice.

Example 3.6:

- S is a lattice.

  On {0, 1} the meet ($\cap$) and join ($\cup$)operators are defined as:

  $0 \cap 0 = 0 \qquad 0 \cup 0 = 0$

  $0 \cap 1 = 0 \qquad 0 \cup 1 = 1$

  $1 \cap 0 = 0 \qquad 1 \cup 0 = 1$

  $1 \cap 1 = 1 \qquad 1 \cup 1 = 1$

  The operators are extended on a component-wise basis:

  $X \cap Y = [x1 \cap y1,\ x2 \cap y2,\ x3 \cap y3]$

  $X \cup Y = [x1 \cup y1,\ x2 \cup y2,\ x3 \cup y3]$

- Let X be a set, and P(X) be the set of all subsets of X. (P(X) is a lattice with the partial ordering "is included in".

  The meet of two subsets of X, A and B, is the intersection $A \cap B$.

  The join of two subsets of X, A and B, is the union $A \cup B$.

3.4 CONCLUSION

In this chapter some basic concepts of orderings and lattices were presented. A number of important properties of the elements of ordered sets were also presented. The elements of ordered sets bearing those properties were identified, i.e., least, greatest, minimal, and maximal elements. Similarly, some properties of partially ordered sets were outlined, namely meet and join.

It is concluded that a partially ordered set is a lattice if it has a greatest and a least element and in which any two elements have both a meet and a join in the set.

CHAPTER IV

MULTILEVEL ORGANIZATIONAL CLASSES

An introduction to the methodology for generating multilevel organizational structures is presented in this chapter. Section 4.1 provides basic concepts of multilevel organizational structures, while Section 4.2 presents the model of a single decision maker. The set of allowable interactions among decision makers is presented in Section 4.3. The modeling procedure for a compound node is described in Section 4.4. All the allowable interactions among compound nodes are discused in Section 4.5. A mathematical model describing the set of interactions is derived in Section 4.6.

4.1 BASIC CONCEPTS

A *Decision Making Organization (DMO)* is seen as an information processing system that performs several functions to accomplish its missions (Minsky, 1986, Levis, 1988). The functions are divided into individual tasks and subtasks, which are performed by *Decision Making Units (DMU)*. A DMU may be capable to perform several tasks or subtasks; the combination of a DMU performing a particular task is called a *role* (Demaël and Levis, 1990).

A DMO has a *variable structure* if the interactions among individual DMUs can vary, i.e., if the roles change according to the task the organization has to perform (Demaël, 1989). Conversely, a system for which the interactions can not vary has a *fixed structure*. This thesis is restricted to *fixed structure multilevel, hierarchical* systems.

4.1.1 Multilevel, Hierarchical System

The concept of a multilevel, hierarchical system is defined in Mesarovic et al. (1970). Some of the characteristics which every hierarchy has are: *vertical arrangement of subsystems which comprise the overall system, priority of action or right of intervention of*

the higher level subsystems, and dependence of the higher level subsystems upon actual performance of the lower level.

Mesarovic et al. (1970) defined three types of hierarchical systems. This classification is based on three notions of levels:

- The level of description or abstraction, the *stratum*.
- The level of decision complexity, the *layer*.
- The organizational level, the *echelon*.

The term *level* is reserved as a generic term referring to any of these notions when there is no need to distinguish between them.

The concept of stratum is used for modeling organizational architectures when viewed from different levels of abstraction, while the concept of layer is introduced in reference to the vertical decomposition of a decision problem into sub-problems. The concept of echelon refers to the mutual relationship between DMUs comprising a system.

It is necessary to make a clear distinction as to which notion of level one is using when describing a hierarchical system. The type of multilevel, hierarchical systems modeled in this thesis are *stratified systems*, where the system is described by a family of structures each concerned with the behavior of the system as viewed from a different level of abstraction, the stratum. A set of rules for defining echelons among subsystems (DMU) of a Stratified Decision Making Organization (SDMO) is also presented in Chapter VIII. The definition of echelons, within a DMO, is necessary to resolve the issues of interactions and connectivity among DMUs at a given stratum.

4.1.2 Stratified Decision Making Organization Classes

Definition 4.1

A *Stratified Decision Making Organization (SDMO)* is defined to be a DMO in which a system on a given stratum is a subsystem on the next higher stratum. In a SDMO, DMUs can be either Decision Making Sub-Organizations (DMSO) or human Decision Makers (DM) depending upon the level of abstraction used to represent the organizational structure of the DMO.

Example 4.1

> The administrative structure of GMU is a SDMO, as it can be viewed from a number of different levels of abstraction, i.e., faculty/staff, departments, Graduate School/Law School, etc.

For illustrative purposes, a description of a general SDMO is presented in Figure 4.1.

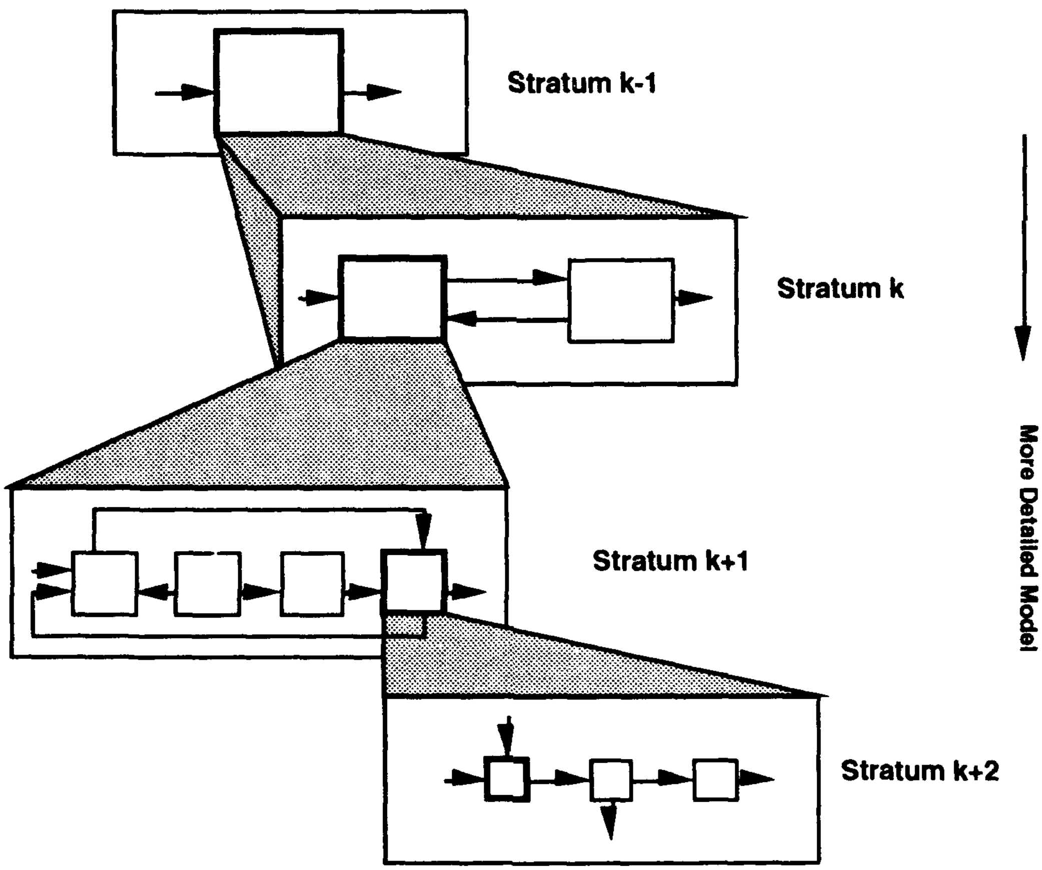


Figure 4.1 A Stratified Decision Making Organization

In a SDMO, the highest stratum, stratum '0', contains only one organizational structure, the *node*, which represents the entire organization (SDMO). The nodes at all other strata are referred to as DMUs. The node at stratum '0' shows the highest level of abstraction that can be used to describe an organizational structure. On the other hand, the n^{th} stratum contains an elaborated and detailed description of the DMO at the lowest level of abstraction that is determined by the designer of the organization. The range of 'n' is defined as $1 \leq n \leq N$, where 'N' represents the lowest possible stratum at which the DMUs can not be decomposed further. The determination of the value of 'N' is application dependent, i.e., it depends upon the kind of organization being modeled, and on the definition of strata used to describe the organization. For example, in human organizations, 'N' represents the stratum at which the DMUs are human decision makers (DM)

A DMU at stratum 'k' , where $1 \leq k < n$, is defined as a *compound node.* All nodes are labeled by an alphanumeric code, $\mathbf{DMU_{ik}}$, where 'i' represents the node number at stratum 'k'. The set of all the nodes at stratum 'k' contains $|\mu_k|$ elements, i.e., $\mu_k = \{1, 2, ..., |\mu_k|\}$ and $i \in \mu_k$.

Proposition 4.1

The following property holds for every stratified decision making organization (SDMO): The number of nodes at a stratum is larger than or equal to the number of nodes in the stratum immediately above it.

$$|\mu_n| \geq |\mu_{n-1}| \geq ... \geq |\mu_{k+1}| \geq |\mu_k| \geq |\mu_{k-1}| \geq ... \geq |\mu_0| = 1 \qquad 1 \leq n \leq N$$

Proof

Proposition 4.1 follows from the fact that a system on a stratum is comprised of a number of subsystems which are defined for the next lower stratum; the number of nodes at a given stratum is given by the sum of the subsystems of the individual nodes at the next higher stratum.

4.2 SINGLE INTERACTING DECISION MAKER

A number of models of a role have already been proposed. The origins of the model can be traced back to the four stage model of the interacting decision maker with bounded rationality introduced by Boettcher and Levis (1982). The formal specification of the interacting decision maker was made by Remy (1986). This specification led to an algorithm, the Lattice algorithm, which generates all feasible fixed structure architectures that meet a number of structural and user cons aints. Andreadakis (1988) introduced an alternative model, which was very similar to the four stage one in terms of the allowable interactions. Levis (to appear in 1992) presented a five stage model of a role that subsumes the four stage model and Andreadakis' extension. The proposed five stage interacting decision maker model is shown in Figure 4.2.

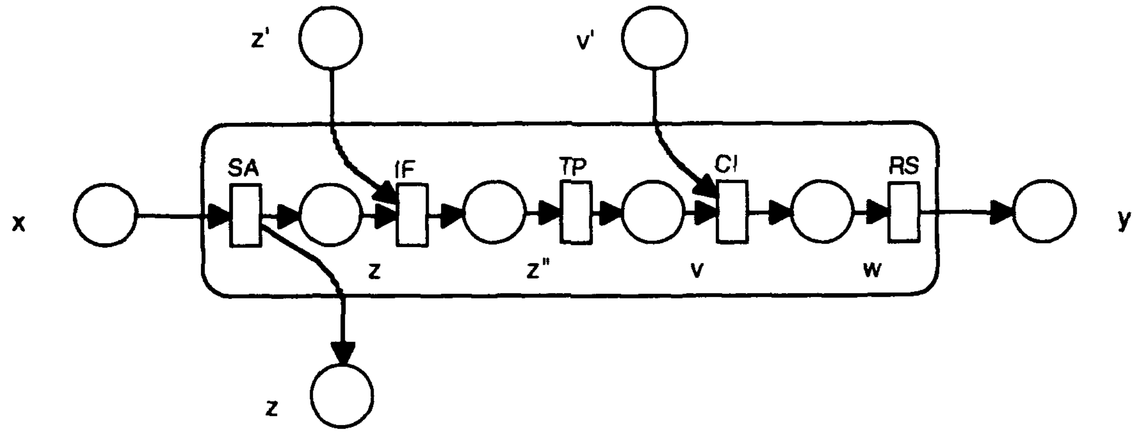


Figure 4.2 Single Decision Maker or Role

The decision maker receives a signal x from the external environment or from another organization member. The situation assessment (SA) stage contains algorithms that process the incoming signal to obtain the assessed situation z. The assessed situation z may be shared with other members. Concurrently, the decision maker can receive a signal z' from another part of the organization; z' and z are then merged together in the information fusion (IF) stage to produce z". The fused information is then processed at the task processing (TP) stage to produce v. The commands from other organization members are received as v'. The command interpretation (CI) stage then combines v and v' to produce the variable w, the input to the response selection (RS) stage. The RS stage contains algorithms that produce the output y.

This model explicitly shows all the stages at which a decision maker can interact with other decision makers or with the environment. A decision maker can receive input from the external environment only at the SA stage. The other inputs, z' and v', can be multiple and originate from different organizational members. Conversely, a decision maker can send output to the external environment only from the RS stage. The output z can only be sent to other organizational members. The output y can be sent to the external environment. The output y can also be sent to other organization members as input x or z' or as command input v'.

A decision maker need not have all five stages while performing a specified task in an organization. Depending upon the inputs and outputs a decision maker can have one of the four possible internal structures.

- SA alone with y = z
- SA, IF, TP, CI and RS
- IF, TP, CI and RS with x = z'
- CI and RS with x = v'

Note that the five stage model presented in this section describes the model of a single node at stratum 'N', the lowest possible stratum.

4.3 INTERACTIONS AMONG DECISION MAKERS

The set of all allowable interactions among the decision makers has been defined by Remy (1986). The following subsections present a review of all the allowable interactions that can exist among decision makers.

4.3.1 Allowable Interactions

The four possible links from a decision maker to another one are shown in Figure 4.3. Note that Figure 4.3 does not represent a feasible organizational structure as some of the allowable interactions shown can not exist simultaneously. A detailed discussion of feasible organizational structures is presented in Chapter VI. The following section describes the physical significance of the four kinds of interactions presented in Figure 4.3.

4.3.2 Physical Significance of the Interactions

• External input to SA of DM_i: e_i

This link represents the presence of an external input to a decision maker DM_i. The content of this information is not the topic of discussion here, and is taken as application-dependent, see Stabile and Levis (1981) and Hall (1982). The nature of the external input, however, is discussed in subsection 4.5.2.

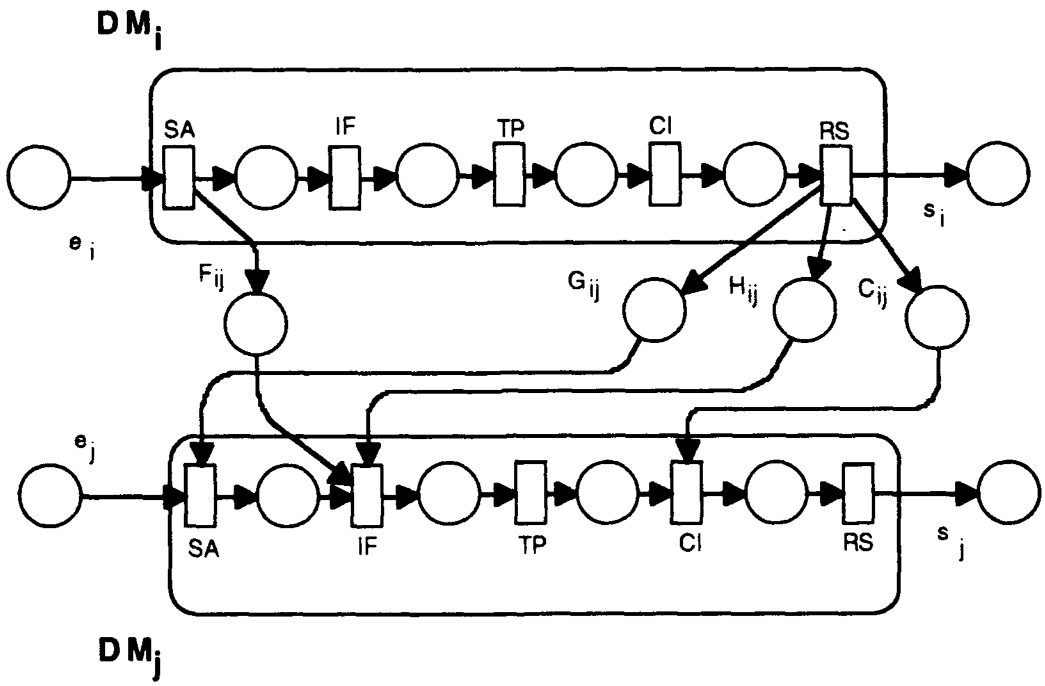


Figure 4.3 Allowable Interactions

• External output from RS of DM_i: s_i

This link represents the presence of an external output from a decision maker DM_i. Again the nature and content of this information are not discussed here.

• SA of DM_i to IF of DM_j: F_{ij}

This link models the transmission of assessed situation from DM_i to DM_j. The presence of this link only represents the fact that such an interaction occurs at this stage between the two decision makers.

• RS of DM_i to SA of DM_j: G_{ij}

This interaction represents the case where the output of decision maker DM_i is the input to another decision maker, DM_j, e.g., a serial or tandem arrangement. This interaction models the situation where in addition to the information about the task, DM_i sends a *control signal* to DM_j in order to trigger the task assigned to the latter.

• RS of DMi to IF of DMj: H_{ij}

This interaction models the result sharing type of information from decision maker DMi to decision maker DMj. The output information sent by decision maker DMi may or may not be taken into account by DMj in formulating his own response.

• RS of DMi to CI of DMj: C_{ij}

This link represents the issuing of a command from DMi to DMj. It introduces a multiechelon hierarchy between two decision makers.

4.4 SINGLE INTERACTING COMPOUND NODE

The second step of the methodology for generating stratified organizational structures is the definition of the compound nodes. As mentioned earlier, a DMU at stratum 'k', where $1 \leq k < n$ is a compound node. A compound node itself is a decision making sub-organization (DMSO) comprised of a number of DMUs defined at the next lower stratum. Therefore, *a compound node structure can be considered as a folded structure of the lower-strata DMUs and their interconnections*. The following subsection describes this process of folding by taking two DMUs, defined at stratum 'N'. As mentioned, the stratum 'N' is the lowest possible stratum that can be defined for a given Stratified Decision

Making Organization (SDMO). The DMUs in stratum 'N' are human Decision Makers (DM) and the rules formulated by Remy (1986) apply to the organizational structures in stratum 'N'. Once an organizational structure in stratum 'N' is folded into a compound node, then rules for interconnecting compound nodes need to be formulated. In case the interactional structure of a compound node follows the same rules as defined for a DM, then the folding of organizational structures with compound nodes will follow the same procedure described for organizational structures in stratum 'N'. As a result, the folding procedure can be generalized to fold an organizational structure in stratum 'k', where $1 \leq k \leq n$. This generalization follows from the fact that a DMU in an arbitrary straum 'k', where 'k' is not equal to 'N', is defined as a compound node. The reason for taking only two DMs to illustrate the folding procedure is that the interactions are defined in terms of a pair of DMs. The technique used for two DMs can be generalized to any number of DMs, taken two at a time.

4.4.1 Folding

Figure 4.4 shows an organizational structure with all allowable interactions from one decision maker to another with parts of the net grouped together. The grouped portions of the net are to be replaced by compound transitions. Figure 4.5 shows the structure of the organization in Figure 4.4 with all subnets compounded (replaced by compound transitions) and all places folded. A suffix 'C'- Compound - is, therefore, added to all five stages in the structure. The compounding procedure has been presented in Subsection 2.3.3. The folding of the places follows the discussion in Subsection 2.3.4. Figures 4.6, 4.7, 4.8, 4.9, and 4.10 present all the subnets that are replaced by the compound transitions. In these figures, the places with labels of the form 'B in/out' represent the input and output port nodes (Subsection 2.3.3) to the subnet replaced by a compound transition. The port nodes preserve the lower stratum connectivity among compound transitions, and hence play the rule of connectors when the structure is unfolded.

The new compounded and folded structure, Figure 4.5, preserves the connectivity of the structure in Figure 4.4. The ideas of compound transitions and folding of places are taken from Hierarchical Petri Nets and Colored Petri Nets respectively For a detailed description of compounding and folding a Petri Net model, see Peterson (1981) and Jensen (1987).

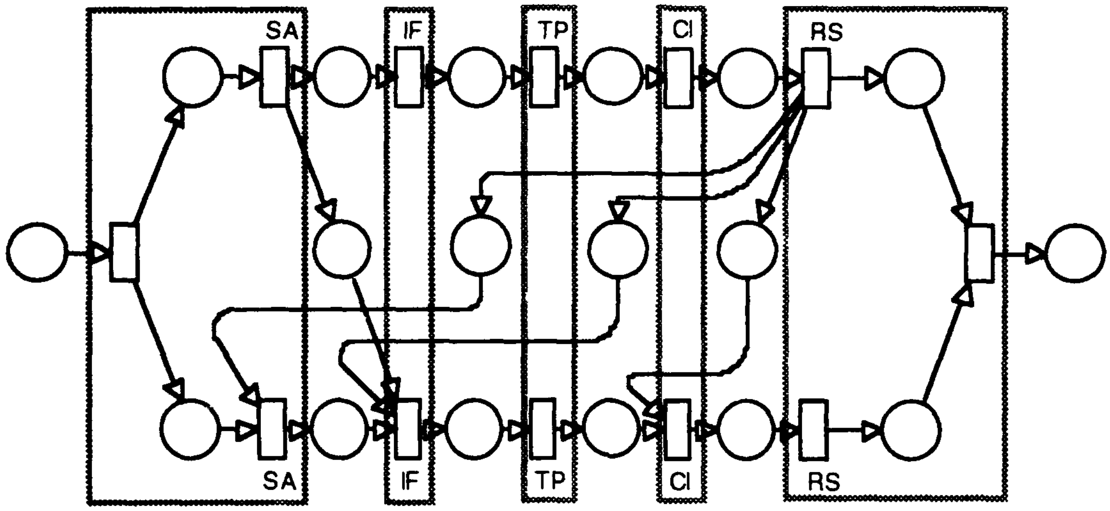


Figure 4.4 Organizational Structure in Stratum 'N' with Allowable Interactions

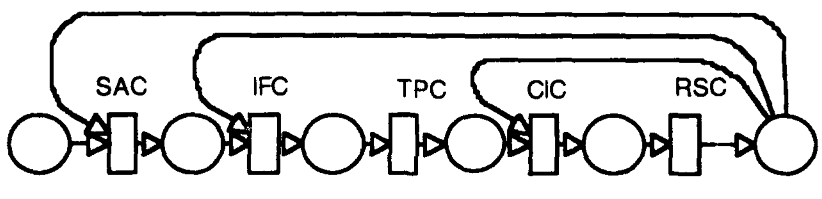


Figure 4.5 Folded Structure

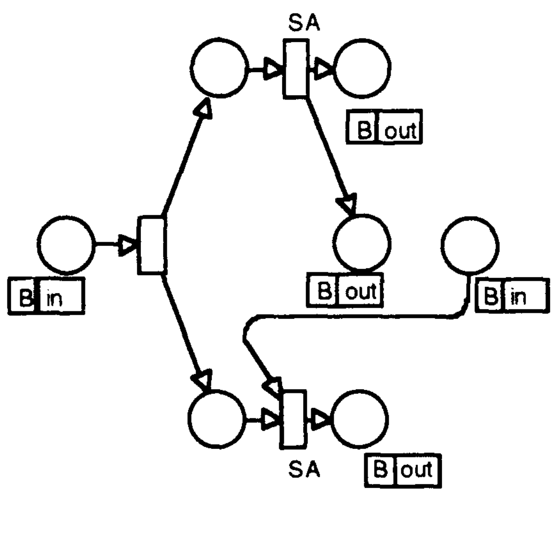


Figure 4.6 Subnet Replaced by SAC

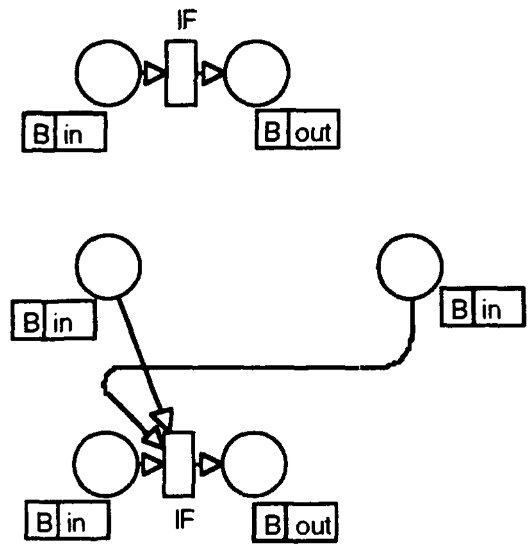


Figure 4.7 Subnet Replaced by IFC

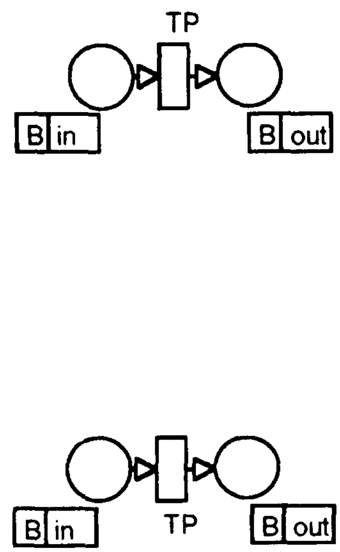


Figure 4.8 Subnet Replaced by TPC

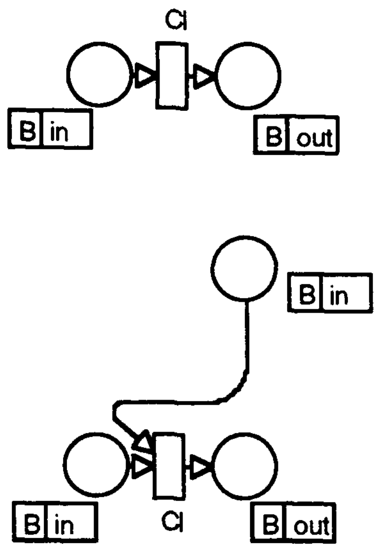


Figure 4.9 Subnet Replaced by CIC

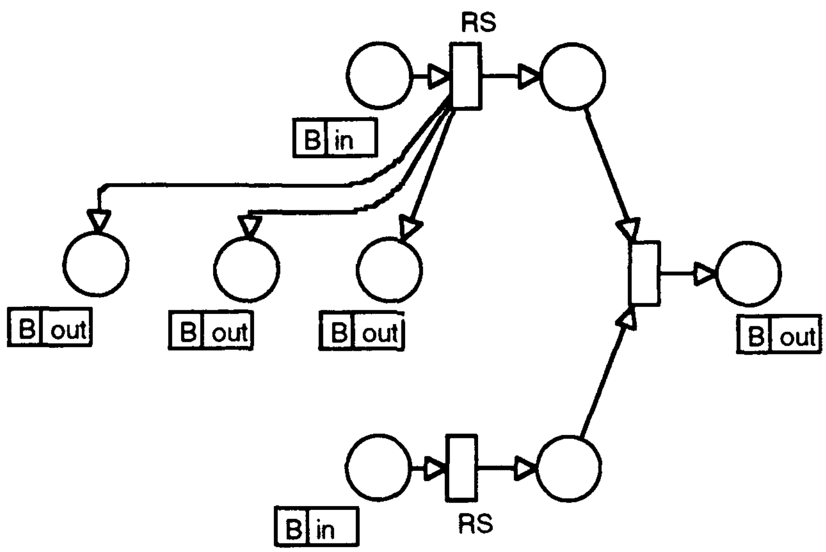
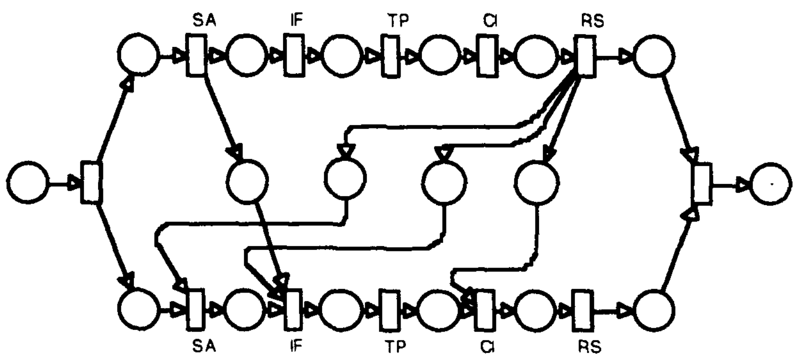


Figure 4.10 Subnet Replaced by RSC

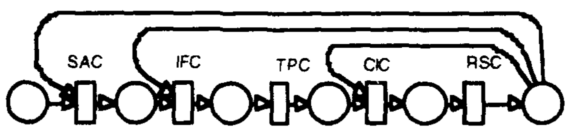
A careful investigation of different organizational structures with 'm' number of decision makers yields eight different folded configurations. Figures 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, and 4.18 show the organizational structures with all the possible interactions that can exist among DMs in stratum 'N' and their corresponding folded structures in stratum 'N-1'. Note that the set of folded configurations given in these figures is an exhaustive set; it contains all the possible configurations, as the elements of the set represent all the possible combinations in which DMs can interact with each other. Again, all possible configurations are illustrated for organizational structures with two DMs.

4.4.2 Compound Node

In all the configurations shown in Figures 4.11 to 4.18, the five stage processing part of the structures is identical. The feedback arcs which appear in some of the configurations represent the interactions among the DMs, where the DMs and their interactional structure is defined in the lower stratum (stratum 'N'). Therefore, the compound node is defined as the five stage structure shown in Figure 4.19. All the possible feedback arcs are suppressed in this model as the compound node structure represents the higher stratum description of an organizational structure. The lower stratum interactions need not appear in the higher stratum description as long as these interactions are preserved in the lower stratum description.

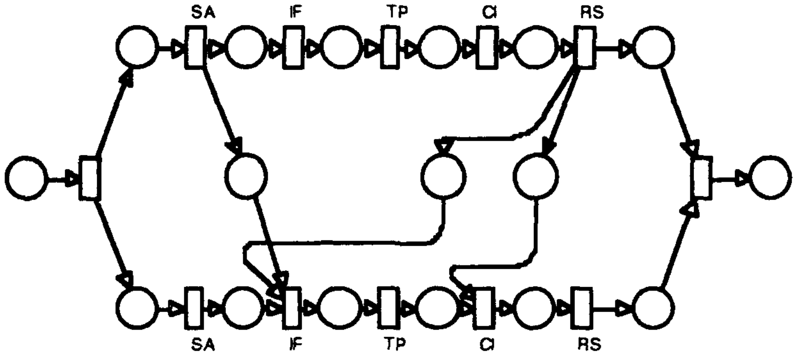


a) Organizational Structure in Stratum 'N'

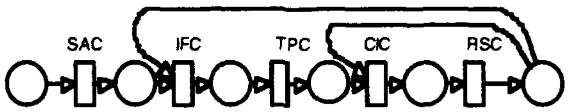


b) Folded Structure in Stratum 'N-1'

Figure 4.11 Organizational Structure 1 and its Compound Node Representation

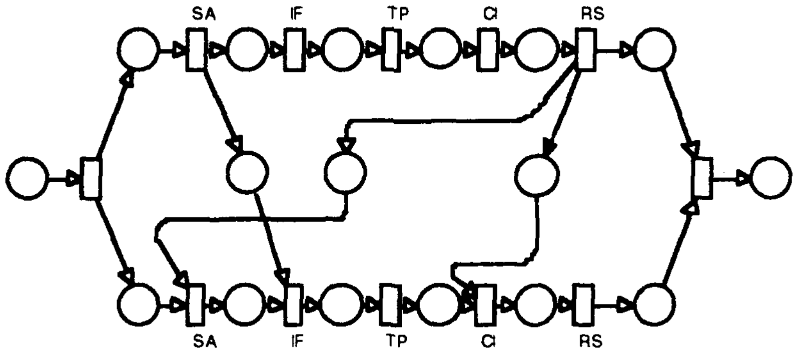


a) Organizational Structure in Stratum 'N'

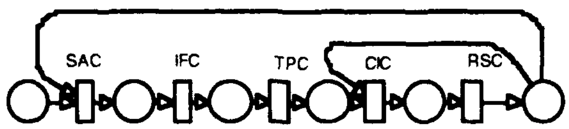


b) Folded Structure in Stratum 'N-1'

Figure 4.12 Organizational Structure 2 and its Compound Node Representation



a) Organizational Structure in Stratum 'N'



b) Folded Structure in Stratum 'N-1'

Figure 4.13 Organizational Structure 3 and its Compound Node Representation

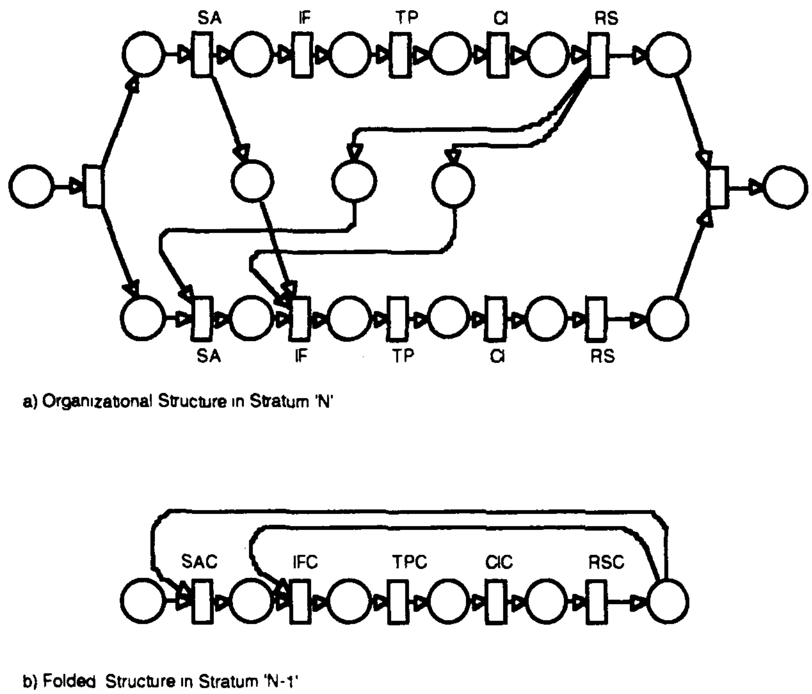


Figure 4.14 Organizational Structure 4 and its Compound Node Representation

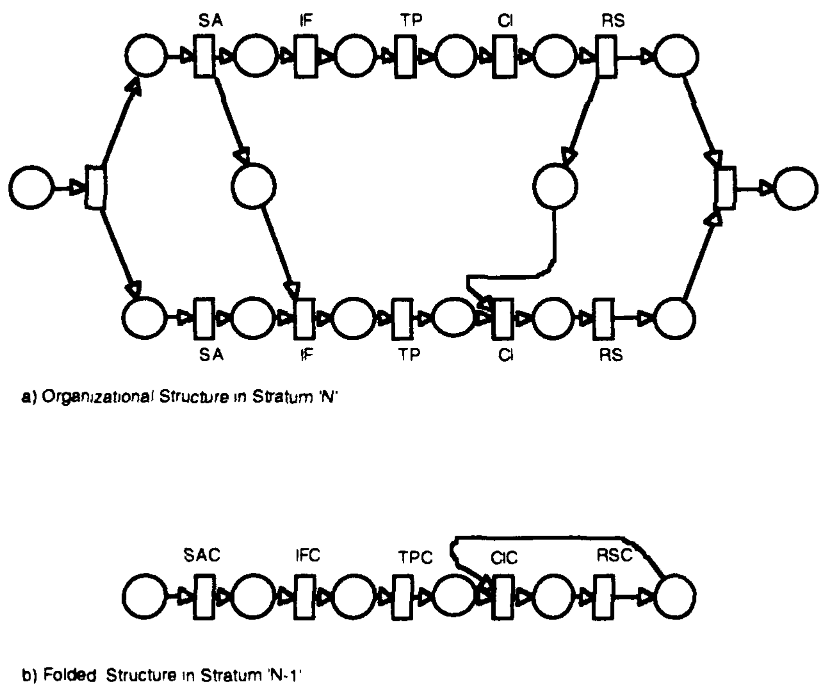


Figure 4.15 Organizational Structure 5 and its Compound Node Representation

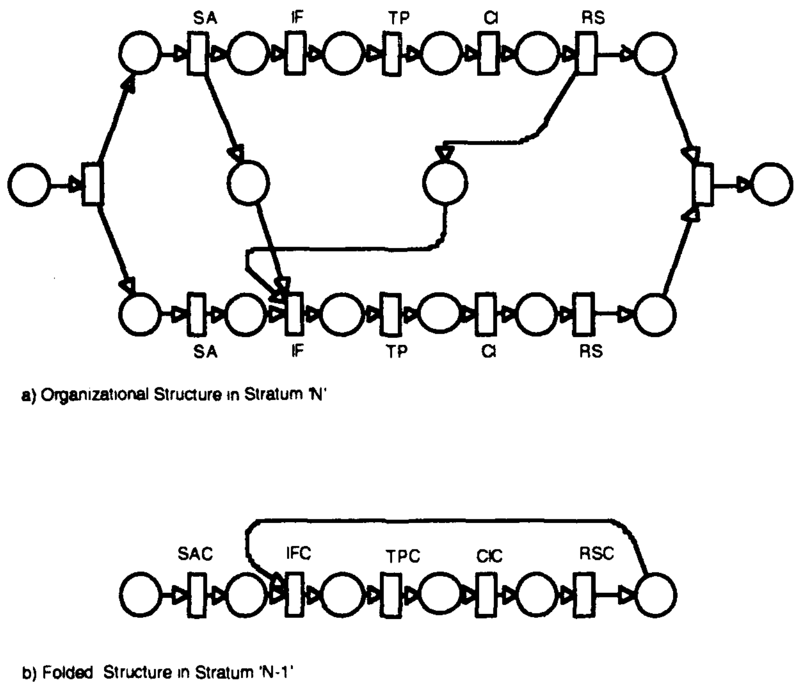


Figure 4.16 Organizational Structure 6 and its Compound Node Representation

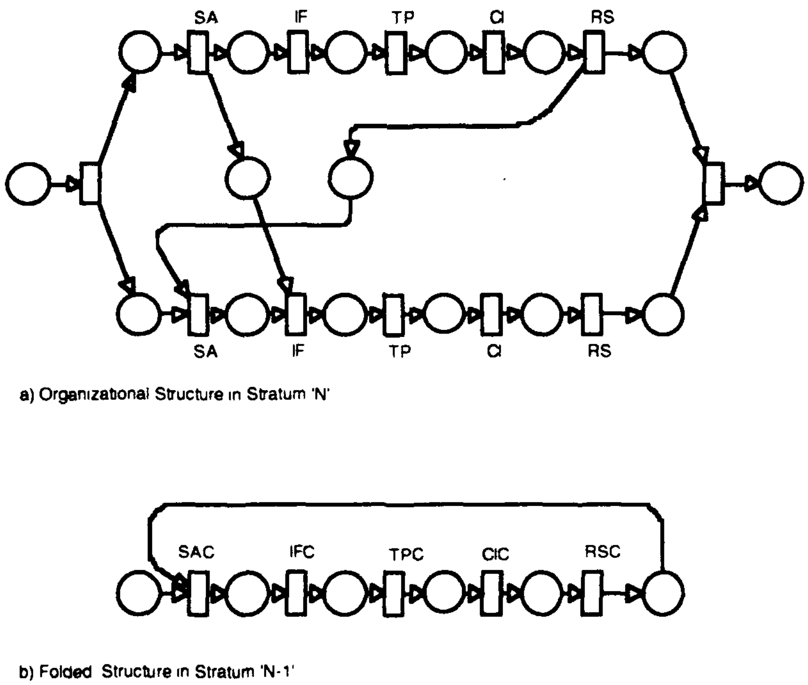
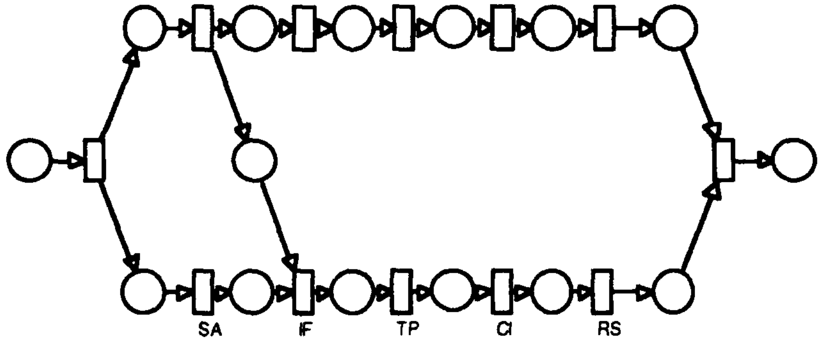


Figure 4.17 Organizational Structure 7 and its Compound Node Representation



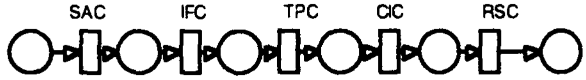


Figure 4.18 Organizational Structure 8 and its Compound Node Representation

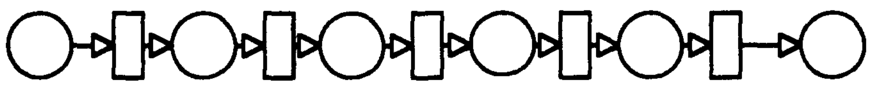


Figure 4.19 Compound Node

4.4.3 Single Interacting Compound Node

The five stage model of a DMU (compound node) presented in the previous section will be, from now on , the only one to be considered. The organizational structures, no matter what their interactional structure is, can be folded into the compound node structure of Figure 4.19. Note that the internal structure of the compound node will always contain all the five stages (SAC, IFC, TPC, CIC, and RSC). The presence of all the stages of a compound node introduces a number of structural constraints to be discussed in Chapter VI.

Figure 4.20 shows all the input and output stages of a single compound node. The input and output stages are the same as those of a DM described in section 4.2. The physical interpretation of these interactions, however, varies slightly from that of a single DM.

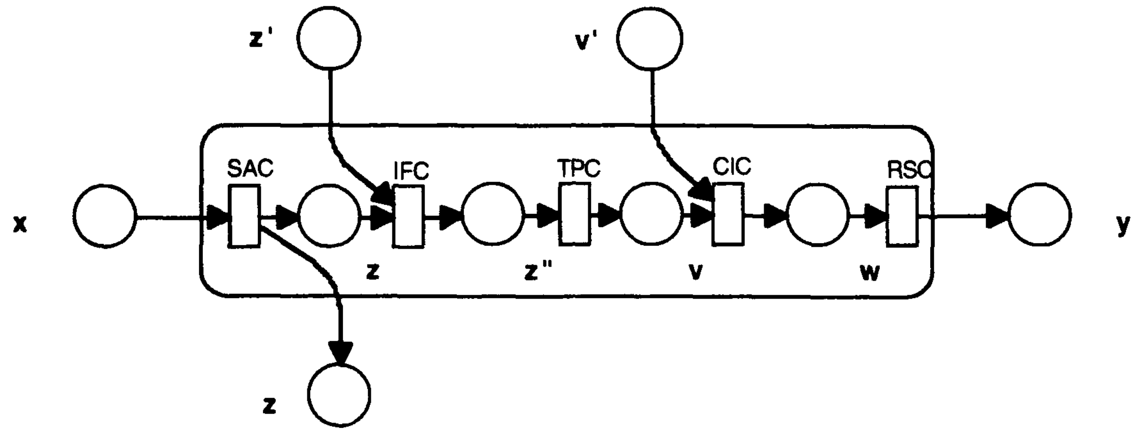


Figure 4.20 Single Interacting Compound Node

A compound node receives input or data **x** from the external environment (sensors) or from other compound nodes of a system. The incoming data are processed in the compound situation assessment (SAC) stage to get the assessed situation **z**. This variable may be sent to other compound nodes. If the compound node receives assessed data from other compound nodes, these data **z'** are fused together with its own assessment **z** in the compound information fusion (IFC) stage to get the revised assessed situation **z"**. The assessed situation is processed further in the compound task processing (TPC) stage to determine the strategy to be used to select a response. The variable **v** contains both the assessed situation and the strategy to be used in the compound response selection stage. A particular compound node may receive a command **v'** from superordinate compound nodes. This is depicted by the use of the compound command interpretation (CIC) stage. The output of that stage is the variable **w** which contains both the revised situation assessment data and the response selection strategy. Finally, the output or the response of the compound node, y, is generated by the compound response selection (RSC) stage.

The input and output stages of a compound node are the same as those of a DM; therefore, the organizational structure with compound nodes as DMUs will have the same kind of topology as of those with human decision makers. The folding procedure described for an organizational structure in stratum 'N' therefore can be generalized for any organizational structure in stratum 'k', where $1 \leq k \leq n$, if it is desired to have a stratum 'k-1' description of the organization.

4.5 INTERACTIONS AMONG COMPOUND NODES

The model of the compound node in Figure 4.20 is the one used to define the interactions that can exist between two compound nodes at a given stratum. This section describes all the allowable interactions that can exist among compound nodes.

The allowable links from a compound node to another are shown in Figure 4.21.

First, consider the inputs and the outputs to the compound nodes in Figure 4.21.

• External Input, Input to SAC of DMU_i: e_i

This link represents the *external input* to a decision making compound node. The external input is defined to be an item of information directly from the environment or a control signal from a compound node structure defined at the next higher stratum. The presence of such a link characterizes the fact that a particular DMU may receive data from the external environment or from another DMU located at the next higher stratum. The term external input is explained with the help of an example.

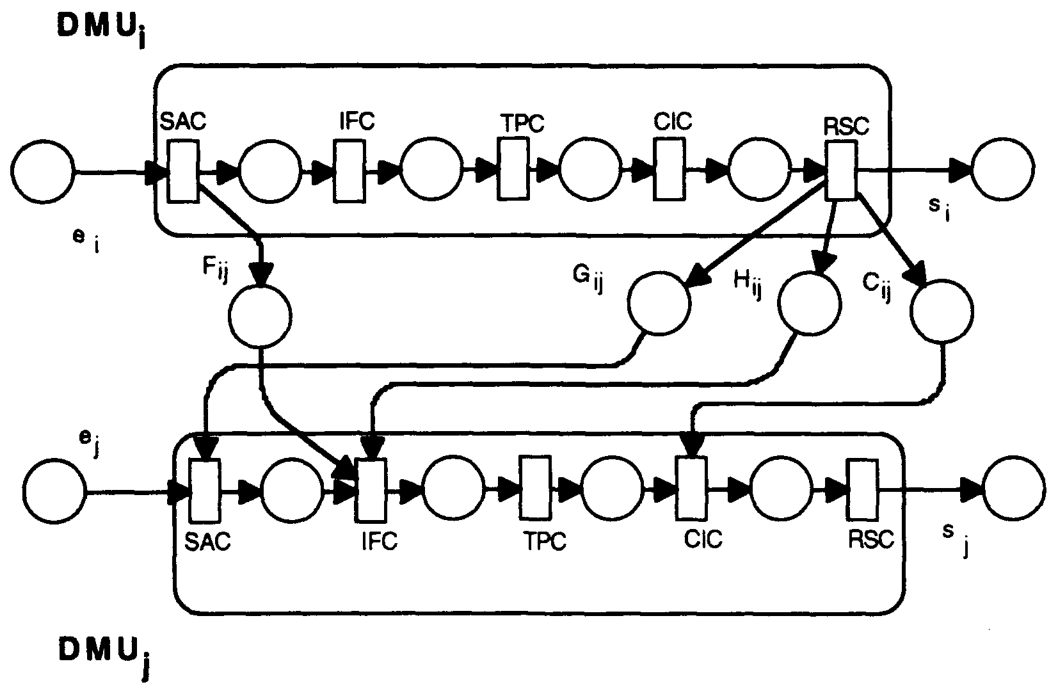


Figure 4.21 Allowable Interactions

Example 4.2

Figure 4.22 shows two compound nodes, DMU_{ik} and DMU_{jk}, and the interactions between them . The organizational structure shown in Figure 4.22 is defined for stratum 'k'. Figure 4.23 shows the system in Figure 4.23 with compound nodes unfolded in order to have a stratum 'k+1' description of the organizational structure. Let each compound node, when it is unfolded, contain two DMUs. The basic decision making units at stratum 'k+1' are defined as DMU_{ak+1}, DMU_{bk+1}, DMU_{ck+1}, and DMU_{dk+1}.

It can be seen in the figure that the links e_a, e_b, e_c exist; The DMUs 'a', 'b', and 'c' are receiving external inputs. The nature of the external input, however, is different for these DMUs. The nodes DMU_{ak+1} and DMU_{bk+1} are receiving inputs from the external environment (the input transition-place pair on the left), while DMU_{ck+1} is receiving an input only from DMU_{ik}. Therefore, an external input at stratum 'k+1' is defined to be an input signal either from the environment or from the systems defined at higher ('k') stratum.

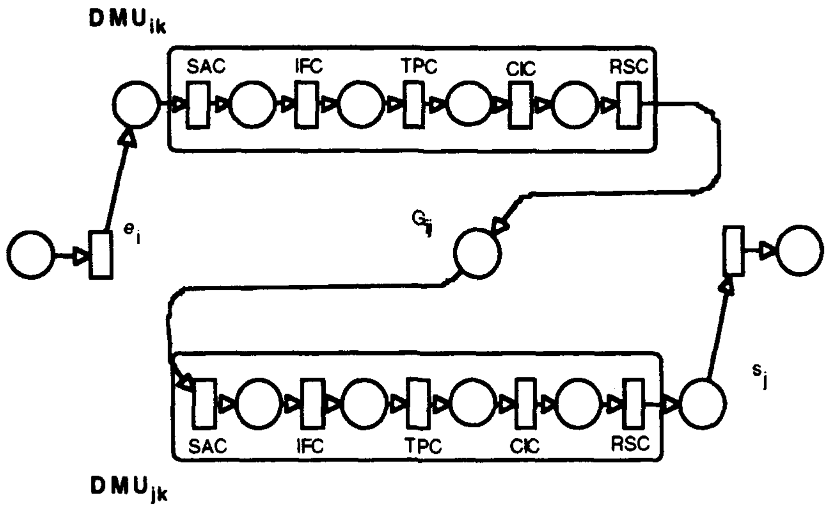


Figure 4.22 Structure in stratum 'k' for Example 4.2

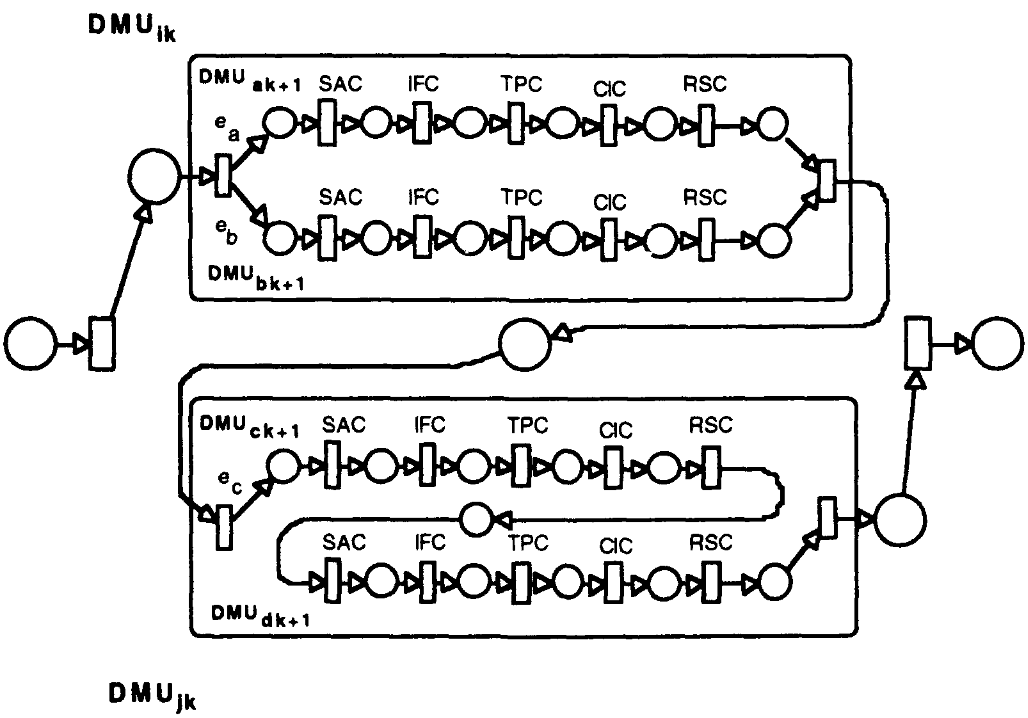


Figure 4.23 Structure in stratum 'k+1' for Example 4.2

• External Output from RSC of DMU_i: s_i

This link represents the *external output* of a decision making compound node to processes external to the organizational structure considered. This output can either be fed to other compound nodes at the next higher stratum or can be directly sent to the external environment depending upon the designer's specifications defined for all other higher strata of the system. The external output is illustrated in Example 4.2, where DMUs 'a', 'b', and 'd' are producing outputs. The output from DMU_{ak+1} and DMU_{bk+1} is fed to the subsystems of DMU_{jk}, while that of DMU_{dk+1} is sent directly to the environment.

• SAC of DMU_i to IFC of DMU_j: F_{ij}

This link models the transmission of assessed situation from compound node i to compound node j. The presence of this link only represents the fact that such an interaction occurs at this stage between two compound node. As a matter of fact this link now characterizes a 'compound link' between two compound nodes. The question of how many such physical links are represented by this 'compound link' is addressed in Chapter VI.

• RSC of DMU_i to SAC of DMU_j: G_{ij}

This interaction models the transmission of *control* from the output of a decision making compound node to the input of another. The two compound nodes are then said to be connected in a serial or tandem arrangement; the processing of one is dependent upon the completion of the processing by the other. The link represented by the coefficient G_{ij} will not be a compound link as explained in the following Chapter.

• RSC of DMU_i to IFC of DMU_j: H_{ij}

This interaction models the result or processed information sharing type of interaction between two decision making compound nodes. The issue of the actual number of links going from one compound node to another when the compound structures are unfolded is addressed in Chapter VI.

• RSC of DMU_i to CIC of DMU_j: C_{ij}

This link represent the flow of instructions or commands from one decision making compound node - DMU_i - to another - DMU_j. It introduces echelon type hierarchical relationship between two sub-organizational structures -compound nodes. The actual number of such links representing this hierarchy at the next lower stratum will be presented in Chapter VI.

When a compound node 'q' in stratum 'k-1' is unfolded in stratum 'k' into an organizational structure with 'm' DMUs, then the maximum number of links (interconnections) between DMUs is given as:

$$(L_{qk})_{max} = 4m^2 - 2m \qquad m \in \mu_k \tag{4.1}$$

4.6 MATHEMATICAL MODEL

The previous section leads to a mathematical representation of interactions between decision making nodes/compound nodes.

4.6.1 Representation of Interactions

The coefficients e_i, s_i, F_{ij}, G_{ij}, H_{ij}, C_{ij} of Figure 4.8 are integer variables taking values in {0, 1}, where 1 will indicate the presence of the corresponding link in the organizational structure at the stratum for which the structure is defined. Note that the value of the coefficient does not indicate the number of such links which actually exist. Similarly, a value 0 for the coefficient will indicate the absence of the link altogether.

The variables are aggregated into two vectors **e** and **s**, and four matrices **F, G, H,** and **C**. As mentioned before, in order to avoid cumbersome notation the stratum and node numbers associated with the six arrays are not shown.

The interaction structure of an m-DMUs compound node 'i', $i \in \mu_k$, is therefore represented by the following tuple.

$$\Sigma_{ik+1} = \{\, \mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C} \,\} \qquad i \in \mu_k, \quad k = 0,1,2,\ldots,n$$

Note that the DMUs of the compound node are defined in stratum 'k+1', Figure 4.24. The structure of the compound node 'i' at stratum 'k' will be the five stage model shown in Figure 4.20. Σ_{ik+1} represents the interactional structure of the compound node 'i', when the level of abstraction used to describe the structure is of stratum 'k+1'. The compound node 'i' itself is defined as a DMU for stratum 'k'.

The six arrays **e, s, F, G, H, C** are defined as follows;

- Two m x 1 vectors **e** and **s** representing the interactions of the m-DMUs ('a' and 'b'), Figure 4.24, with the external processes.

$$\mathbf{e} = [e_a] \qquad a = 1,2,...,m \qquad m \in \mu_{k+1}$$
$$\mathbf{s} = [s_a] \qquad a = 1,2,...,m \qquad m \in \mu_{k+1}$$

- Four m x m matrices F, G, H, C representing the interactions among the decision making nodes/compound nodes of the organizational structure represented by compound node 'i'.

$$\mathbf{F} = [F_{ab}] \qquad \mathbf{G} = [G_{ab}] \qquad a = 1, 2, ..., m$$
$$\mathbf{H} = [H_{ab}] \qquad \mathbf{C} = [C_{ab}] \qquad b = 1, 2, ..., m$$
$$m \in \mu_{k+1}$$

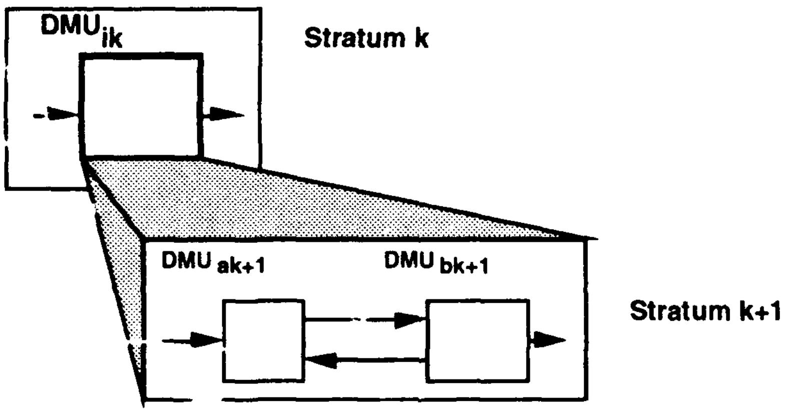


Figure 4.24 A Stratified Organization

The diagonal elements of the matrices **F**, **G**, **H**, and **C** are set to '0'; DMUs are not allowed to interact with themselves.

$$F_{aa} = G_{aa} = H_{aa} = C_{aa} = 0 \quad \text{for } a = 1, 2, \ldots, m \quad \text{where } m \in \mu_{k+1}$$

4.6.2 Well Defined Net

The six-tuple Σ_{ik+1} is called a **Well Defined Net** (WDN) of compound node 'i' which is located at stratum 'k'. The dimension of the WDN is 'm', where 'm' is the number of decision making units (DMU) in 'i'. The set of all WDN of dimension m will be denoted by $W_{ik+1}(m)$. It is clear that the set of all WDNs defined for compound node 'i' is isomorphic to the set $\{0, 1\}^{(L_{ik+1})max}$, Equation 4.1. The dimension of $W_{ik+1}(m)$ is therefore

$$2^{(L_{ik+1})max} = 2^{(4m^2 - 2m)}$$

4.6.3 Lattice Structure of $W_{ik+1}(m)$

The formalism of the lattice theory, as presented in Chapter III, is applied to the set of all Well Defined Nets, $W_{ik+1}(m)$, of a compound node 'i' located at stratum 'k' with dimension m. As a result, an order is defined on this set as follows

Definition 4.2

Let

$\Sigma_{ik+1} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$ and $\Sigma'_{ik+1} = \{\mathbf{e'}, \mathbf{s'}, \mathbf{F'}, \mathbf{G'}, \mathbf{H'}, \mathbf{C'}\}$ be two WDNs of the same dimension m.

Σ'_{ik+1} is said to be a **subnet** of Σ_{ik+1}, denoted by $\Sigma'_{ik+1} \leq \Sigma_{iK+1}$, if and only if

$$\mathbf{e'} \ll \mathbf{e} \qquad \mathbf{F'} \ll \mathbf{F} \qquad \mathbf{G'} \ll \mathbf{G}$$
$$\mathbf{s'} \ll \mathbf{s} \qquad \mathbf{H'} \ll \mathbf{H} \qquad \mathbf{C'} \ll \mathbf{C}$$

The relation '«' has been defined in Chapter 2, i.e., A « A' if and only if every element of A' is less than or equal to the corresponding element of A.

The set of all WDNs is therefore a partially ordered set. The following properties are the result of some of the definitions presented in Chapter III and are therefore given without any further proof.

- The least element of the set of all WDNs of a compound node is defined to be the WDN whose arrays have all their elements equal to 0. It is denoted by ω^m .

- The greatest element of the set of all WDNs of a compound node is defined to be the WDN whose arrays have all their elements equal to 1 except for the diagonal elements of the arrays **F**, **G**, **H**, and **C**. It is denoted by Ω^m.

- A WDN Σ_{ik+1} will **cover** another WDN Σ'_{ik+1} if and only if $\Sigma'_{ik+1} \leq \Sigma_{ik+1}$ and Σ_{ik+1} has exactly one more link than Σ'_{ik+1}, i.e., $d[\Sigma_{ik+1}] = d[\Sigma'_{ik+1}] + 1$.

According to Theorem 3.4 of Subsection 3.2, W_{ik+1} satisfies the Jordan-Dedekind chain condition. The following propositions give a characterization of the join and meet of two WDNs of dimension m.

Proposition 4.2

Let

$\Sigma'_{ik+1} = \{ \mathbf{e}', \mathbf{s}', \mathbf{F}', \mathbf{G}', \mathbf{H}', \mathbf{C}' \}$ and $\Sigma''_{ik+1} = \{ \mathbf{e}'', \mathbf{s}'', \mathbf{F}'', \mathbf{G}'', \mathbf{H}'', \mathbf{C}'' \}$ be two WDNs of dimension m.

The join of Σ'_{ik+1} and Σ''_{ik+1}, $\Sigma_{il} = \Sigma'_{ik+1} \cup \Sigma''_{ik+1}$, will be a WDN represented by the arrays **e, s, F, G, H,** and **C** with

$$\mathbf{e} = \mathbf{e}' \cup \mathbf{e}'' \qquad \mathbf{F} = \mathbf{F}' \cup \mathbf{F}'' \qquad \mathbf{C} = \mathbf{C}' \cup \mathbf{C}''$$

$$\mathbf{s} = \mathbf{s}' \cup \mathbf{s}'' \qquad \mathbf{G} = \mathbf{G}' \cup \mathbf{G}'' \qquad \mathbf{H} = \mathbf{H}' \cup \mathbf{H}''$$

The binary operator '$\cup$' has already been defined in Chapter III. The operator is then extended to arrays taking values on the set {0 , 1}, on an element to element basis. Note that the same notation '$\cup$' has been used for three different operations: the composition law defined on the set {0 , 1}, the extension of this law to arrays, and the join operation between two WDNs.

The *join* of two WDNs is defined to be a new net that contains all the interactions that appear in either one of the two WDNs or both.

Proposition 4.3

Let

$\Sigma'_{ik+1} = \{\mathbf{e}', \mathbf{s}', \mathbf{F}', \mathbf{G}', \mathbf{H}', \mathbf{C}'\}$ and $\Sigma''_{ik+1} = \{\mathbf{e}'', \mathbf{s}'', \mathbf{F}'', \mathbf{G}'', \mathbf{H}'', \mathbf{C}''\}$ be two WDNs of dimension m.

The meet of Σ'_{ik+1} and Σ''_{ik+1}, $\Sigma_{ik+1} = \Sigma'_{ik+1} \cap \Sigma''_{ik+1}$, will be a WDN represented by the arrays **e, s, F, G, H,** and **C** with

$$\mathbf{e} = \mathbf{e}' \cap \mathbf{e}'' \qquad \mathbf{F} = \mathbf{F}' \cap \mathbf{F}'' \qquad \mathbf{C} = \mathbf{C}' \cap \mathbf{C}''$$

$$\mathbf{s} = \mathbf{s}' \cap \mathbf{s}'' \qquad \mathbf{G} = \mathbf{G}' \cap \mathbf{G}'' \qquad \mathbf{H} = \mathbf{H}' \cap \mathbf{H}''$$

The binary operator '$\cap$' has already been defined in Chapter III. The operator '$\cap$' is then extended to arrays taking values on the set {0 , 1}, on an element to element basis. Note that the same notation '$\cap$' has been used for three different operations: the composition law defined on the set {0, 1}, the extension of this law to arrays, and the meet operation between two WDNs.

The *meet* of two WDNs is defined to be a new net that contains only the interactions that appear in both WDNs.

From Propositions 4.2 and 4.3 it is clear that the join and meet of any two WDNs can always be defined and are within the set of WDNs. Since the partially ordered set has a least and a greatest elements, Proposition 4.4 follows.

Proposition 4.4

The set W_{ik+1} of all WDNs of dimension m of a compound node 'i' located at stratum 'k' is a **lattice**.

4.7 CONCLUSION

An introduction to the methodology for generating Stratified Decision Making Organizations (SDMOs) was presented in this chapter. The first part of the chapter was comprised of basic concepts and definitions of Multilevel Hierarchical systems. A review of the model of a single interacting decision maker was then presented. In the second part of the chapter, an organizational structure with human decision makers was folded to obtain a compound node structure. The folding procedure was then generalized to all organizational structures defined at any arbitrary stratum. A compound node model was selected against a number of possible folded structures. The third part of the chapter dealt with the definition and interpretation of the interactions among compound nodes. Finally, a mathematical model was presented for the interactions, and the concept of Well Defined Nets (WDNs) was introduced based on the mathematical model. It was found out that the set of all WDNs W_{ik+1} of a compound node i in stratum k with m subsystems is a lattice.

CHAPTER V

REPRESENTATION OF MULTILEVEL ORGANIZATIONAL CLASSES

5.1 INTRODUCTION

The labeling used by Remy (1986) to represent places and transitions of an organizational structure is extended to label the places and transitions of the stratified organizational structures. Note that the labeling technique used by Remy applies only to the nodes at stratum 'N', the lowest stratum with human decision makers (DM) as DMUs. A brief review of the said labeling technique is given in this section.

The labeling technique used by Remy (1986) was introduced primarily for computational purposes. Table 5.1 gives the labels associated with all possible transitions of an organizational structure.

TABLE 5.1 Labeling of the Transitions

Description	**Label**
Input transition	t_0
Output transition	t_6
SA of decision maker 'i'	t_{1i}
IF of decision maker 'i'	t_{2i}
TP of decision maker 'i'	t_{3i}
CI of decision maker 'i'	t_{4i}
RS of decision maker 'i'	t_{5i}

The generic label of an internal transition of an m-decision maker organization is given by t_{sr} with $1 \leq s \leq 5$ and $1 \leq r \leq m$. The index 's' corresponds to the stage and 'r' to the decision maker.

Table 5.2 indicates the scheme used for labeling all the internal and interactional places of an organizational structure.

TABLE 5.2 Labeling of the Places

Transitions Input		Transitions Output	Corresponding Places
	$\rightarrow$	t_0	p_0
t_0	$\rightarrow$	t_{1i}	p_{1i}
t_{1i}	$\rightarrow$	t_{2i}	p_{2i}
t_{1i}	$\rightarrow$	t_{2j}	p_{2ij}
t_{2i}	$\rightarrow$	t_{3i}	p_{3i}
t_{3i}	$\rightarrow$	t_{4i}	p_{4i}
t_{4i}	$\rightarrow$	t_{5i}	p_{5i}
t_{5i}	$\rightarrow$	t_6	p_{6i}
t_{5i}	$\rightarrow$	t_{1j}	p_{6ij1}
t_{5i}	$\rightarrow$	t_{2j}	p_{6ij2}
t_{5i}	$\rightarrow$	t_{4j}	p_{6ij4}
t_6	$\rightarrow$		p_7

Example 5.1

Figure 5.1 shows two decision maker., 'i' and 'j', with all allowable interactions from decision maker 'i' to decision maker 'j'. The places and transitions are shown labeled according to the schemes defined in Tables 5.1 and 5.2.

The following section describes a modified labeling technique for representing stratified organizational architectures.

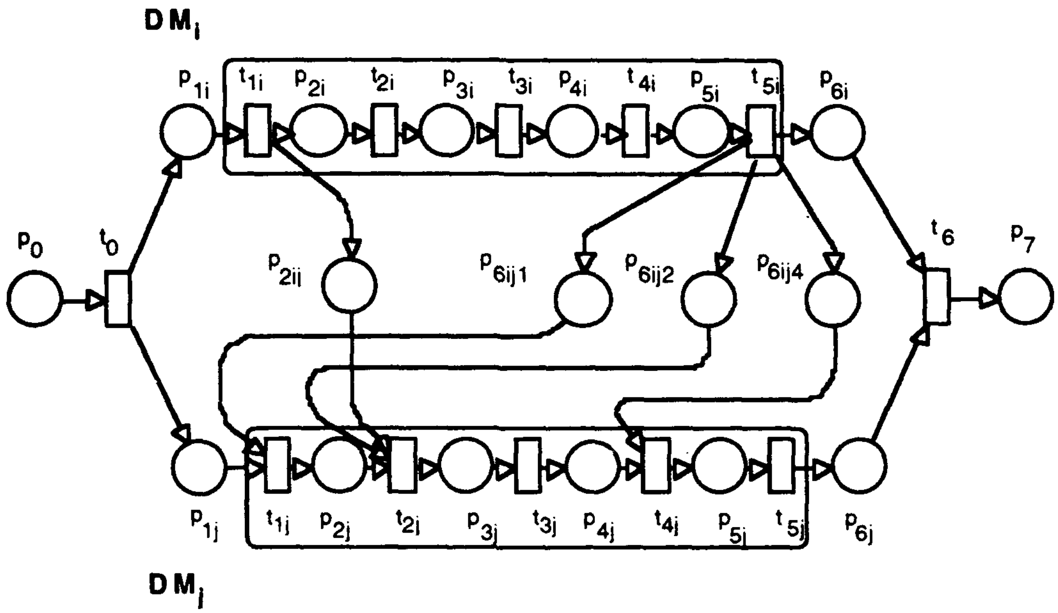


Figure 5.1 Labeling of Transitions and Places

5.2 COMPOUND NODE IN STRATUM 'N-1'

Figure 5.2 shows a stratified organization with n = N = 1. The decision making entities at stratum '1' are DMs 'i' and 'j'. There is a single compound node at stratum '0'. The generic label of DMU is used to represent all the three nodes. The process of folding an organizational structure at stratum '1' to obtain a compound node structure at stratum '0' will give insight to the generalized labeling scheme necessary to be implemented for stratified organizational structures. The generalized labeling scheme will be presented in later sections.

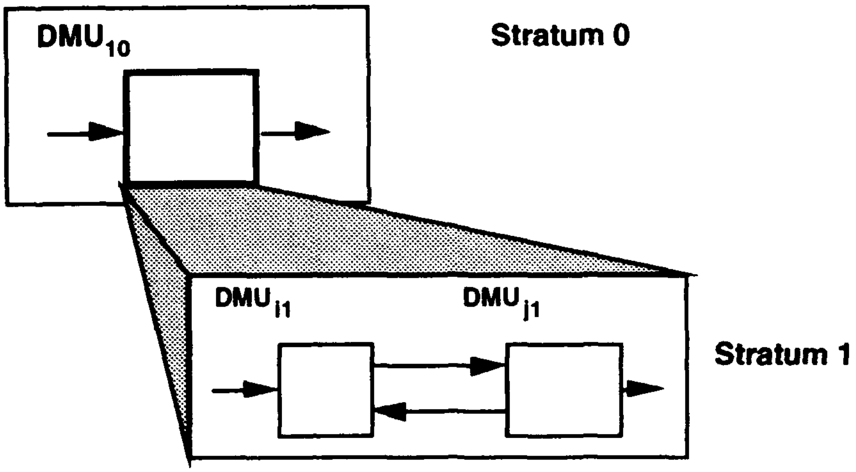


Figure 5.2 Stratified Organization with n = 1

Figure 5.3 shows two DMUs 'i' and 'j' at stratum '1' with all allowable interactions from DMU_{i1} to DMU_{j1}. It is shown in the figure that the labeling scheme presented earlier is modified slightly and one more digit is added to each place and transition label. The added digit depicts the stratum number to which a particular DMU belongs. The label of an internal transition is now given as t_{srk}, where 'k' represents the stratum number for which the DMU is defined. Similarly, the labels for all internal and interactional places are modified. The input and output transitions and places are treated in a different manner. The said places and transitions are assumed to be the processes which belong to the compound node at the stratum 'k' if the interactional structure is described at stratum 'k +1'. Therefore, the label associated with the input and output transitions and places have '0' as the stratum number and '1' as the DMU number instead of '1' for the stratum and 'i' or 'j' for the DMU number.

The modified labeling scheme is given in Table 5.3 and Table 5.4

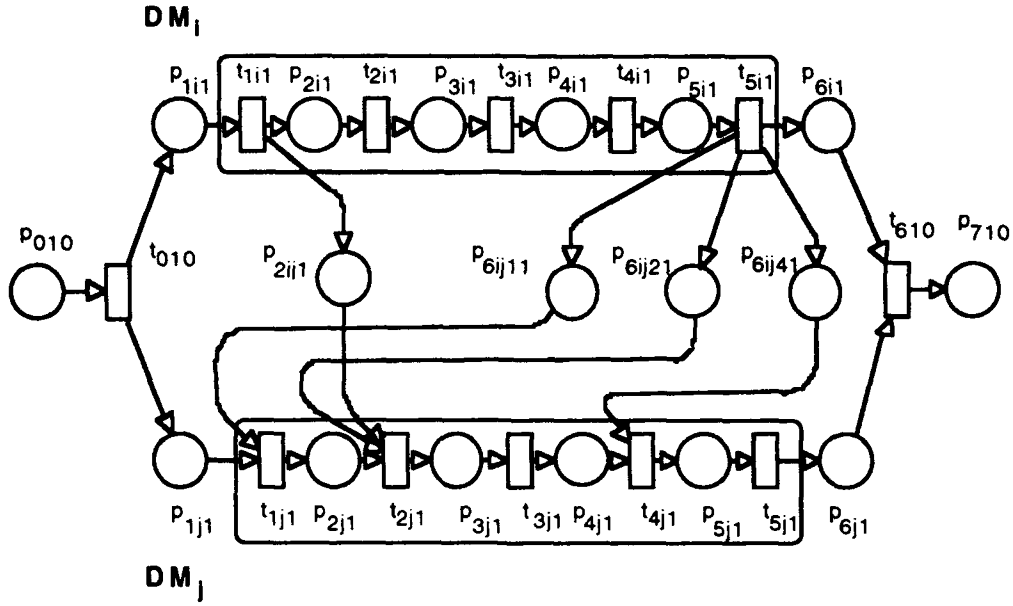


Figure 5.3 Modified Labeling Scheme

TABLE 5.3 Labeling of the Transitions in Figure 5.3

Description	**Label**
Input transition	t_{010}
Output transition	t_{610}
SA of decision maker 'i'	t_{1i1}
IF of decision maker 'i'	t_{2i1}
TP of decision maker 'i'	t_{3i1}
CI of decision maker 'i'	t_{4i1}
RS of decision maker 'i'	t_{5i1}

TABLE 5.4 Labeling of the Places in Figure 5.3

Transitions Input		Transitions Output	Corresponding Places
	→	t_{010}	p_{010}
t_{010}	→	t_{1i1}	p_{1i1}
t_{1i1}	→	t_{2i1}	p_{2i1}
t_{1i1}	→	t_{2j1}	p_{2ij1}
t_{2i1}	→	t_{3i1}	p_{3i1}
t_{3i1}	→	t_{4i1}	p_{4i1}
t_{4i1}	→	t_{5i1}	p_{5i1}
t_{5i1}	→	t_{610}	p_{6i1}
t_{5i1}	→	t_{1j1}	p_{6ij11}
t_{5i1}	→	t_{2j1}	p_{6ij21}
t_{5i1}	→	t_{4j1}	p_{6ij41}
t_{610}	→		p_{710}

The sequence of processes carried out to fold the structure, at stratum '1', in order to obtain a compound node structure of the same organization, in stratum '0', is shown in Figure 5.4 and Figure 5.5. A detailed discussion on folding the organizational structures will be presented in later sections. The folding procedure adopted here follows the discussion in Section 4.4, Chapter IV.

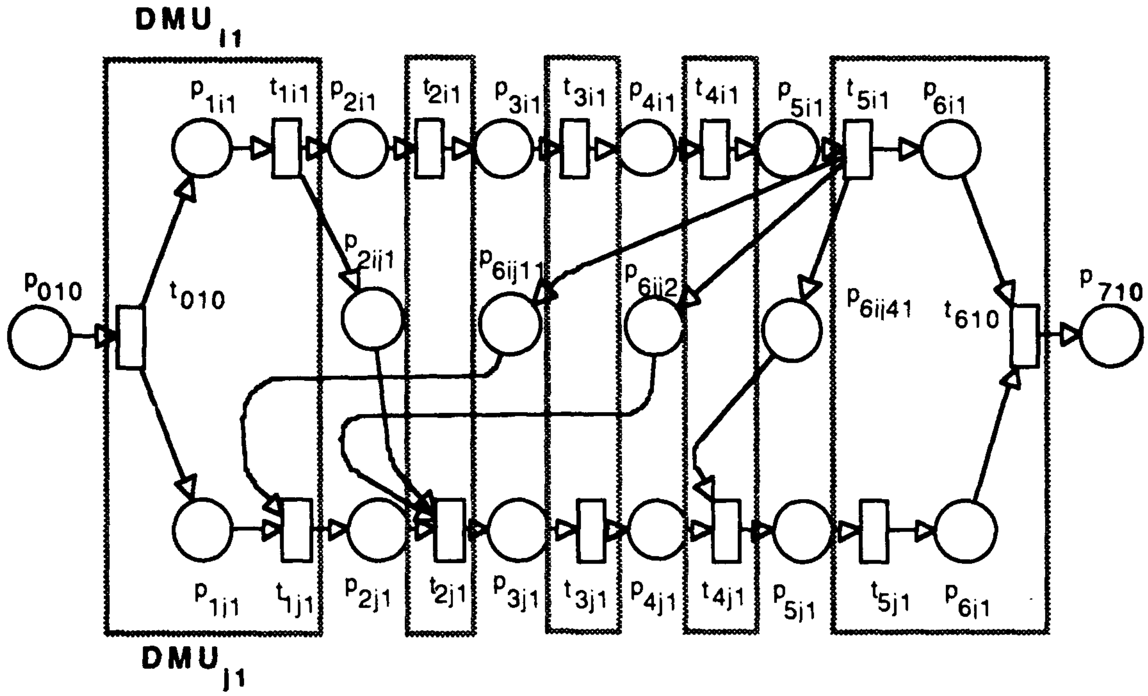


Figure 5.4 Structure to be Folded in Stratum '1'

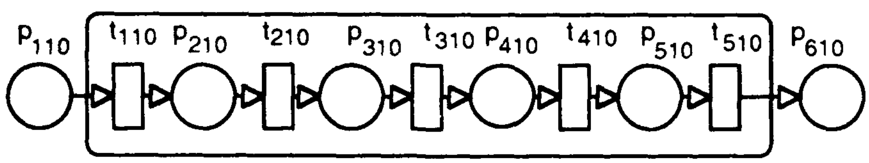


Figure 5.5 Compound Node in Stratum '0'

The labeling of the compound node structure in Figure 5.5 now can be generalized for all compound nodes in an arbitrary stratum 'n-1', $1 \leq n \leq N$. The following sections generalize the results obtained in this section.

5.3 TRANSITIONS OF A WDN IN STRATUM 'k'

The transitions of a compound node are *compound* transitions; they represent a subnet comprised of transitions and places defined at a lower stratum. The following section presents a labeling scheme which is applicable to the transitions of all the DMUs defined in an arbitrary stratum 'k'. In Figure 5.6, a DMU 'q' is shown in stratum 'k-1' with two subsystems, DMUs 'i' and 'j', in stratum 'k'. The DMU 'i' has two subsystems defined at a lower stratum, namely DMUs 'a' and 'b', while DMU 'j' has DMUs 'c' and 'd' as its subsystems at stratum 'k+1'. Figure 5.6 will be used as a reference for the variables defined throughout this chapter.

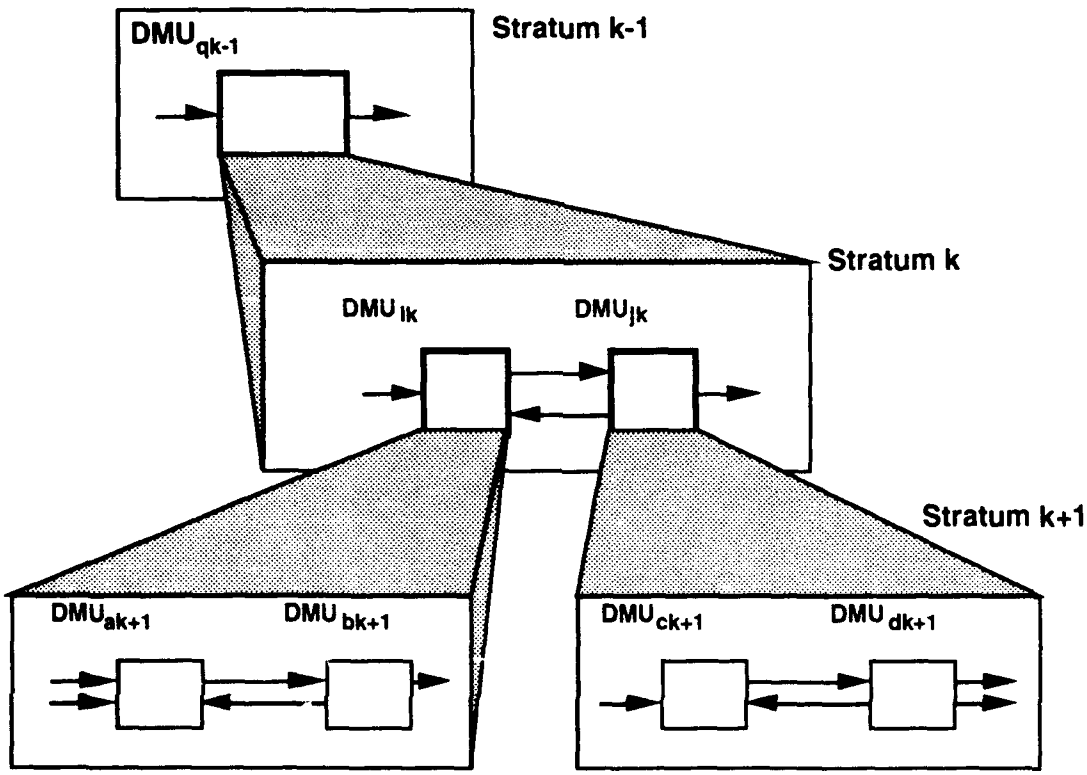


Figure 5.6 A Stratified Organization

Table 5.5 gives the labels associated with all possible transitions of a WDN at stratum 'k'. It can be seen in the table that transitions are labeled to reflect the DMU they belong to, the stage they represent, and the stratum for which they are defined. This labeling technique is introduced primarily for computational purposes and also to provide an algorithmic approach for folding or unfolding an organizational structure. Note the labels of the two supplementary transitions that represent the interactions of the organization with the external environment or processes defined at higher strata. Their labels represent the compound node number of the organizational structure under study. Therefore, it can be said that the labels of the input and output transitions of an organizational structure behave as a mapping function between the system defined at a higher stratum to its subsystems (DMUs) in the lower stratum. Whenever organizational structures are folded, with portions of the net replaced by compound transitions, only those places and transitions are compounded that belong to the subsystems of a single compound node in higher stratum. The compound transitions inherit the node number from the labels of input and output transitions of the lower stratum system's description.

TABLE 5.5 Modified Labeling of Transitions

Description	Label
Input transition	t_{0qk-1}
Output transition	t_{6qk-1}
SA/SAC of DMU 'i'	t_{1ik}
IF/IFC of DMU 'i'	t_{2ik}
TP/TPC of DMU 'i'	t_{3ik}
CI/CIC of DMU 'i'	t_{4ik}
RS/RSC of DMU 'i'	t_{5ik}

The generic label of an internal transition will now be t_{srk} with $1 \leq s \leq 5$ and $1 \leq r \leq m$ and $0 \leq k \leq n$. The index 's' represents the stage, 'i' the DMU number, and 'k' the stratum.

5.4 PLACES OF A WDN IN STRATUM 'k'

The places of a WDN defined for a stratum 'k', where $0 \leq k \leq n-1$, are compound places obtained by folding the places of the WDNs defined at a lower stratum - stratum 'k+1'. The places of the Petri Net representation of a WDN at a stratum 'k' can be partitioned into four groups: **Internal** places represent connections that remain within the boundaries of a node or compound node; **Interactional** places are the places which correspond to the interactions among nodes or among compound nodes; a **sink** and a **source** represent the interaction of the WDN with processes defined at higher strata. The following sections present a labeling scheme that could be used to represent the places of an organizational structure for a given stratum.

5.4.1 Internal Places of a Decision Maker

There are four types of internal places characterized by the stages they are related to:

- SA/SAC to IF/IFC
- IF/IFC to TP/TPC
- TP/TPC to CI/CIC
- CI/CIC to RS/RSC

As mentioned earlier, all five stages (SA, IF, TP, CI, and RS) need not be present in a particular node defined at stratum 'N'. On the other hand, a compound node structure will have all the five stages (SAC, IFC, TPC, CIC, and RSC). The existence of an internal place in a node at stratum 'N' is determined by the following rules (Remy 1986):

- SA to IF: A place will exist between the SA and IF stages of a node (decision maker) if and only if

    * SA has at least one interactional input place **and** IF has at least one interactional input place.

or

* SA has at least one interactional input place **and** CI has at least **one** interactional input place.

or

* SA has at least one interactional input place **and** RS has at least **one** interactional output place.

• IF to CI: A place will exist between the IF an CI stages of a node if and only if IF has at least one input place (interactional or internal).

• CI to RS: A place will exist between the CI and RS stages of a decision maker if and only if CI has at least one input input place (interactional or internal).

Note that the above rules insure that the net representing a decision maker can not be partitioned into two separate subnets. The rules also guarantee compliance with Section 4.2 where it is stated that only four internal configurations of a decision maker are allowable : SA alone, SA-IF-TP-CI-RS, IF-TP-CI-RS, and CI-RS.

5.4.2 Labeling of Places

The labeling of places, like the labeling of transitions, is introduced primarily for computational purposes. It also provides an algorithmic approach for folding and unfolding the organizational structures at different strata. A description of the labeling technique follows. A place will be labeled with a minimum of three and a maximum of five digits. The minimum number of digits necessary to completely characterize a place will be used. The complete characterization of a place involves the stage it represents, the DMU to which it belongs, and the stratum for which it is defined. Three digits are used to characterize internal places: The first one corresponds to the **input stage of the place**, the second one corresponds to the **DMU**, and the third one corresponds to the stratum where the DMU is located. The interactional places representing the sharing of assessed situation between two DMUs require four digits. The first digit characterizes the type of place under consideration - for example, an SA/SAC to IF/IFC interactional place -, the other two specify the DMUs sharing this information, while the last digit characterizes the stratum number where the two DMUs are located. Lastly, the remaining interactional places among the DMUs are labeled with five digits. The first one characterizes the type of place, the second and third ones specify the DMUs exchanging information, the fourth one allows to differentiate between output stages (SA/SAC, IF/IFC, CI/CIC) and the last one indicates the stratum number. The modified labeling scheme for places of a WDN is given in the Table 5.6. The variables used in the labels correspond to variables described in Figure 5.6.

TABLE 5.6 Modified Labeling of Places

Transitions Input		Transitions Output	Corresponding Place Label
	$\rightarrow$	t_{0qk-1}	p_{0qk-1}
t_{0qk-1}	$\rightarrow$	t_{1ik}	p_{1ik}
t_{1ik}	$\rightarrow$	t_{2ik}	p_{2ik}
t_{1ik}	$\rightarrow$	t_{2jk}	p_{2ijk}
t_{2ik}	$\rightarrow$	t_{3ik}	p_{3ik}
t_{3ik}	$\rightarrow$	t_{4ik}	p_{4ik}
t_{4ik}	$\rightarrow$	t_{5ik}	p_{5ik}
t_{5ik}	$\rightarrow$	t_{6qk-1}	p_{6ik}
t_{5ik}	$\rightarrow$	t_{1jk}	p_{6ij1k}
t_{5ik}	$\rightarrow$	t_{2jk}	p_{6ij2k}
t_{5ik}	$\rightarrow$	t_{4jk}	p_{6ij4k}
t_{6qk-1}	$\rightarrow$		p_{7qk-1}

5.5 MAXIMUM NUMBER OF TRANSITIONS AND PLACES

5.5.1 Maximum Number of Transitions and Places in a WDN

When a compound node 'q' in stratum 'k-1' is unfolded in stratum 'k' into an organizational structure with 'm' DMUs, then the maximum number of transitions in that WDN is given as:

$$(M_{qk})_{max} = 5m + 2 \qquad q \in \mu_{k-1} \qquad m \in \mu_k \tag{5.1}$$

The maximum number of places in the m-dimensional WDN can be determined as presented in Table 5.7. Places are listed according to their input transitions, which is equivalent to a listing according to the first digit of the numerical part of their label.

TABLE 5.7 Maximum Number of Places in a WDN

Description	**Corresponding Place Label**	**Maximum Number**
• Source Place	p_{0qk-1}	1
• Source → SA/SAC	p_{1ik}	$N_{1qk} = m$
• SA/SAC → IF/IFC	p_{2ik}	
	p_{2ijk}	$N_{2qk} = m^2$
• IF/IFC → TP/TPC	p_{3ik}	$N_{3qk} = m$
• TP/TPC → CI/CIC	p_{4ik}	$N_{4qk} = m$
• CI/CIC → RS/RSC	p_{5ik}	$N_{5qk} = m$
RS/RSC → SA/SAC	p_{6ij1k}	$m^2 - m$
→ IF/IFC	p_{6ij2k}	$m^2 - m$
→ CI/CIC	p_{6ij4k}	$m^2 - m$
→ Sink	p_{6ik}	m
• Subtotal RS/RSC →		$N_{6qk} = 3m^2 - 2m$
Sink Place	p_{7qk-1}	1
TOTAL		$4m^2 + 2m + 2$

The maximum number of places of a WDN is obtained by adding the starred entries in Table 5.7 and is given as:

$$(N_{qk})_{max} = 4m^2 + 2m + 2 \quad (5.2)$$

5.5.2 Maximum Number of Transitions and Places in Stratum 'k'

When an organizational structure at stratum '0' is decomposed at an arbitrary stratum 'k', where $1 \leq k \leq n$, the maximum number of transitions in that description of the structure is given as:

$$M_{k\,max} = 5|\mu_k| + 2\sum_{l=0}^{k-1} |\mu_l| \quad (5.3)$$

The first term in Equation 5.3 corresponds to the total number of transitions representing different stages (SA/SAC, IF/IFC, TP/TPC, CI/CIC, RS/RSC) of the DMUs in stratum 'k'. Since the number of DMUs in the organizational structure is determined by the total number $|\mu_k|$ of the set μ_k of all the nodes in stratum 'k', the maximum number of transitions in the net at stratum 'k' will be five (total number of stages in a DMU) times the total number of DMUs $|\mu_k|$. The second term in the equation accounts for the exact number of input and output transitions in the net at stratum 'k'. Since each compound node has exactly one input and one output transitions when it is unfolded to the next lower stratum, the total number of such transitions in an arbitrary stratum 'k' will be two times the total number of compound nodes that are unfolded in the process of unfolding the organizational structure from stratum '0' to stratum 'k'. From the discussion above it can be easily inferred that Equation 5.1 is an special case of Equation 5.3, where $|\mu_k| = m$ and $k = 1$.

The maximum number of places in the organizational structure at stratum 'k' is determined by Table 5.8.

The maximum number of places of an organizational structure is obtained by adding all the starred entries in Table 5.8.

$$N_{k\,max} = 4\,|\mu_k|^2 + 2\,|\mu_k| + 2\sum_{l=0}^{k-1} |\mu_l| \tag{5.4}$$

TABLE 5.8 Maximum Number of Places in an Organizational Structure in stratum 'k'

Description	Maximum Number
• Source Place	$\sum_{l=0}^{k-1} \lvert\mu_l\rvert$
• Source → SA/SAC	$N_{1k} = \lvert\mu_k\rvert$
• SA/SAC → IF/IFC	$N_{2k} = \lvert\mu_k\rvert^2$
• IF/IFC → TP/TPC	$N_{3k} = \lvert\mu_k\rvert$
• TP/TPC → CI/CIC	$N_{4k} = \lvert\mu_k\rvert$
• CI/CIC → RS/RSC	$N_{5k} = \lvert\mu_k\rvert$
RS/RSC → SA/SAC	$\lvert\mu_k\rvert^2 - \lvert\mu_k\rvert$
→ IF/IFC	$\lvert\mu_k\rvert^2 - \lvert\mu_k\rvert$
→ CI/CIC	$\lvert\mu_k\rvert^2 - \lvert\mu_k\rvert$
→ Sink	$\lvert\mu_k\rvert$
• Subtotal RS/RSC →	$N_{6k} = 3\,\lvert\mu_k\rvert^2 - 2\,\lvert\mu_k\rvert$
• Sink Place	$\sum_{l=0}^{k-1} \lvert\mu_l\rvert$

5.6 CORRESPONDENCE BETWEEN MATRIX AND PETRI NET REPRESENTATIONS

There is a direct-one-to one correspondence between interactional places and the non zero elements of the matrix representation of a WDN described in Section 4.6. Each '1' in the arrays representing a WDN corresponds to the presence of a particular kind of interaction between two DMUs or between a DMU and the external environment. Table 5.9 lists all possible *links* and gives for each of them the correspondence between the matrix and Petri Net representations. A link characterizes the presence of a particular kind of interaction in terms of a place and its input and output arcs. The actual number of such links present in the lower-strata description of the organization will be determined in Chapter VI. In Table 5.9, the Petri Net representation of the links is given in terms of the interactional places and their corresponding input and output transitions.

TABLE 5.9 Correspondence Between Matrix and Petri Net Representations

Matrix Representation	Corresponding Transitions Input		Output	Corresponding Place Label
$e_i = 1$	t_{0qk-1}	→	t_{1ik}	p_{1ik}
$s_i = 1$	t_{5ik}	→	t_{6qk-1}	p_{6ik}
$F_{ij} = 1$	t_{1ik}	→	t_{2jk}	p_{2ijk}
$G_{ij} = 1$	t_{5ik}	→	t_{1jk}	p_{6ij1k}
$H_{ij} = 1$	t_{5ik}	→	t_{2jk}	p_{6ij2k}
$C_{ij} = 1$	t_{5ik}	→	t_{4jk}	p_{6ij4k}

Once the interactional places are defined, internal places are uniquely determined. The rule for the determination of internal places is trivial for compound nodes as all the stages must be present in a compound node structure. Therefore, no matter what interactions one compound node has, it will contain all the internal places. The rules for the determination of internal places for a DMU representing a human decision maker is presented in Subsection 5.4.1. These rules were developed by Remy (1986).

The development in this section will be illustrated with two examples. In both cases the matrix representation is given first. The interactional places of the Petri Net are then defined and then the internal places are added to complete the structure. The places and transitions are annotated with appropriate labels.

5.6.1 Example 5.1

Figure 5.7 gives the matrix representation of Σ_{12}, a 2-dimensional WDN of a compound node '1' in stratum '1' with the interactional structure defined for stratum '2'. The stratum '2' is defined to be the lowest stratum with n = N = 2, therefore, the DMUs in stratum '2' are human decision makers (DMs). Figure 5.8 shows the corresponding interactional places in the Petri Net representation of Σ_{12}. Finally, Figure 5.9 presents the entire Petri Net structure, with internal, input, and output places added.

$$\Sigma_{12} = \{\ \mathbf{e, s, F, G, H, C}\ \}$$

$$e = [\,1 \quad 1\,] \qquad F = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$s = [\,0 \quad 1\,] \qquad H = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Figure 5.7 Matrix Representation of Σ_{12}

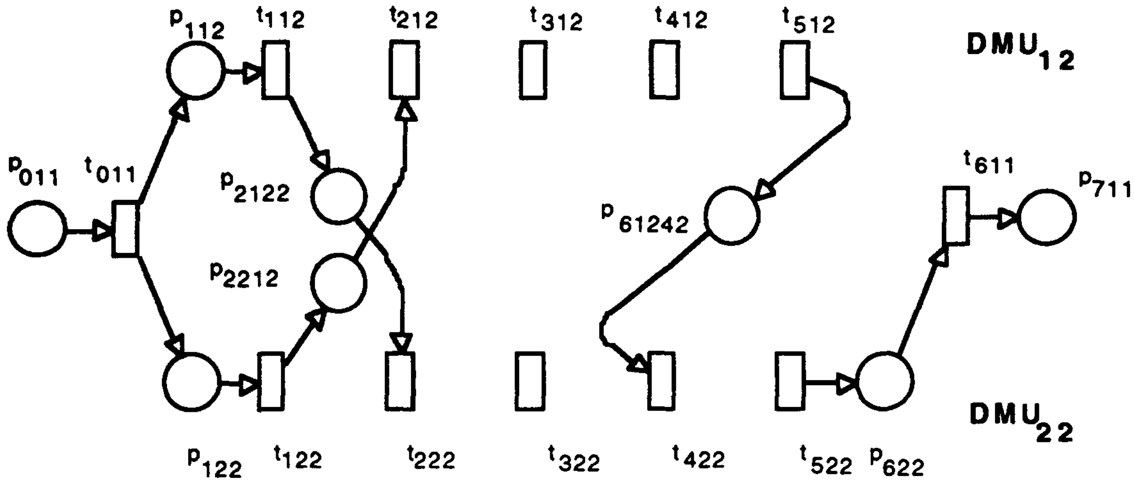


Figure 5.8 Interactional Places of Σ_{12}

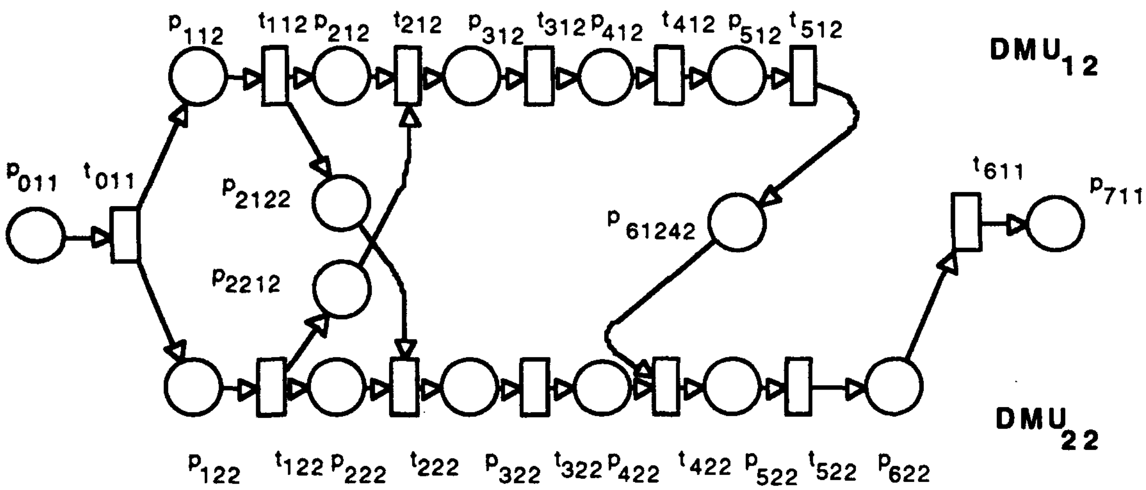


Figure 5.9 Petri Net Representation of Σ_{12}

5.6.2 Example 5.2

Figure 5.10 gives the matrix representation of Σ_{22}, a 4-dimensional WDN of a compound node '2' in stratum '1' with the interactional structure defined for stratum '2'. The stratum '2' is defined as the lowest stratum (n = N = 2). The DMUs comprising the compound node '2' are DMU_{32}, DMU_{42}, DMU_{52}, and DMU_{62}. Figure 5.11 shows the corresponding interactional places in the Petri Net representation of Σ_{22}. Figure 5.12 presents the entire Petri Net structure.

$$\Sigma_{22} = \{ \mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C} \}$$

$$e = [\,0 \quad 0 \quad 1 \quad 1\,] \qquad F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$s = [\,1 \quad 0 \quad 0 \quad 1\,] \qquad H = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 5.10 Matrix Representation of Σ_{22}

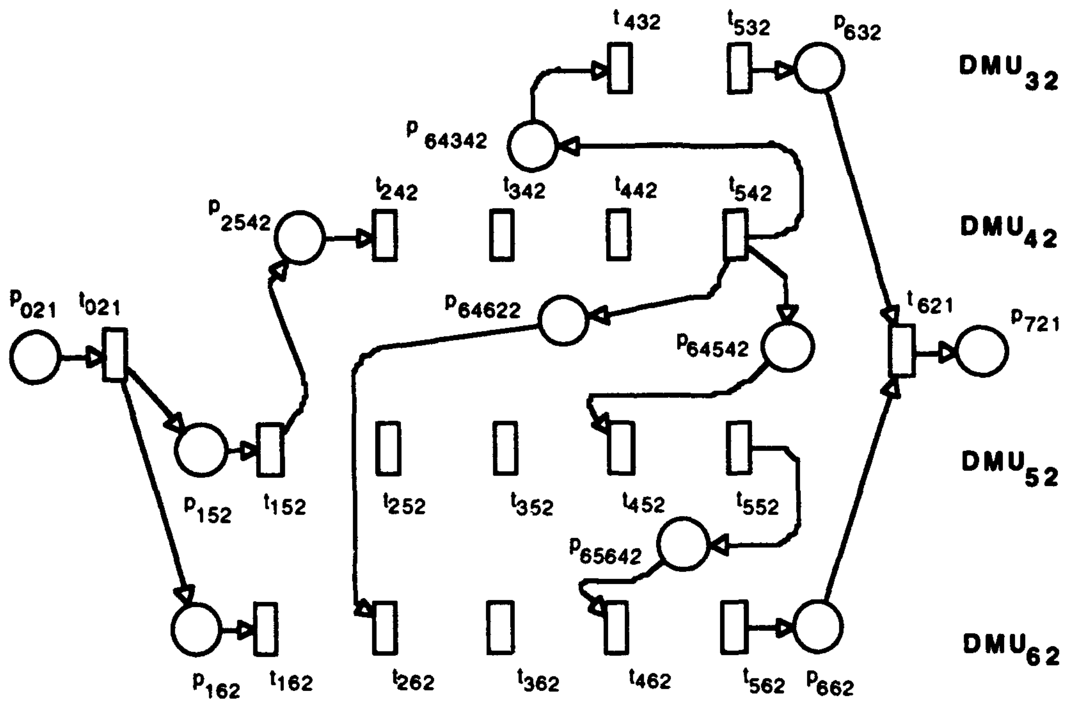


Figure 5.11 Interactional Places of Σ_{22}

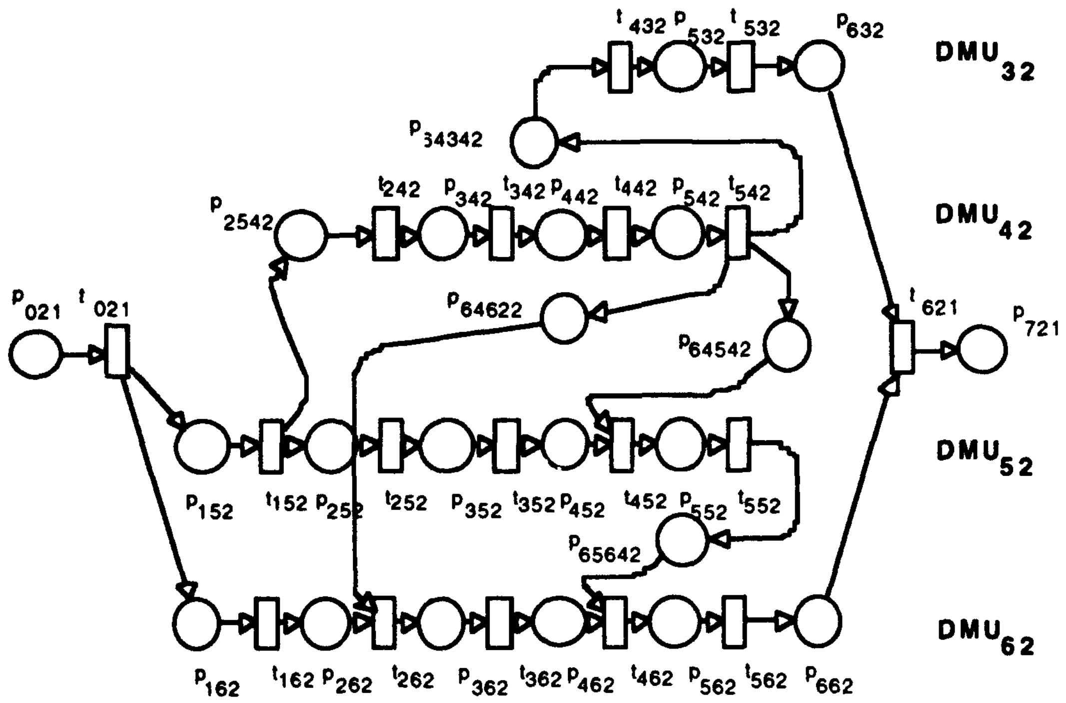


Figure 5.12 Petri Net Representation of Σ_{22}

5.7 INCIDENCE MATRIX

As described in Chapter II, a Petri Net can be represented by an integer matrix reflecting its topological structure. The matrix, called incidence matrix, is the basis of a number of algebraic computations that are made on pure Petri Net structures. This section shows the construction of the incidence matrix of a WDN and its folding or unfolding as a result of describing the same organizational structure at different strata. An example is presented at the end of this section to illustrate the results.

5.7.1 Construction of the Incidence Matrix

As mentioned in Subsection 4.6.2, Σ_{qk} represents the WDN in stratum 'k' of a compound node 'q' in stratum 'k-1'. The incidence matrix $\Delta_{q,k-1,l}$ of the WDN Σ_{qk} is defined as follows. $\Delta_{q,k-1,l}$ is a N_l x M_l matrix, where $k-1 \leq l \leq n$ is the stratum at which the organizational structure of 'q' is described by the matrix. The columns of $\Delta_{q,k-1,l}$ correspond to the transitions of the net and the rows to the places of the Petri Net representation of the node 'q' in stratum 'l'. The matrix $\Delta_{q,k-1,k-1}$ of the compound node 'q' is trivially shown in Figure 5.13. Note that the matrices of all the compound nodes have the same structure as that of Figure 5.13. The folding or unfolding of matrices to obtain different strata description of an organizational structure is discussed in later sections.

	t_{1qk-1}	t_{2qk-1}	t_{3qk-1}	t_{4qk-1}	t_{5qk-1}
p_{1qk-1}	-1	0	0	0	0
p_{2qk-1}	1	-1	0	0	0
p_{3qk-1}	0	1	-1	0	0
p_{4qk-1}	0	0	1	-1	0
p_{5qk-1}	0	0	0	1	-1
p_{6qk-1}	0	0	0	0	1

Figure 5.13 Matrix Representation of a Compound Node

The elements of the incidence matrix take values in {-1, 0, 1}. An element with 1 or -1 as its value indicates the presence of a single link at the very stratum for which the incidence matrix is defined and also depicts the presence of lower strata elements of the block of incidence matrix representing lower-strata connections of the link. Therefore, the elements of the incidence matrix represents blocks, if the lower strata description of an organizational structure is desired.

The determination of the non-zero elements of $\Delta_{q,k-1,k}$ (the incidence matrix of the WDN with 'm' DMUs, where $m \in \{i, j\}$, in stratum 'k', of a compound node 'q' in stratum 'k-1') is done as follows: The labeling of places presented in Section 5.4.2 is designed in such a manner that, once a place is identified in the net, its input and output transitions can be determined by just reading the place label. The location of the non-zero elements can be determined by the input and output transitions. The labeling of the places is the only information used to determine explicitly the elements of $\Delta_{q,k-1,k}$ associated with the compound node 'q' in stratum 'k-1'.

In the following development, transition corresponding to the n-th column of $\Delta_{q,k-1,k}$ will be denoted by $t_{s'm'k}$, where s' denotes the stage ($0 \le s' \le 6$), $m' \in \{i, j\}$, and 'k' depicts the stratum number. To characterize α_{rn} (the element of the r-th row and n-th column), the Kronecker delta is used. The Kronecker delta is defined as follows:

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

Let us consider the r-th row of $\Delta_{q,k-1,k}$; this row will correspond to a place. Five cases are distinguished according to the number of digits in the label of a place.

- Place has a three digit label: p_{sqk-1}, $s = 0, 7$

  p_{0qk-1} is the source of the organization and has only an output transition, t_{0qk-1}, while p_{7qk-1} is the sink and has only an input transition, t_{6qk-1}. Since all organizational structures are required to have a sink and a source, the following equations hold:

$$\alpha_{rn} = -1 \qquad s = 0 \text{ and } s' = 0 \tag{5.6}$$

$$\alpha_{rn} = 1 \qquad s = 7 \text{ and } s' = 6 \tag{5.7}$$

All other elements of the first and last rows are equal to zero.

- Place has a three digit label: P_{sik}, $1 \leq s \leq 6$

The analytical equations are given as:

$$\alpha_{rn} = (\delta_{i\,s'+1} - \delta_{i\,s'}) \times \delta_{j\,m'} \qquad 1 \leq s' \leq 5 \tag{5.8}$$

$$\alpha_{rn} = (\delta_{i\,s'+1} - \delta_{i\,s'}) \qquad s' = 0 \text{ or } s' = 6 \tag{5.9}$$

- Place has a four digit label: P_{2ijk}

This case is covered by the following equation:

$$\alpha_{rn} = (\delta_{1\,s'} \times \delta_{i\,m'}) - (\delta_{2\,s'} \times \delta_{j\,m'}) \tag{5.10}$$

- Place has a five digit label: $P_{sij\gamma k}$, $\gamma = 1, 2, 4$

A generic index s is used to accommodate places representing interactions introduced by *special constraints*. Special constraints are introduced in Chapter VI. The analytical equation, however, is given as:

$$\alpha_{rn} = (\delta_{s-1\,s'} \times \delta_{i\,m'}) - (\delta_{\gamma\,s'} \times \delta_{j\,m'}) \tag{5.11}$$

Equations (5.6), (5.7), (5.8), (5.9), (5.10), and (5.11) completely characterize the incidence matrix of the compound node 'q' in stratum 'k-1' when represented at stratum 'k'. In order to describe the same organizational structure at lower strata, a methodology of unfolding the organizational structure is presented in Subsection 5.7.4.

5.7.2 Examples

In this section, the incidence matrix of the examples presented in Section 5.5 are constructed with the help of the equations presented in the previous section.

Example 1 (Figure 5.10)

The incidence matrix Δ_{112} of Σ_{12} is given in Figure 5.14

	t_{011}	t_{112}	t_{122}	t_{212}	t_{222}	t_{312}	t_{322}	t_{412}	t_{422}	t_{512}	t_{522}	t_{611}
p_{011}	-1	0	0	0	0	0	0	0	0	0	0	0
p_{112}	1	-1	0	0	0	0	0	0	0	0	0	0
p_{122}	1	0	-1	0	0	0	0	0	0	0	0	0
p_{212}	0	1	0	-1	0	0	0	0	0	0	0	0
p_{2122}	0	1	0	0	-1	0	0	0	0	0	0	0
p_{222}	0	0	1	0	-1	0	0	0	0	0	0	0
p_{2212}	0	0	1	-1	0	0	0	0	0	0	0	0
p_{312}	0	0	0	1	0	-1	0	0	0	0	0	0
p_{322}	0	0	0	0	1	0	-1	0	0	0	0	0
p_{412}	0	0	0	0	0	1	0	-1	0	0	0	0
p_{422}	0	0	0	0	0	0	1	0	-1	0	0	0
p_{512}	0	0	0	0	0	0	0	1	0	-1	0	0
p_{522}	0	0	0	0	0	0	0	0	1	0	-1	0
p_{61242}	0	0	0	0	0	0	0	0	-1	1	0	0
p_{622}	0	0	0	0	0	0	0	0	0	0	1	-1
p_{711}	0	0	0	0	0	0	0	0	0	0	0	1

Figure 5.14 The incidence matrix Δ_{112} of Σ_{12}

Example 2 (Figure 5.13)

The incidence matrix Δ_{212} of Σ_{22} is given in Figure 5.15

5.7.3 Folding an Organizational Structure

In order to obtain the stratum 'l', where $0 \leq l \leq k-1$, description of an organizational structure in stratum 'k', the structure in stratum 'k' is first folded to obtain the stratum 'k-1' description of the organization. In order to fold an organization structure in stratum 'k', all the subsystems and their interactions that are defined in stratum 'k' are folded into compound node structures. These compound nodes are now defined as DMUs of the stratum 'k-1' description of the organization. All the interactions that were defined only in stratum 'k' are no longer present in this description.

	t_{021}	t_{152}	t_{162}	t_{242}	t_{252}	t_{262}	t_{342}	t_{352}	t_{362}	t_{432}	t_{442}	t_{452}	t_{462}	t_{532}	t_{542}	t_{552}	t_{562}	t_{621}
p_{021}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{152}	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{162}	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{252}	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{2542}	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{262}	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
p_{342}	0	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
p_{352}	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0
p_{362}	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0
p_{442}	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0
p_{452}	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0
p_{462}	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0
p_{532}	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0
p_{542}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0
p_{552}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0
p_{562}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0
p_{632}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1
p_{64342}	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	1	0	0	0
p_{64542}	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	0	0	0
p_{64622}	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0
p_{65642}	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	0	0
p_{662}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1
p_{721}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 5.15 The Incidence Matrix Δ_{212} of Σ_{22}

On the other hand, the interactions defined in higher strata are still represented in the description. Therefore, while folding an organizational structure in stratum 'k' to obtain the stratum 'k-1' description of the organization, only those subsystems and interactions are folded that have 'k' as their stratum number in their labeling scheme with the exception of input and output transitions and places that have 'k-1' as the stratum number. As mentioned earlier, these transitions and places are used to map the subsystems to their compound node representation in the next higher stratum. Figure 5.16 shows a subsystem of an organizational structure identified in a stratum 'k' description of the organization. The figure also shows an interactional place that is defined at stratum 'k-1'. The place represents an interaction between the subsystem identified and some other subsystem of the organization not shown in the figure. Note that the subsystem identified for illustration has only two DMUs 'i' and 'j' and all the allowable interactions from 'i' to 'j' are shown in the figure. The reason for selecting two DMUs for illustration is evident from the fact that the interactional structures of organizations or suborganizations are defined in terms of the interactions between pairs of their DMUs. Therefore, the folding process illustrated by two DMUs can be applied to any number of DMUs comprising an organization or suborganization. Figure 5.17 presents the compound node structure of the subsystem in Figure 5.16. Note that all the interactions defined in stratum 'k' do not have their representation in stratum 'k-1' description of the subsystem, whereas the interactional place $p_{6rq4k-1}$ is present in the description. It can also be seen that the transitions and places of the compound node inherited the compound node number from the input and output places and transitions of the subsystem in Figure 5.16.

Once an organizational structure in stratum 'k' is folded to stratum 'k-1', the same procedure can be applied iteratively to fold the structure to any stratum higher than the current stratum. Note that the folding procedure must be applied sequentially; it is not possible to fold the structure in stratum 'k' to stratum 'k-2' without having an intermediate stratum 'k-1' description of the organization. Also note that the nets obtained after folding process are executable Petri Nets.

Figure 5.18 presents all the subnets of the structure in Figure 5.16 that are replaced by compound transitions. The places annotated with labels of the form 'B in/out' represent port nodes. The concept of port nodes has been presented in Section 4.4, Chapter IV.

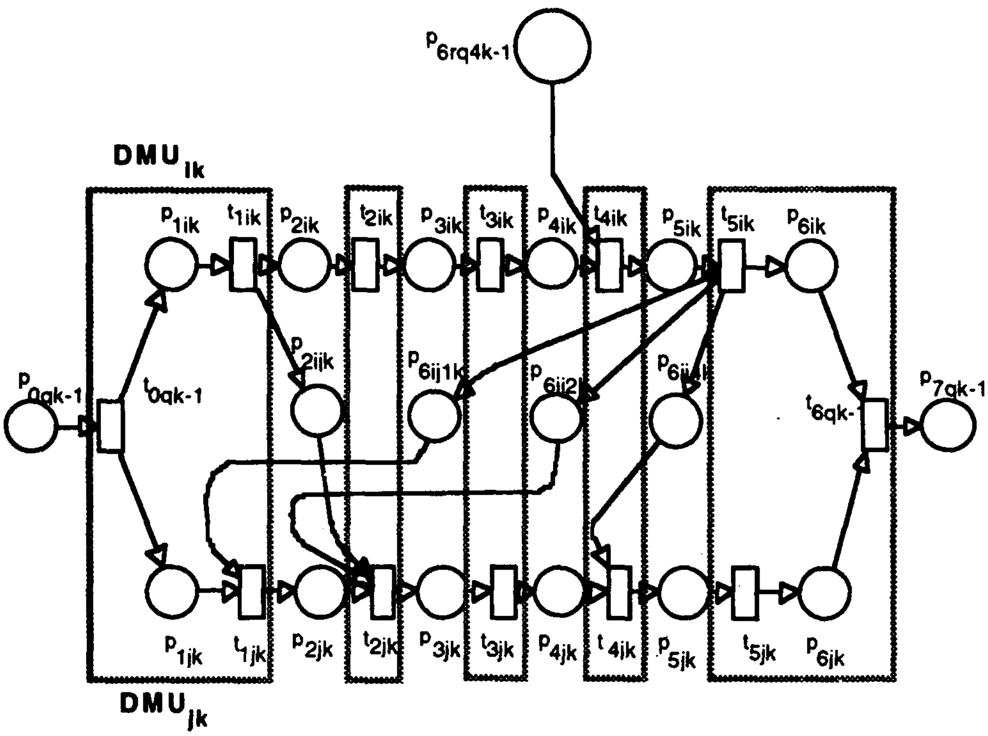


Figure 5.16 Subsystem of an Organizational Structure

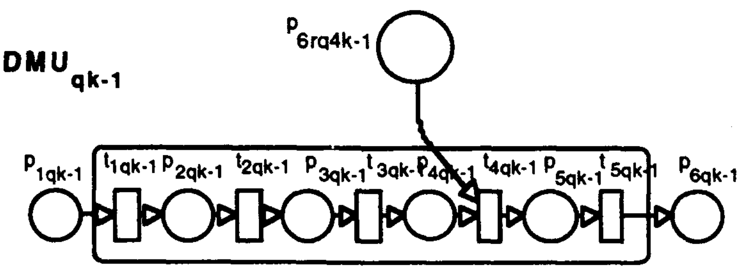


Figure 5.17 Compound Node Representation of the Structure in Figure 5.16

The incidence matrix of the structure in Figure 5.16 is shown in Figure 5.19. The incidence matrix for the compound node structure in Figure 5.17 is calculated by replacing all the incidence matrices of the subnets in Figure 5.18 by their compound transition representation. The places that have their port node representation in the lower stratum description of these compound transitions are grouped according to the scheme presented in

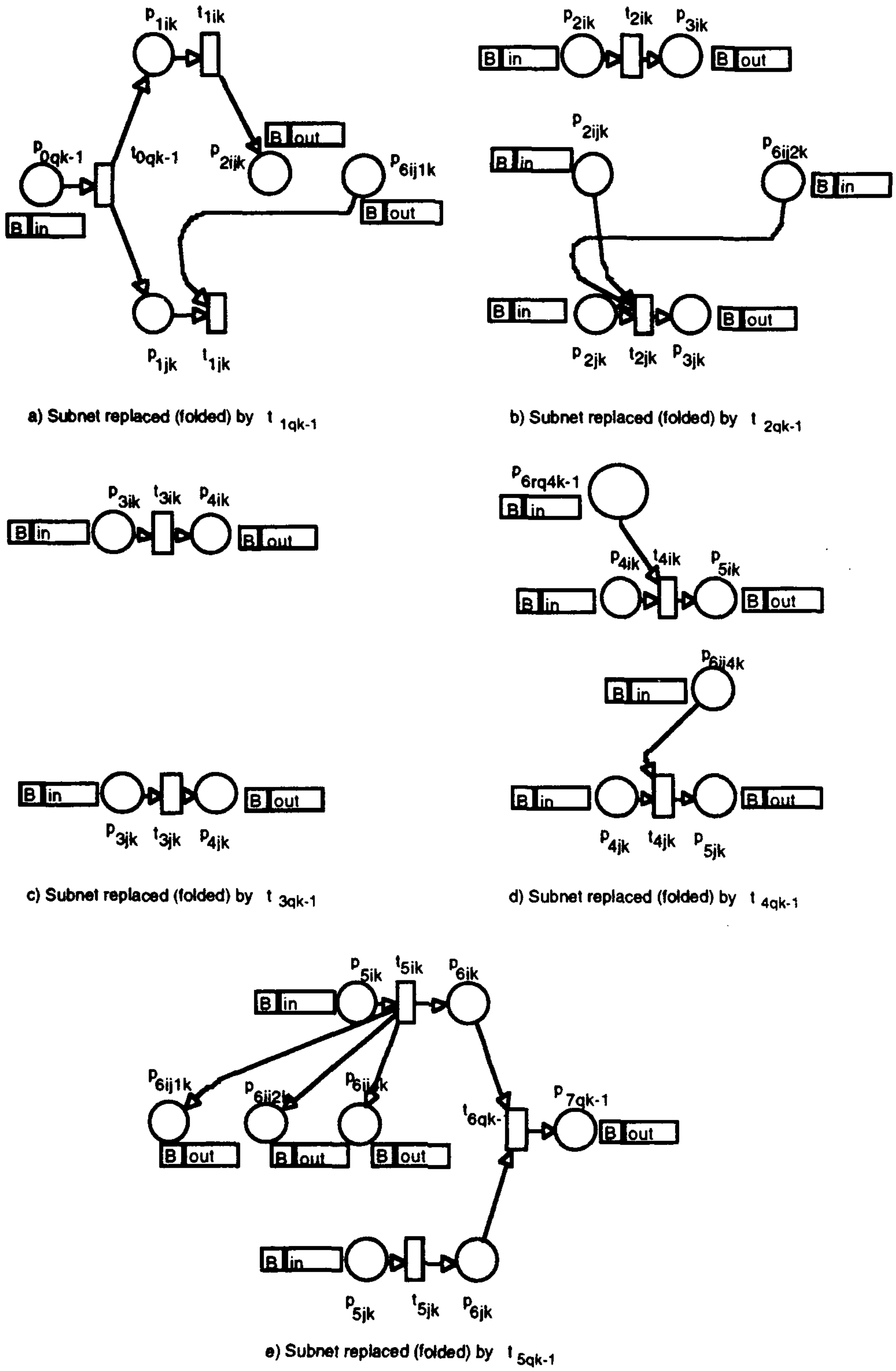


Figure 5.18 Lower Stratum Representation of the Compound Transitions in Figure 5.17

The places p_{1ik}, p_{1jk}, p_{6ik}, and p_{6jk} are not included in Table 5.10 as these places are not defined as the port nodes for any of the subnets presented in Figure 5.18. Note that the interactional places defined in a higher stratum are also not included in the table as they retain their original representation in all those strata that are lower than or equal to the stratum in which they are defined. As a result of folding the incidence matrix of the structure in Figure 5.16, the incidence matrix of the folded structure in Figure 5.17 is shown in Figure 5.20. As stated earlier, the transitions in the incidence matrix in Figure 5.20 of the compound node in Figure 5.17 represent the incidence matrices of the subnets shown in Figure 5.18. The incidence matrix of the subnet that is folded by the compound node transition t_{1qk-1} is shown in Figure 5.21. In Figure 5.21, port nodes are marked with an asterisk (*).

	t_{0qk-1}	t_{1ik}	t_{1jk}	t_{2ik}	t_{2jk}	t_{3ik}	t_{3jk}	t_{4ik}	t_{4jk}	t_{5ik}	t_{5jk}	t_{6qk-1}
p_{0qk-1}	-1	0	0	0	0	0	0	0	0	0	0	0
p_{1ik}	1	-1	0	0	0	0	0	0	0	0	0	0
p_{1jk}	1	0	-1	0	0	0	0	0	0	0	0	0
p_{2ik}	0	1	0	-1	0	0	0	0	0	0	0	0
p_{2jk}	0	0	1	0	-1	0	0	0	0	0	0	0
p_{2ijk}	0	1	0	0	-1	0	0	0	0	0	0	0
p_{3ik}	0	0	0	1	0	-1	0	0	0	0	0	0
p_{3jk}	0	0	0	0	1	0	-1	0	0	0	0	0
p_{4ik}	0	0	0	0	0	1	0	-1	0	0	0	0
p_{4jk}	0	0	0	0	0	0	1	0	-1	0	0	0
p_{5ik}	0	0	0	0	0	0	0	1	0	-1	0	0
p_{5jk}	0	0	0	0	0	0	0	0	1	0	-1	0
p_{6ik}	0	0	0	0	0	0	0	0	0	1	0	-1
p_{6jk}	0	0	0	0	0	0	0	0	0	0	1	-1
p_{6ij1k}	0	0	-1	0	0	0	0	0	0	1	0	0
p_{6ij2k}	0	0	0	0	-1	0	0	0	0	1	0	0
p_{6ij4k}	0	0	0	0	0	0	0	0	-1	1	0	0
p_{711}	0	0	0	0	0	0	0	0	0	0	0	1
$p_{6rq4k-1}$	0	0	0	0	0	0	0	0	-1	0	0	0

Figure 5.19 Incidence Matrix of the Subsystem in Figure 5.16

TABLE 5.10 Equivalent Places in Stratum 'k-1' of the Places in Stratum 'k'

Places in Stratum 'k'	Corresponding Equivalent Places in Stratum 'k-1'
p_{0ik}	p_{1qk-1}
[p_{2ik} p_{2ijk} p_{2ijrk} p_{2jk} p_{2jik} p_{2jirk}]	p_{2qk-1}
[p_{3ik} p_{3jk}]	p_{3qk-1}
[p_{4ik} p_{4jk}]	p_{4qk-1}
[p_{5ik} p_{5jk}]	p_{5qk-1}
[p_{6ij1k} p_{6ij2k} p_{6ij4k} p_{6ijrk} p_{6ji1k} p_{6ji2k} p_{6ji4k} p_{6jirk} p_{7qk-1}]	p_{6qk-1}

$$
\begin{array}{c} \\ p_{1qk-1} \\ p_{2qk-1} \\ p_{3qk-1} \\ p_{4qk-1} \\ p_{5qk-1} \\ p_{6qk-1} \\ p_{6rq4k-1} \end{array}
\begin{array}{c} \begin{array}{ccccc} t_{1qk-1} & t_{2qk-1} & t_{3qk-1} & t_{4qk-1} & t_{5qk-1} \end{array} \\
\left[\begin{array}{ccccc}
-1 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 1 & -1 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & -1 & 0
\end{array}\right] \end{array}
$$

Figure 5.20 Incidence Matrix of the Folded Structure in Figure 5.17

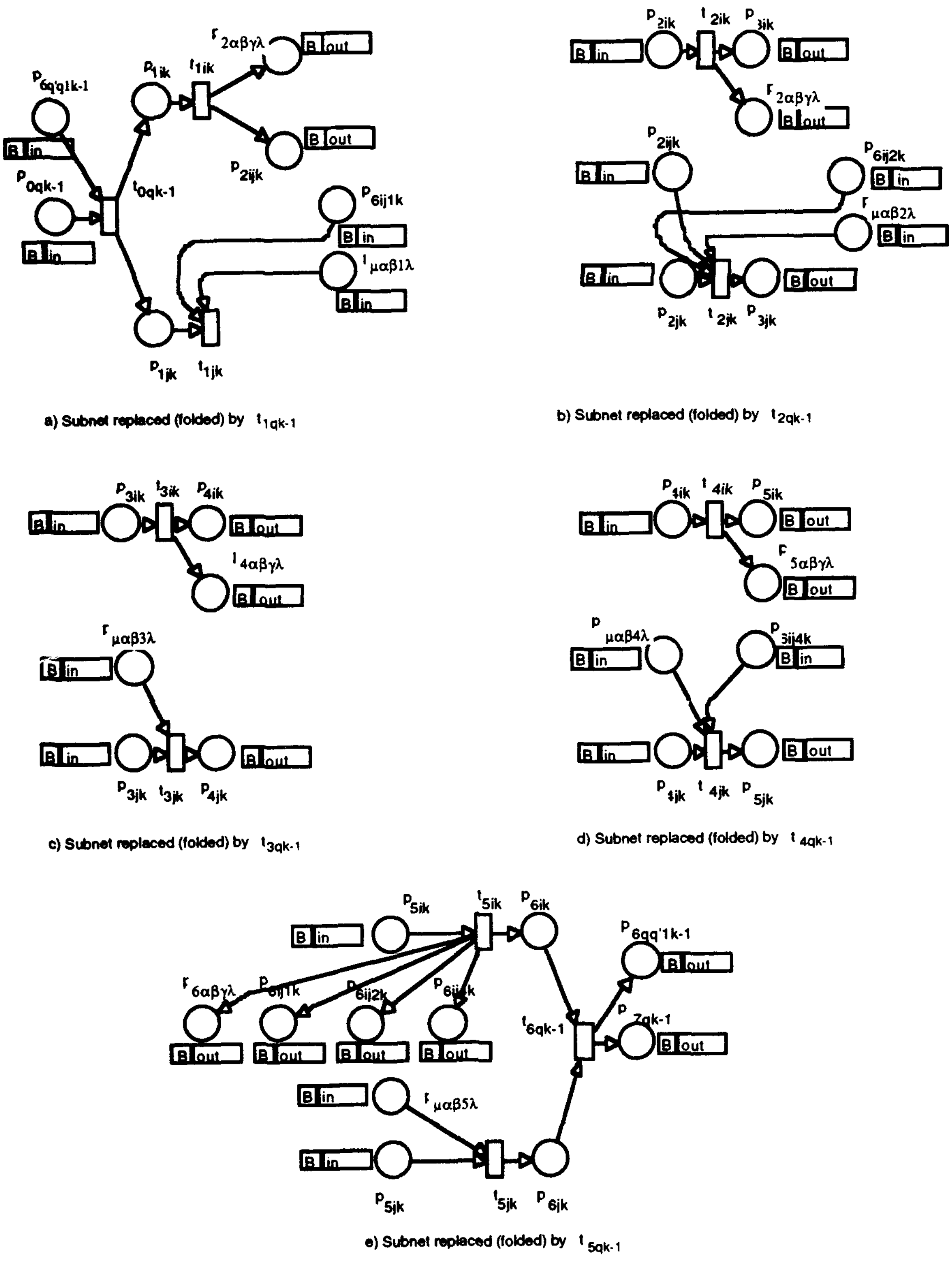
	t_{0qk-1}	t_{1ik}	t_{1jk}
p_{0qk-1}^*	-1	0	0
p_{1ik}	1	-1	0
p_{1jk}	1	0	-1
p_{2ijk}^*	0	1	0
p_{6ij1k}^*	0	0	-1

Figure 5.21 Incidence Matrix of the Subnet Replaced by Compound Transition t_{1qk-1}

The folding procedure can now be generalized as follows:

In order to fold an organizational structure in stratum 'k' to obtain a stratum 'k-1' description of the structure, all the subsystems defined in stratum 'k' are identified by their input and output transition-place pairs. All the interactions defined in stratum 'k' are also identified. The subnets of the organizational structure are then replaced by their compound transition representations and the places of the net identified as defined in stratum 'k' are replaced by their equivalent representation as discussed before. The interactional places that are defined on higher strata than stratum 'k' retain their original representations.

Figure 5.22 presents the lower stratum description, with all possible interactions, of compound transitions t_{1qk-1}, t_{2qk-1},, t_{3qk-1},, t_{4qk-1},, and t_{5qk-1}. In the figure, all the possible input interactions are shown with the transitions representing the stages of DMU_{ik}, while all the possible output interactions are shown with the transitions of DMU_{jk}. In describing different interactions, generic labels are used for interactions that are defined in a stratum 'λ', where $k \le \lambda \le 1$, among generic DMUs a and b. The generic labels 'μ' and 'γ' are used to represent the stages of DMUs, therefore $1 \le \mu, \gamma \le 5$. The generic labels account for all those interactions that are either defined at a higher stratum than stratum 'k' or the interactions implemented by special constraints, R_p. All the places shown with label 'B in/out' are defined as port nodes. The port nodes will retain their existence in the stratum 'k-1' description of the organization, if they are not replaced by their equivalent representation. The transitions and non-port nodes in Figure 5.22 represent the actual subnets being replaced by the compound transitions.



Figures 5.22 Lower Stratum Representation of Compound Transitions

Figures 5.23, 5.24, 5.25, 5.26, and 5.27 show the incidence matrices of the subnets in stratum 'k' that are represented by corresponding compound transitions in stratum 'k-1' as a result of folding the organizational structure in stratum 'k' to stratum 'k-1'. The x's and -x's shown in the figures represent potential input and output connections that might exist between a transition-place pair respectively. The x's take their values from the set {0, 1}, a 1 denotes an output connection, while a -1 indicates an input connection. The subnets shown in Figure 5.22 are moved to subpages when the corresponding subnets are replaced by their compound transition representation in the attempt to fold the organizational structure. The incidence matrices representing these subnet are shown with all possible kinds of interactional places that may or may not be present in the organizational structure, therefore generic labels for stratum node numbers, and stages of a DMU are used to accommodate all possible interactional structures. Note, the subnets in Figure 5.22 and the incidence matrices of the subnets are presented in terms of only two generic DMUs 'i' and 'j' , the reason for which has already been explained. The places with an asterisk (*) represent port nodes.

	t_{0qk-1}	t_{1ik}	t_{1jk}
p_{0qk-1}^*	-x	0	0
p_{1ik}	x	-x	0
p_{1jk}	x	0	-x
p_{2ik}^*	0	x	0
p_{2jk}^*	0	0	x
p_{2ijk}^*	0	x	0
p_{2jik}^*	0	0	x
p_{6ij1k}^*	0	0	-x
p_{6ji1k}^*	0	-x	0
$p_{6q'q1k-1}^*$	-x	0	0
$p_{2\alpha\beta\gamma\lambda}^*$	0	x	x
$p_{\mu\alpha\beta1\lambda}^*$	0	-x	-x

Figure 5.23 Incidence Matrix of the Subnet Represented by t_{1qk-1}

$$
\begin{array}{c} \\ p_{2ik}^{*} \\ p_{2jk}^{*} \\ p_{3ik}^{*} \\ p_{3jk}^{*} \\ p_{2ijk}^{*} \\ p_{2jik}^{*} \\ p_{6ij2k}^{*} \\ p_{6ji2k}^{*} \\ p_{3\alpha\beta\gamma\lambda}^{*} \\ p_{\mu\alpha\beta2\lambda}^{*} \end{array}
\begin{array}{c} \begin{array}{cc} t_{2ik} & t_{2jk} \end{array} \\ \begin{bmatrix} -x & 0 \\ 0 & -x \\ x & 0 \\ 0 & x \\ 0 & -x \\ -x & 0 \\ 0 & -x \\ -x & 0 \\ x & x \\ -x & -x \end{bmatrix} \end{array}
$$

Figure 5.24 Incidence Matrix of the Subnet Represented by t_{2qk-1}

$$
\begin{array}{c} \\ p_{3ik}^{*} \\ p_{3jk}^{*} \\ p_{4ik}^{*} \\ p_{4jk}^{*} \\ p_{4\alpha\beta\gamma\lambda}^{*} \\ p_{\mu\alpha\beta3\lambda}^{*} \end{array}
\begin{array}{c} \begin{array}{cc} t_{3ik} & t_{3jk} \end{array} \\ \begin{bmatrix} -x & 0 \\ 0 & -x \\ x & 0 \\ 0 & x \\ x & x \\ -x & -x \end{bmatrix} \end{array}
$$

Figure 5.25 Incidence Matrix of the Subnet Represented by t_{3qk-1}

	t_{4ik}	t_{4jk}
p_{4ik}^*	-x	0
p_{4jk}^*	0	-x
p_{5ik}^*	x	0
p_{5jk}^*	0	x
p_{6ij4k}^*	0	-x
p_{6ji4k}^*	-x	0
$p_{5\alpha\beta\gamma\lambda}^*$	x	x
$p_{\mu\alpha\beta4\lambda}^*$	-x	-x

Figure 5.26 Incidence Matrix of the Subnet Represented by t_{4qk-1}

	t_{5ik}	t_{5jk}	t_{6qk-1}
p_{5ik}^*	-x	0	0
p_{5jk}^*	0	-x	0
p_{6ik}	x	0	-x
p_{6jk}	0	x	-x
p_{6ij1k}^*	x	0	0
p_{6ij2k}^*	x	0	0
p_{6ij4k}^*	x	0	0
p_{6ji1k}^*	0	x	0
p_{6ji2k}^*	0	x	0
p_{6ji4k}^*	0	x	0
$p_{6qq'1k-1}^*$	0	0	x
$p_{6\alpha\beta\gamma\lambda}^*$	x	x	0
$p_{\mu\alpha\beta5\lambda}^*$	-x	-x	0

Figure 5.27 Incidence Matrix of the Subnet Represented by t_{5qk-1}

5.7.4 Unfolding the Organizational Structure

Unfolding is the process in which an organizational structure in a particular stratum is decomposed into its subsystems and their mutual interactions defined in lower strata. The process yields a more elaborate and detailed description of the organization under study. In this process, the compound transitions are replaced by the subnets representing these compound transition in a lower stratum. This process of uncompounding the compound transitions continues till the desired degree of abstraction used to describe the system is achieved. As a result of uncompounding the compound transitions, the equivalent places in a stratum 'k' representing the places of the subnets in stratum 'k+1' are automatically replaced by the places whose equivalence they are depicting. The port nodes defined in the subnets replacing compound transitions are used to connect all the subnets replaced. The subnets have already been presented in Figure 2.22. The incidence matrix of an unfolded organizational structure is constructed by replacing compound transitions by the incidence matrices of the subnets representing these transitions in the next lower stratum, Figures 5.23 to 5.27. Once all the incidence matrices of the subnets are put together, they are merged into a single incidence matrix representing the organization in stratum 'k+1' by joining all those rows of the individual incidence matrices which have the same place labels and, as a result, constructing a single row in the incidence matrix. All other unspecified elements of the incidence matrix are set to zero since these elements represent the interactions that are either not permissible or not defined in the organizational structure.

Once an organizational structure in stratum 'k' is unfolded to stratum 'k+1', all the interactional links present in the stratum 'k' description of the organization need to be translated to their lower stratum representation in case these lower stratum connections are not specified. A number of connectivity rules are needed to be formulated in order to resolve this issue. Chapter VII deals with this problem and proposes a number of connectivity rules to be implemented in order to translate a higher stratum interactional link to its lower stratum representation. At this point, the only assumption that is made regarding this connectivity issue is as follows. *An interactional link whose input and output connectivity is defined in stratum 'k' is translated into a single link at stratum 'k+1' between the subsystems of the input and output compound transitions having this interactional link in stratum 'k'.* The issue of connectivity is further elaborated by examining all possible interactions one at a time and observing the possibilities that might occur in an attempt to translate the interaction to its lower stratum representation.

- Figure 5.28(a) shows an interactional link between the SAC stage of DMU_{ik} to IFC stage of DMU_{jk}, while Figure 5.28(b) presents all the possible input and output connections that might represent the single interactional link in terms of the subsystems in stratum 'k+1' of compound nodes 'i' and 'j'. Each 'x' in the figure represents a potential output connection, while a '-x' indicates a potential input connection. Only one 'x' and a '-x' should get a value of '1' and '-1' according to the assumption mentioned above. Figure 5.29 presents the Petri Net representation of the problem discussed Note that the variables used in the labels of the subsystems of compound nodes 'i' and 'j' refer to the scheme presented in Figure 5.6.

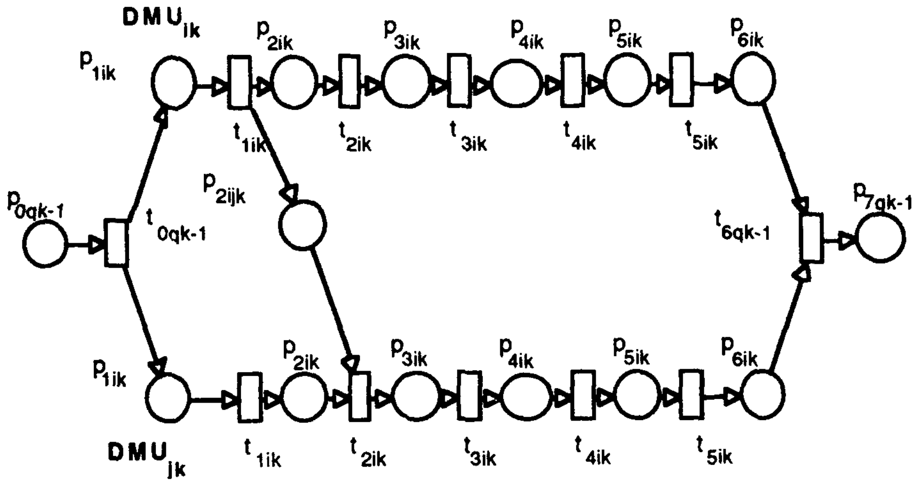
$$\begin{array}{cc} & \begin{array}{cc} t_{1ik} & t_{2jk} \end{array} \\ p_{2ijk} & \begin{bmatrix} 1 & -1 \end{bmatrix} \end{array}$$

(a)

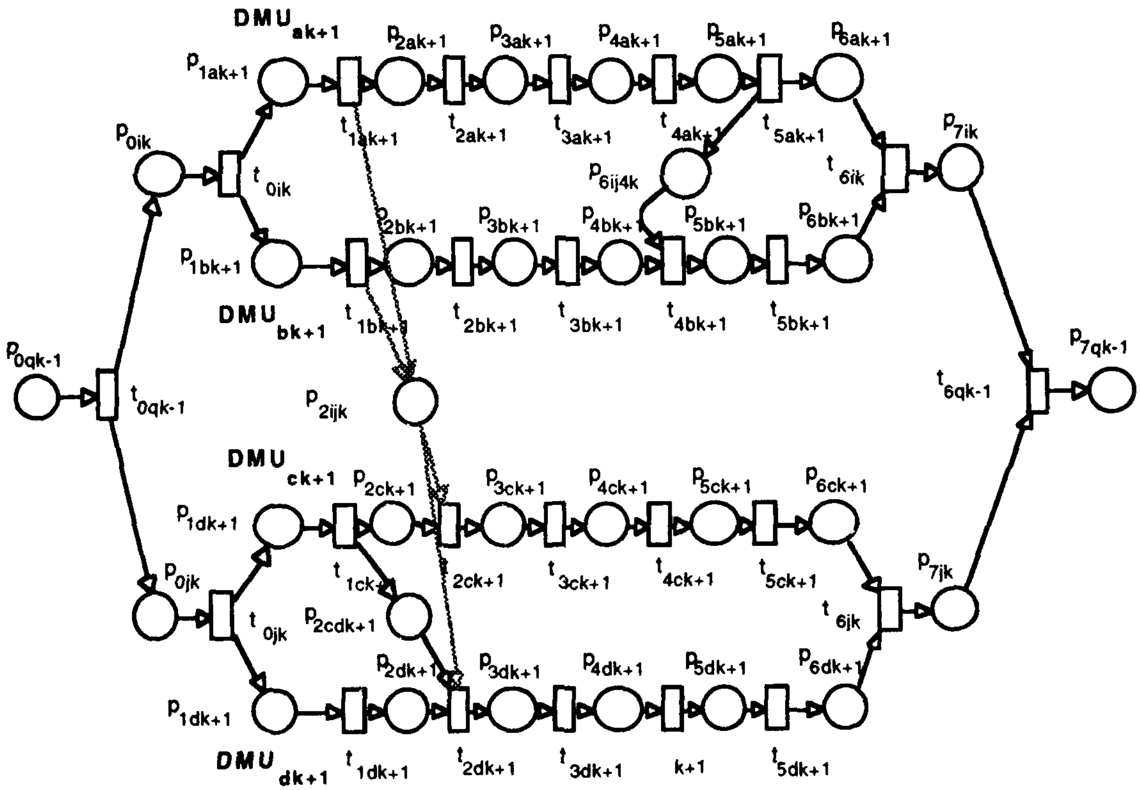
$$\begin{array}{cc} & \begin{array}{cccc} t_{1ak+1} & t_{1bk+1} & t_{2ck+1} & t_{2dk+1} \end{array} \\ p_{2ijk} & \begin{bmatrix} x & x & -x & -x \end{bmatrix} \end{array}$$

(b)

Figure 5.28 Strata 'k' and 'k+1' Representation of p_{2ijk}



(a) Stratum 'k' Representation



(b) Stratum 'k+1' Representation

Figure 5.29 Petri Net Representation of Figure 5.28

- Figure 5.30(a) shows an interactional link between the RSC stage of DMU_{ik} to SAC stage of DMU_{jk}, while Figure 5.30(b) presents the input and output connections that represent the single interactional link in terms of the subsystems in stratum 'k+1' of compound nodes 'i' and 'j'. This case has already been discussed in Section 4.5 and is different from the one discussed above. It is evident that for these kind of interactions no connectivity rules are needed. Figure 5.31 shows the Petri Net representation of the situation depicted in Figure 5.30.

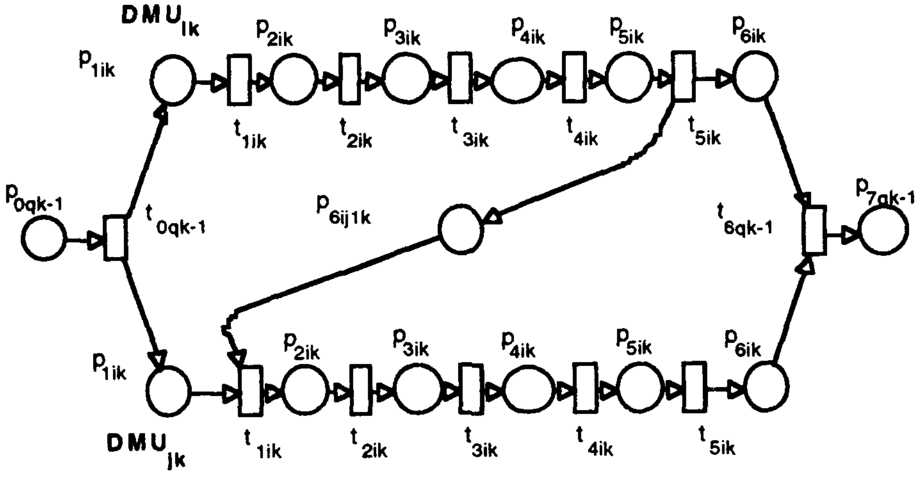
$$\begin{array}{c|cc} & t_{5ik} & t_{1jk} \\ \hline p_{6ij1k} & 1 & -1 \end{array}$$

(a)

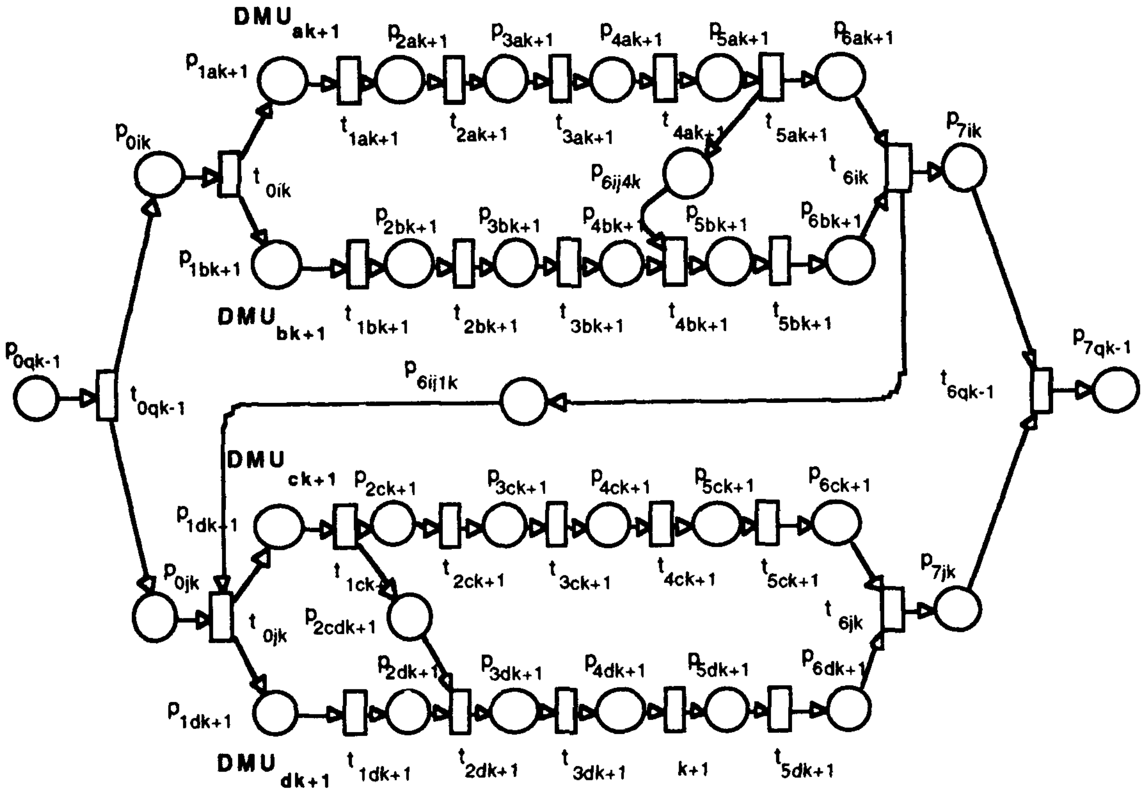
$$\begin{array}{c|cc} & t_{6ik} & t_{0jk} \\ \hline p_{6ij1k} & 1 & -1 \end{array}$$

(b)

Figure 5.30 Strata 'k' and 'k+1' Representation of p_{6ij1k}



(a) Stratum 'k' Representation



(b) Stratum 'k+1' Representation

Figure 5.31 Petri Net Representation of Figure 5.30

- Figure 5.32(a) shows an interactional link between the RSC stage of DMU_{ik} to IFC stage of DMU_{jk}, while Figure 5.32(b) presents all the possible input and output connections that might represent the single interactional link in terms of the subsystems in stratum 'k+1' of compound nodes 'i' and 'j'. Figure 5.33 presents the Petri Net representation of the problem discussed.

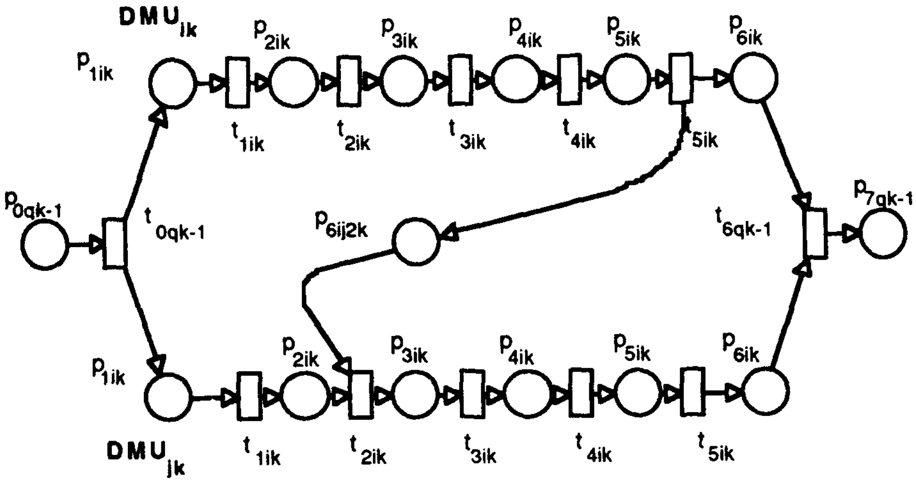
$$\begin{array}{c} \\ p_{6ij2k} \end{array} \begin{array}{c} \begin{array}{cc} t_{5ik} & t_{2jk} \end{array} \\ \begin{bmatrix} 1 & -1 \end{bmatrix} \end{array}$$

(a)

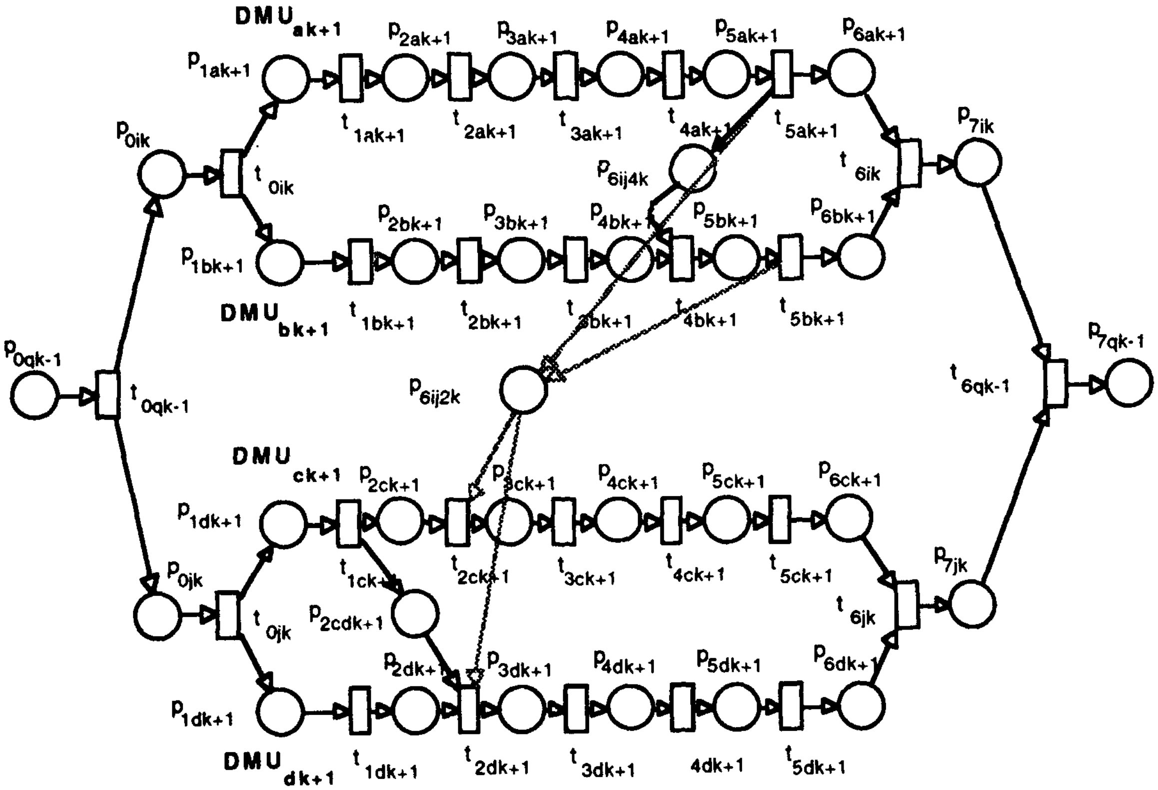
$$\begin{array}{c} \\ p_{6ij2k} \end{array} \begin{array}{c} \begin{array}{cccc} t_{5ak+1} & t_{5bk+1} & t_{2ck+1} & t_{2dk+1} \end{array} \\ \begin{bmatrix} x & x & -x & -x \end{bmatrix} \end{array}$$

(b)

Figure 5.32 Strata 'k' and 'k+1' Representation of p_{6ij2k}



(a) Stratum 'k' Representation



(b) Stratum 'k+1' Representation

Figure 5.33 Petri Net Representation of Figure 5.32

- Figure 5.34(a) shows an interactional link between the RSC stage of DMU_{ik} to CIC stage of DMU_{jk}, while Figure 5.34(b) presents all the possible input and output connections that might represent the single interactional link in terms of the subsystems in stratum 'k+1' of compound nodes 'i' and 'j'. Figure 5.35 presents the Petri Net representation of the problem.

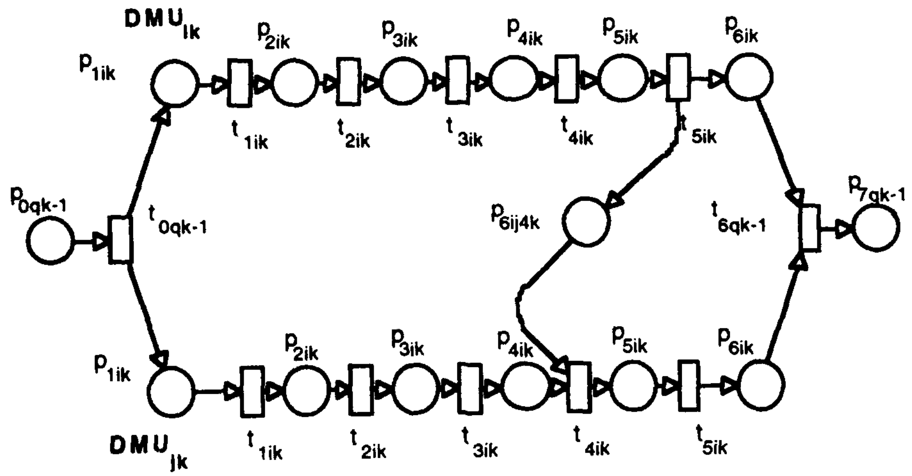
$$\begin{array}{c} \\ p_{6ij4k} \end{array} \begin{array}{cc} t_{5ik} & t_{4jk} \\ \lbrack\; 1 & -1 \;\rbrack \end{array}$$

(a)

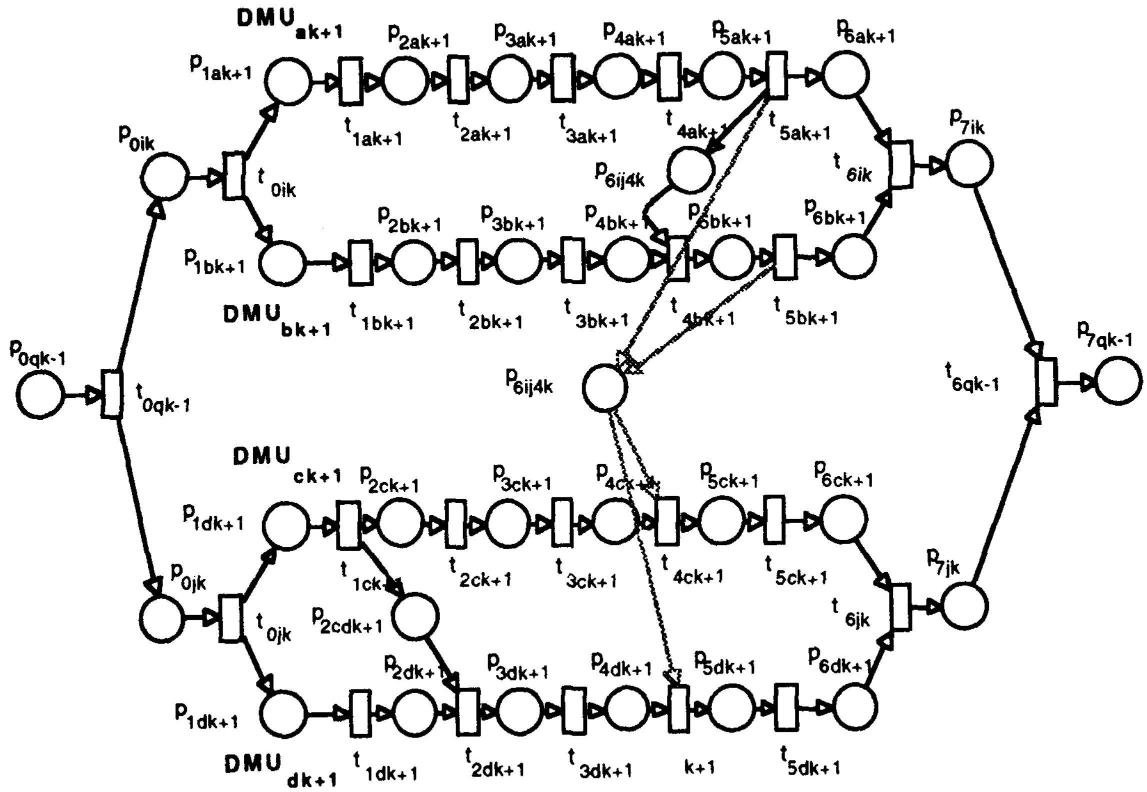
$$\begin{array}{c} \\ p_{6ij4k} \end{array} \begin{array}{cccc} t_{5ak+1} & t_{5bk+1} & t_{4ck+1} & t_{4dk+1} \\ \lbrack\; x & x & -x & -x \;\rbrack \end{array}$$

(b)

Figure 5.34 Strata 'k' and 'k+1' Representation of p_{6ij4k}



(a) Stratum 'k' Representation



(b) Stratum 'k+1' Representation

Figure 5.35 Petri Net Representation of Figure 5.34

- Finally, a generic label is used for the places representing an interactional link in stratum 'λ', $k \leq \lambda < n$, among DMUs 'α' and 'β' with the input and output stages of the places denoted by 'μ' and 'γ', $1 \leq \mu, \gamma \leq 5$. Figure 5.36(a) shows the situation where an interactional place defined in stratum 'λ' is translated into an interactional link between the corresponding stages of DMUs 'i' and 'j' in stratum 'k'. Figure 5.36(b) presents the situation where the said interactional link is translated in terms of the subsystems of DMUs 'i' and 'j' which are defined in stratum 'k+1'. The generic case shown in Figure 5.36 also accounts for the interactional links that are either defined at a stratum higher than stratum 'k' or implemented by the special constraints R_p.

$$\begin{array}{cc} & t_{\mu ik} \quad t_{\gamma jk} \\ p_{\mu\alpha\beta\gamma\lambda} & \begin{bmatrix} 1 & -1 \end{bmatrix} \end{array}$$

(a)

$$\begin{array}{cc} & t_{\mu ak+1} \quad t_{\mu bk+1} \quad t_{\gamma ck+1} \quad t_{\gamma dk+1} \\ p_{\mu\alpha\beta\gamma\lambda} & \begin{bmatrix} x & x & -x & -x \end{bmatrix} \end{array}$$

(b)

Figure 5.36 Stratum 'k' and 'k+1' Representation of $p_{\mu\alpha\beta\gamma\lambda}$

After unfolding an organization structure to the next lower stratum, the Petri Net representation is checked for any internal sinks or sources. All the internal sinks and sources, if they are found, are then deleted. In the incidence matrix representation the check is performed by searching all rows of the matrix, except for the first and the last ones, for any row with a single non-zero element.

Appendix B describes implementation of the folding and unfolding procedures on a software package called *Design/CPN™*.

5.7.5 Example 5.3

Figure 5.37 gives a block description of a 2-strata Stratified Decision Making Organization (SDMO) with two nodes '1' and '2' in stratum '1', where each of these two nodes has its subsystems defined in stratum '2' - the lowest stratum. The organizational structures of the compound nodes '1' and '2' are taken from Examples 5.1 and 5.2.

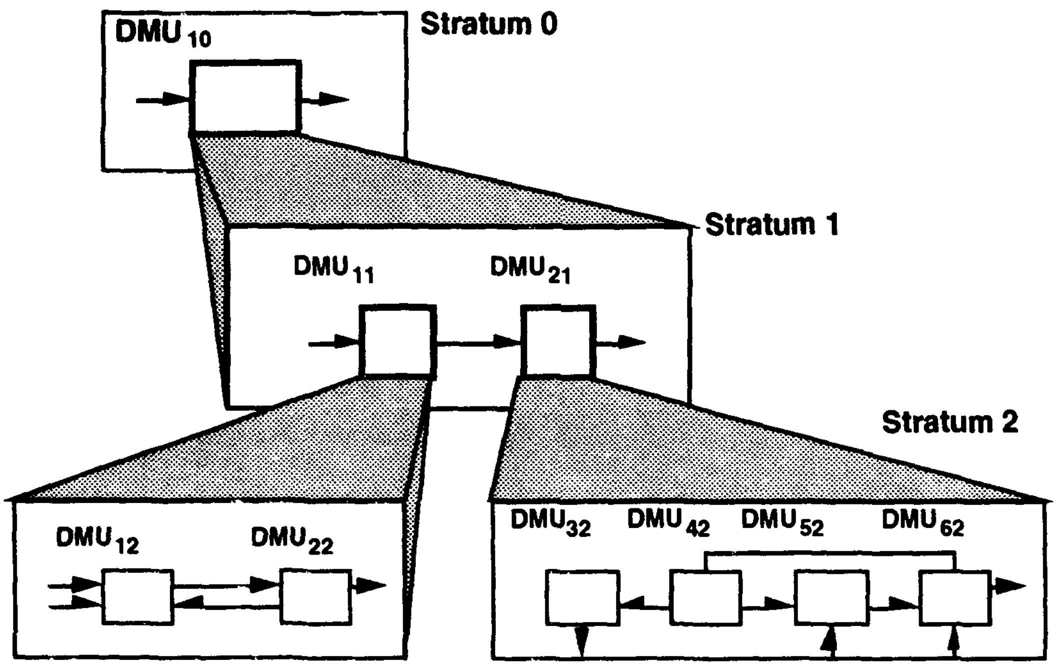


Figure 5.37 2-Strata SDMO For Example 5.3

Figure 5.38 gives the matrix representation of Σ_{11}, a 2-dimensional WDN of the compound node '1' in stratum '0' with two subsystems '1' and '2' and their interactions defined in stratum '1'. Figure 5.40 shows the corresponding Petri Net representation of Σ_{11}. Finally, Figure 5.39 presents the Petri Net representation of the SDMO in stratum '2' with the nodes '1' and '2' in stratum '1' unfolded to stratum '2'. The dotted arcs in Figure 5.40 show the possible connections that can exist as a result of the translation of a higher stratum interactional link to its lower stratum representation. The connectivity rules, presented in Chapter VII, are applied to choose a pair of input and output dotted arcs to represent a fixed link at the lower stratum.

$$\Sigma_{11} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$$

$$e = [\,1 \quad 1\,] \qquad F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$s = [\,1 \quad 1\,] \qquad H = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Figure 5.38 Matrix Representation of Σ_{11}

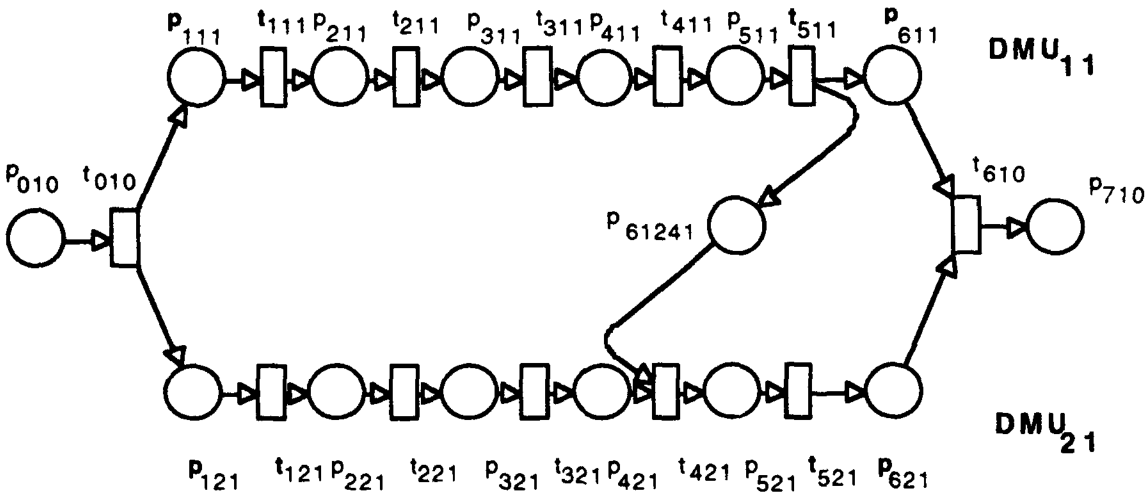


Figure 5.39 Petri Net Representation of Σ_{11}

The incidence matrix Δ_{102} of node '1', in stratum '0', when unfolded to stratum '2' is given in Figure 5.41. In Figure 5.41, the x's represent the possible connections shown by dashed arcs in Figure 5.40.

5.7.6 Equivalence Between the Representations of a WDN

A WDN can be represented in three different ways, i.e., the matrix representation as presented in Section 4.6, the Petri Net representation given by the incidence matrix of the net and the Petri Net representation given by the labeling of the places.

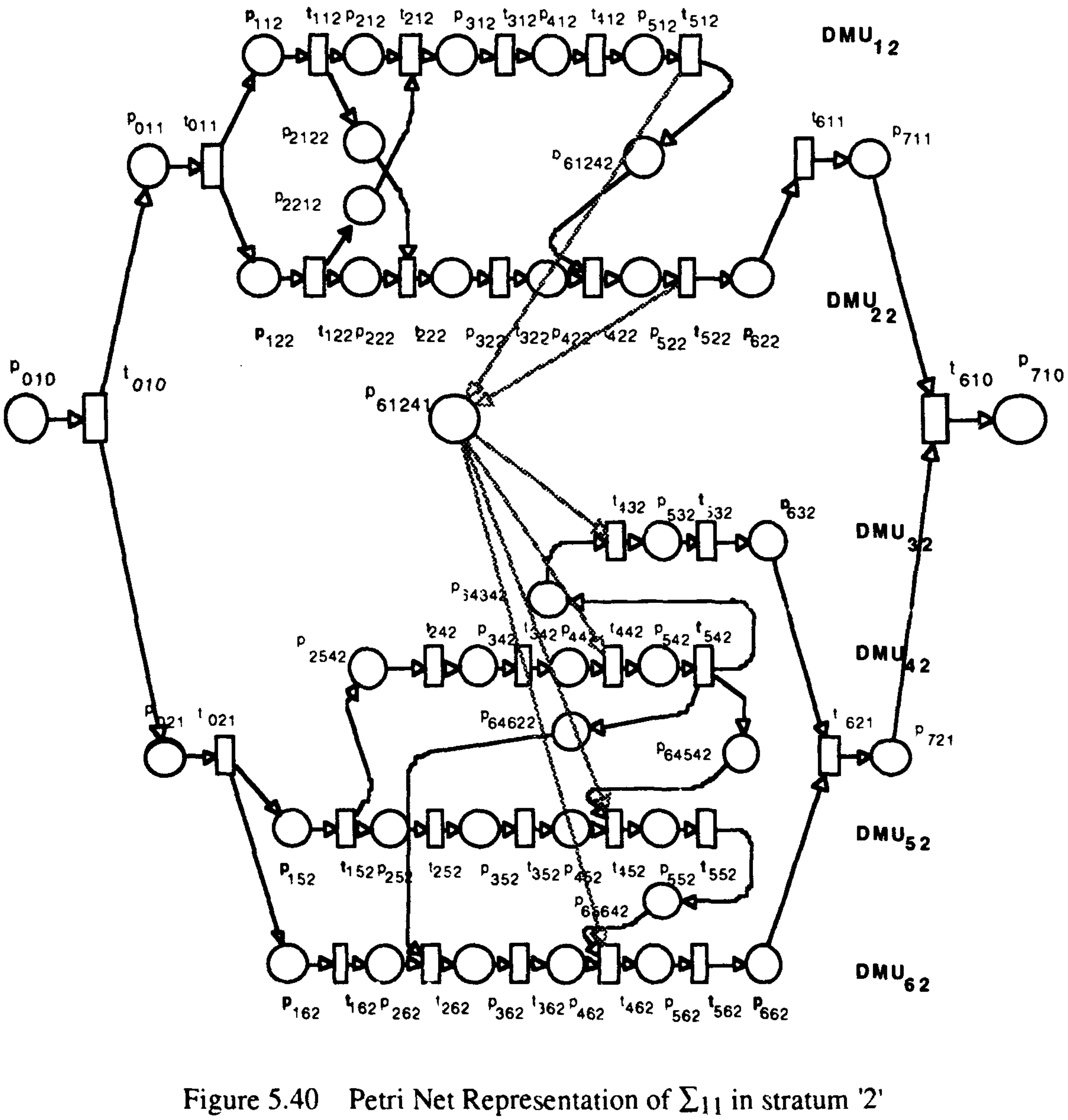


Figure 5.40 Petri Net Representation of Σ_{11} in stratum '2'

The three different representation, however, are equivalent, i.e., a one-to-one correspondence exists between any two of them. For proof of the previous statement, see Remy (1986).

5.8 GENERIC PROPERTIES OF WDNs

The following proposition by Remy (1986) characterizes a class of Ordinary Petri Nets that model the elements of W.

$$\Delta_{102} = \left[\begin{array}{cccccccccccccccccccccccccccccccc}
-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-x&-x&-x&-x&x&x&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0\\
0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&-1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right]$$

Figure 5.41 Incidence Matrix Δ_{102} of node '1' in stratum '0' Unfolded in Stratum '2'

Proposition 5.1

Let the source and sink places of the Ordinary Petri Net representing a WDN Σ_{ik+1} in stratum 'k+1' of a node 'i' stratum 'k' be combined into a unique place, *the external place*. If the resulting Petri Net is strongly connected, it is a marked graph.

The proof is straightforward. Each internal or interactional place has exactly one input and one output transition. The sink has one input transition but no output transitions, while the opposite holds for the external place. If the external place and the sink are merged together into one single place, every place in the net will therefore have one input and one output transition. Since the net is strongly connected, every transition has at least one input place and one output place, and the net is a marked graph.

5.9 CONCLUSION

A *modified labeling scheme* was presented to label the places and transitions of a Stratified Decision Making Organization (SDMO). The labeling technique is primarily introduced for computational purposes but it also gives an algorithmic approach to fold and unfold organizational structures from one stratum to another. An upper bound to the maximum number of places and transition in an organizational structure defined in an arbitrary stratum was presented. A one-to-one correspondence was established between the Petri Net and matrix representations of an organizational structure. The construction of the incidence matrix of a Petri Net representation of an organizational structure was presented. The incidence matrix is the basis of a number of algebraic computations that are made on Ordinary Petri Net structures. The folding and unfolding of organizational structures was presented to describe system's architectures at different levels of abstraction. The folding and unfolding processes were applied to both Petri Net and incidence matrix representations. Finally, an example was presented to illustrate the folding and unfolding processes.

CHAPTER VI

CONSTRAINTS

Well defined fixed structures in multilevel organizational classes have been described in Chapter IV. The mathematical and Petri Net representations of the Well Defined Nets (WDNs) have been presented in Chapter V. In this chapter, the constraints that must be verified by WDNs defined at a particular stratum are described.

6.1 INTRODUCTION

A number of *structural* and *user-defined* constraints have already been introduced by Remy (1986) and Demaël (1989). The existing set of constraints fulfills the requirement of an organizational form when defined at the lowest stratum 'N', with DMUs as human decision makers (DMs). The introduction of the stratified organizational forms and the concept of compound node leads to the definition of an extended set of constraints that must be satisfied by the organizational structures defined at stratum 'k', where $1 \leq k \leq n$ ($n \leq N$). For illustration purposes, two different set of constraints are presented. These two sets of constraints are defined as follows:

(i) *Global Constraints* : The set of constraints that must be satisfied by all the organizational forms regardless of the stratum for which they are defined.

(ii) *Compound Node Constraints* : The set of constraints that are defined only for the organizational forms which have compound nodes as DMUs.

These constraints not only eliminate the WDNs that do not represent realistic organizational forms, but also reduce the dimensionality of the design problem. The introduction of the constraints proceeds from two different considerations.

(a) Some WDNs correspond to patterns of interactions among DMUs that do not make any physical sense, might introduce deadlocks, cause circulation of messages in the organization, and result in partially connected organizational forms. These WDNs should be eliminated, if realistic organizational forms are to be generated. No designer wants to obtain these structures as candidate architectures for the system under study. There is thus a need to define *structural constraints*, which rule out the types of WDNs that have the mentioned problems.

(b) Any practical design procedure should provide the designer with only those candidate structures that satisfy the structural specifications of the system. The designer must thus be given the possibility of translating his knowledge of the system's structural specifications into mathematical terms by imposing *user-defined constraints.*

6.2 STRUCTURAL CONSTRAINTS

Structural constraints on the set of fixed structures have been defined in Remy (1986) using a model of a single-stratum organization. In the sequel, these constraints are adapted to the problem of this thesis. As mentioned earlier, the set of constraints are classified as global and compound node constraints. The structural constraints defined in Remy (1986) map directly to the set of global constraints.

6.2.1 Global Constraints

Let Σ_{qk} be an organizational form in stratum 'k' defined for node 'q' in stratum 'k-1'. Then the fixed structure associated with it must satisfy

• (R1)

(a) The Ordinary Petri Net that corresponds to Σ_{qk} 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 source place to every node of the PN and from every node to the sink.

- (R2) The Ordinary Petri Net that corresponds to Σ_{qk} should have no loops. i.e., the structure must be acyclic.
- (R3) In the Ordinary Petri Net that corresponds to Σ_{qk}, there can be at most one link from the RS/RSC stage of a DMU 'i' to another DMU 'j', i.e., for each 'i' and 'j', only one element of the triplet $\{ G_{ij}, H_{ij}, C_{ij} \}$ can be non-zero.

- (R4) Information fusion can take place only at the IF/IFC and CI/CIC stages. Consequently, the SA/SAC stage of a DMU can either receive information from the external environment, or a control signal from another DMU.

Constraint R1(a) eliminates any organizational structure that does not represent a single structure. Constraint R1(b) insures that the flow of information is continuous within the organizational structure. It eliminates internal sink or source places. For the kind of organizational structures modeled in this thesis, R1(b) implies R1(a).

Constraint R2 allows acyclical organizational structures only. This restriction is imposed to avoid deadlocks and infinite circulation of messages within the organization (Levis, 1984). Note, however, that constraint R2 does not imply that the graphical representation of the stratified organizational forms is acyclical, because the folding of acyclical nets can yield a structure with loops. The constraint of acyclicity is restricted to the elements of the set of WDN W_{qk} of a node 'q' in stratum 'k-1' defined in stratum 'k'.

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 DMU, this output is stored in its internal memory and can be accessed at later stages.

Constraint R4 has to do with the nature of the IF/IFC stage. The IF/IFC stage has been introduced explicitly to perform a fusion between the situation assessments performed by other DMUs. It prevents a DMU from receiving more than one input at the SA/SAC stage (Balbes and Dwinger, 1974). Subsection 6.3.3 presents a way of circumventing this restriction without increasing the dimensionality of the design problem.

6.2.2 Compound Node Constraints

Let Σ_{qk} be the organizational form in stratum 'k' defined for node 'q' in stratum 'k-1' with DMUs 'i' and 'j' being the compound nodes. Then the fixed structure associated with Σ_{pk}, in addition to the global constraints, must also satisfy the following compound node constraints.

- (C1) In the Ordinary Petri Net that corresponds to Σ_{qk}, there must be an *input link* to the SAC stage of a DMU 'i'. This input link can be an external input or a control signal from another DMU 'j'.

- (C2) In the Ordinary Petri Net that corresponds to Σ_{qk}, there must be at least one *output link* from the RSC stage of a DMU 'i'. This output link can be an external output or control signal to another DMU 'j', or both.

Constraint C1 insures an input connection to a compound node DMU. As mentioned earlier, a compound node has all of its five stages present in an organizational structure with the compound node as a DMU. The constraint insures the presence of the SAC stage of a compound node.

Constraint C2 insures an output connection to a compound node DMU. The constraint realizes the presence of the RSC stage of a compound node. Once the SAC and RSC stages are present, all the intermediate stages must also be present, thus satisfying the condition that all the stages should appear in a compound node structure.

The application of constraint R1 on organizational forms with compound nodes as DMUs implies constraints C1 and C2.

6.3 USER-DEFINED CONSTRAINTS

As mentioned in the Introduction, a design procedure should allow the designer of an organization to introduce constraints that reflect specific structural considerations. He may rule in or rule out some links, force a certain pattern of interaction, or express hierarchical echelon type relationship between the DMUs.

These restrictions and specification will be denoted as user-defined constraints. They can be introduced in two different ways.

6.3.1 Constraints R_f

The designer can place appropriate 0's and 1's in the arrays { **e**, **s**, **F**, **G**, **H**, **C** } defining the WDN.

6.3.2 Constraints R_p

To accommodate some very special kind of interactions not covered by the arrays mentioned above, the designer of an organization is allowed to introduce *special constraints*, R_p. The links introduced as special constraints may be the ones that are not covered by the allowable interactions presented in Figures 4.3 and 4.8. The links, however, are fixed and therefore do not increase the dimensionality of the design problem, rather they introduce some flexibility in the design procedure. The rationale behind the introduction of special constraints is given in Remy (1986).

6.3.3 Conflict Among Constraints

In general, no conflict is allowed between the structural and user-defined constraints. There is a single exception, however, to this generic rule. This exception has been introduced by Remy (1986) to alleviate the restriction imposed by constraint R4. The exception is adapted to the context of this thesis as it also solves a number of design problems introduced by the restrictions imposed by R4. The exception is the following: R4 will not apply to the special constraints. Consequently, a DMU can have more than one input at his SA/SAC stage, provided that all those inputs but at most one be special constraints.

6.4 MATHEMATICAL REPRESENTATION OF THE CONSTRAINTS

Definition 6.1: *External Place*

If the source place p_{0qk-1} and the sink place p_{7qk-1} of a WDN Σ_{qk} of a node 'q' in stratum 'k-1' are merged into a single place, the resulting place is termed as the external place.

The constraints on WDNs can be translated into formal ones as follows:

Constraint R1

Let the source place and the sink place of the WDN Σ_{qk} be merged into the external place, R1(a) and R1(b) can now be formulated as
The Petri Net representing Σ_{qk} should be strongly connected

Corollary

Proposition 5.1 by Remy (1986) indicates that any element of W_{qk} that satisfies R1 and whose sink and source places are merged into the external place is a marked graph.

Constraint R2

Once the constraint R1 is satisfied and the external place is defined, Constraint R2 becomes

All simple information flow paths of the Petri Net contain the external place where simple information flow paths have been defined in Chapter 2.

Constraint R3

The analytical expression of this constraint is given as:

$$\forall\ (i, j) \in [1..|\mu_k|]^2 \qquad G_{ij} + H_{ij} + C_{ij} \leq 1 \qquad i \neq j \tag{6.1}$$

Constraint R4

The translation of this constraint into mathematical terms follows:

$$\forall\ j \in [1..|\mu_k|] \qquad e_j + \sum_{i=1}^{m} G_{ij} \leq 1 \tag{6.2}$$

Constraint C1

The constraint is trivially expressed as:

$$\forall\ j \in [1..|\mu_k|] \qquad e_j + \sum_{i=1}^{m} G_{ij} = 1 \tag{6.3}$$

Constraint C2

Like C1, C2 is translated easily into

$$\forall\ j \in [1..|\mu_k|] \qquad s_j + \sum_{i=1}^{m} G_{ji} \geq 1 \tag{6.4}$$

Constraint R_f

As mentioned earlier, the constraints R_f are defined by assigning 0 or 1 to elements of the arrays **e**, **s**, **F**, **G**, **H**, and **C**.

Constraint R_p

An R_p constraint is characterized by an interactional link not covered by the arrays mentioned above or a link that can be represented in a WDN but the introduction of the link violates one of the constraints listed in Section 6.3. It is designated by its input and output transition pair, (t_{sik}, t_{rjk}). t_{sik} is the input transition, while t_{rjk} is the output transition - 's' and 'r' represent the stages, while 'i' and 'j' represent the DMUs. The following restrictions apply to the set of R_p:

- $i \neq j$: the two DMUs should be different.
- All those links that can be represented in a WDN should not appear in R_p except for the case where
  - $s = 5$ and $r = 1$: a link between RS/RSC and SA/SAC stages.
  - If $s = 5$ and $r = 1, 2, 4$; provided that the introduction of these links in WDN violates constraints R3.

In the Petri Net representation, each special constraint will be represented by an interactional place. The labeling of the place will be determined by its input and output transitions as:

(t_{sik}, t_{rjk}) will correspond to $p_{s+1ijrk}$.

6.5 TERMINOLOGY

6.5.1 Set of Constraints, R

The set of structural constraints is denoted as R_S, while the set of user-defined constraints is represented by R_u, which in turn is given as:

$$R_u = R_f \cup R_p$$

$$R = R_u \cup R_s$$

where

$$R_s = R1 \cup R2 \cup R3 \cup R4 \cup C1 \cup C2 \qquad (6.5)$$

The binary operator '$\cup$' is defined as the *union* of its operands. Therefore, the set of structural constraints R_S is described as the union of all the constraints defined in Section 6.3, while the set R is shown as the union of sets R_u and R_S.

6.5.2 Well Defined Structure, WDS

A Well Defined Structure is defined to be a WDN at stratum 'k+1' of a node 'i' at stratum 'k' that fulfills the special constraints R_p:

$$WDS = (WDN, R_p)$$

In order to avoid cumbersome notation, node and stratum variables are not added to the labels used to denote WDSs.

Since the special constraints are taken as fixed constraints throughout the design procedure, the WDS is trivially defined as:

$$WDS = (WDN, R_p) \rightarrow WDN$$

6.5.3 Admissible Organizational Form

A WDS that fulfills the set of user-defined constraints R_u has been defined as an Admissible Organization Form (AOF). The set of all AOFs will be denoted as $W(R_u)_{ik+1}$.

6.5.4 Feasible Organization

An AOF that fulfills the set of constraints R is called a Feasible Organization (FO). It is defined to be a WDN that fulfills the complete set of constraint, R. It is denoted as $W(R)_{ik+1}$.

If the set of special constraints and set of user-defined constraints R_f are given and are not empty sets, the following inclusion holds. Since the introduction of a constraint either rules in or rules out certain links, the restriction imposed by the constraint excludes a a number of WDNs from the set of WDNs satisfying the constraint.

$$W_{ik+1} \supset W(R_u)_{ik+1} \supset W(R)_{ik+1} \tag{6.6}$$

6.6 CONVEXITY OF THE CONSTRAINTS

6.6.1 Convexity

Definition 6.1: *Interval*

If a and b are elements of a partially ordered set A satisfying $a \leq b$, then an interval $[a, b]$ is defined to be $\{x \in A \mid a \leq x \leq b\}$.

Definition 6.2: *Convex Subset*

A subset A1 of the partially ordered set A is convex if and only if the following implication holds:

$$(\forall\ (a1,b1) \in A1^2)\ (a1 \leq b1) \Rightarrow (A1 \supset [a1, b1]) \tag{6.7}$$

Definition 6.3: *Convexity of the Property*

A property S defined on A is convex if and only if every element x of A located in the interval [a, b], where a and b satisfy S, also satisfies S.

Proposition 6.1 (Remy)

If a property S is convex on A, a convex subset A1 of A that satisfies S is completely characterized by its minimal and maximal elements as:

$$A1 = \{x \in A1 \mid \exists (a1, b1) \in A1_{min} \times A1_{max} \quad a1 \leq x \leq b1\} \tag{6.8}$$

If a set is convex, its structure can be assessed with three simple tools, a partial ordering, a set of minimal elements and a set of maximal elements. Any element that is below one maximal element and above one minimal element belongs to the set. There is no need for an extensive, and possibly combinatorial, description of all the elements. Finding convex subsets in the set of WDS is quite important since convexity allows the description of the subsets without resorting to a combinatorial computational problem. In that case, the set of solutions can be obtained in terms of the minimal and maximal elements of the set.

6.6.2 Convexity of the Constraints

This section applies the results presented in the previous subsection to characterize the constraints. The constraints R are properties on the set W_{ik+1} since a constraint is either satisfied or violated by a given WDS. We can therefore apply the concept of convexity defined in the previous subsection to the different constraints in R.

The advantage of having convex constraints is obvious since in that case the set $W(R)_{ik+1}$ can be characterized by its minimal and maximal elements. The minimal and maximal elements of set $W(R)_{ik+1}$ are defined in Chapter VIII.

Proposition 6.2 (Remy)

The constraints R2, R3, R4 defined on the set W_{ik+1} are convex.

Proof

The proof of the proposition is given in Demaël (1989) and is very direct. Let us consider R2. If a WDS is acyclical, i.e., fulfills R2, then any WDS obtained by removing links from the initial WDS will also be acyclical. Loops can not be created in a loop-free structure by removing links. The same argument applies to the constraints R3 and R4. For a detailed proof of the Proposition for constraints R3 and R4 see Demaël (1989).

Proposition 6.3 (Remy)

Constraint R1 defined on the set W_{ik+1} is not convex.

Proof

The constraint R1 is not convex as it is possible to break the connectivity of a fixed structure by removing a link as well as by adding a link. This happens, for example, if a link that is added to the structure originates from a transition of the current net but does not terminate at a transition that was previously in the net. In that case, a transition without output place is created, which violates R1. Figure 6.1 describes a sequence in which R1 is fulfilled, violated, and fulfilled again by successively adding two links.

Proposition 6.4

Constraints C1, C2 are convex

Proof:

The restrictions imposed by C1 and C2 are realized by placing 1s at the appropriate places in the arrays **e**, **s**, and **G** in order to ensure that a compound node structure has both input and output links. Let Σ'_{ik+1} and Σ''_{ik+1} be two elements of the set W_{ik+1} satisfying constraints C1 and C2, then a WDN Σ_{ik+1} located in the interval $[\Sigma'_{ik+1}, \Sigma''_{ik+1}]$ will also satisfy these constraints since the addition or removal of all other links except the ones placed by C1 and C2 do not have any effect on these constraints.

Proposition 6.5

Constraints R_u are convex

Proof: The specifications defined by R_u are realized by placing 1s and/or 0s at the appropriate places in the arrays **e**, **s**, **F**, **G**, **H**, and **C** in order to rule in and/or rule out certain interactional links between DMUs. Let Σ'_{ik+1} and Σ''_{ik+1} be two elements of the set W_{ik+1} satisfying constraints R_u, then a WDN Σ_{ik+1} located in the interval $[\Sigma'_{ik+1}, \Sigma''_{ik+1}]$ will also satisfy these constraints since the addition or removal of all other links except the ones placed by C1 and C2 do not have any effect on these constraints.

The set $W(R_u)$, therefore, can be characterized by its minimal and maximal elements.

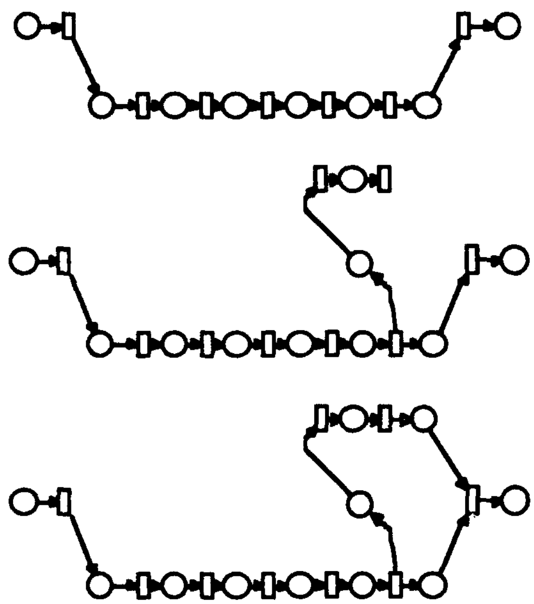


Figure 6.1 Successive Structures

6.7 CONCLUSION

The constraints that must be satisfied by the WDNs were presented in this chapter. The constraints were defined as global and compound constraints. It was shown that the constraints applied to organizational structures with human decision makers as DMUs (global constraints) are not all convex. The problem posed by constraint R1 has been solved by Remy (1986) by using the concept of *simple paths*. The solution to the design problem will be presented in Chapter VIII. Fortunately, the set of structural constraints for compound node organizations are all convex, as the introduction of constraints C1 and C2 implies R1, and both have been proved convex in Proposition 6.4. The set of FOs can, therefore, be characterized easily by its minimal and maximal elements for the organizational structures comprised of compound nodes.

CHAPTER VII

SOLUTION TO THE DESIGN PROBLEM

This chapter addresses the problem of connectivity. The proposed methodology resolves the issue of the lower-strata connectivity of an interactional link defined at a higher stratum. This issue has been raised a number of times in Chapter IV and Chapter V. In Section 7.1, the messages flowing in an organizational structure are classified in three different categories. Section 7.2 discusses the Petri Net representation of the different classes of messages in an organizational structure. Section 7.3 characterizes the hierarchical echelon type relationship among the DMUs of an organizational structure defined in a particular stratum on the basis of the classification presented in Sections 7.1 and 7.2. An analytical model of the echelon hierarchies is given in Section 7.4. Finally, Section 7.5 presents some rules that are defined on the basis of the hierarchical relationship presented in Section 7.4 to resolve the lower-strata connectivity problem of higher-strata links. An example is worked out in Section 7.5.

7.1 TYPES OF DATA

The messages that flow in a multilevel organizational structure are classified in the following categories and subcategories according to their contents :

- *Information, INF*
    - *Input/Output*
    - *Assessment*
    - *Response*

- *Control Signal, CTR*

- *Command, CMD*

As mentioned above, messages conveying information (INF) are further divided into three subcategories, *inputs/outputs, assessments*, and *responses*. Inputs represent observations from the external environment (sensors) or the external inputs defined in Subsection 4.5.2. Assessments are defined to be the outputs of the situation assessment stage of a DMU. The messages containing information about the response of a DMU are taken as responses.

The control signal is defined to be a signal which, in addition to a limited amount of information about the task, contains an enabling signal for the initiation of a subtask.

If the response or course of action selected by a DMU is dependent upon the message sent by another DMU, then such a message is termed a command or order.

7.2 PETRI NET REPRESENTATION

The interactional links presented in Chapter IV can be classified according to the different types of messages they carry. The designer of an organizational structure determines the contents of a signal flowing in the structure; the definition of the type of messages is application dependent. The interactional links presented in Chapter IV have been given a generalized physical interpretation in Subsection 4.5.2. Figures 7.1 and 7.2 show a possible interpretation of the interactional links in terms of the different types of messages introduced in the previous section. The figures map the classes of messages presented in the previous section to their Petri Net representation in view of the physical interpretation of the interactional links presented in Subsection 4.5.2. Figure 7.1 shows the interactional links corresponding to the different types of messages being input to a DMU. Figure 7.2, however, presents the Petri Net representation of the different classes of messages as viewed in terms of the output from a DMU. It should be noted that a designer is free to interpret the interactional links present in the organizational structure in terms of the classes of messages presented in Section 7.1 to a degree of refinement that suits the very application.

7.3 DEFINITION OF ECHELONS IN AN ORGANIZATIONAL STRUCTURE

As mentioned in Chapter IV, echelons refer to the mutual relationship among DMUs of an organizational structure. The echelons define *superordinate* and *subordinate* DMUs within an organization. The identification of a superordinate or a subordinate DMU can only be done by analyzing the type of messages that a particular DMU is receiving from and/or sending to other organization members. Therefore, the interactional structure of a DMU is taken as criterion for defining the relative position of the DMU in the multiechelon hierarchy of an organization under study. In an attempt to define a multiechelon hierarchical structure of an organization in terms of the interactional structure of the decision making units (DMUs) comprising the organizational structure, the interactions of a DMU are divided into two classes: *input interactions,* and *output interactions.* The following sections present two separate multiechelon hierarchical structures in terms of all the possible input and output interactional structures that a DMU can have while performing its assigned task in an organizational environment. An echelon with lower index value is considered to be at a higher level than one with high index value, i.e., the echelon '1' is the superordinate level as compared to the echelon '2'. The echelon '0' is taken as the highest echelon; a DMU defined at echelon '0', if it exists, is considered as the *executive* of the organization under consideration.

7.3.1 Ordering in Terms of Inputs

The three classifications of the organizational data given in Section 7.1 yield 2^3-1 different input interactional structures of a DMU. The seven possible ways in which a DMU can receive input messages are given in the first column of Table 7.1. The line of reasoning used to define a multiechelon hierarchy in terms of the input interactional structure of the DMUs is as follows:

Let the outputs from a set of DMUs be taken as constant and let only the input interactions of the DMUs be considered. Then, a DMU receiving CTR or CMD type of messages is considered at a lower echelon than the one receiving INF messages. A number of sublevels are also defined within the DMUs having INF as input interaction. The DMUs receiving responses are taken at a higher echelon than the DMUs receiving inputs or assessments. Similarly, DMUs with assessment type of input interactions are considered at a higher echelon than the DMUs with input type of INF.

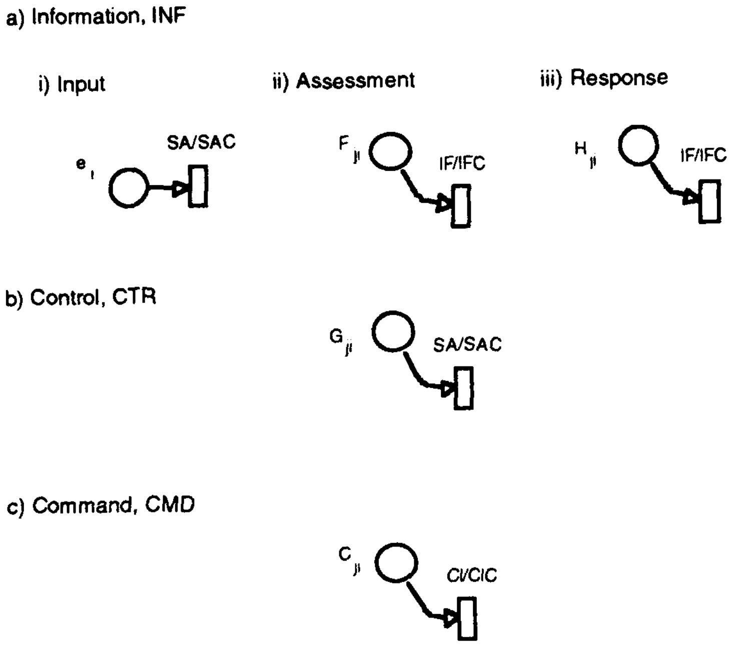


Figure 7.1 Classification of Input Interactions

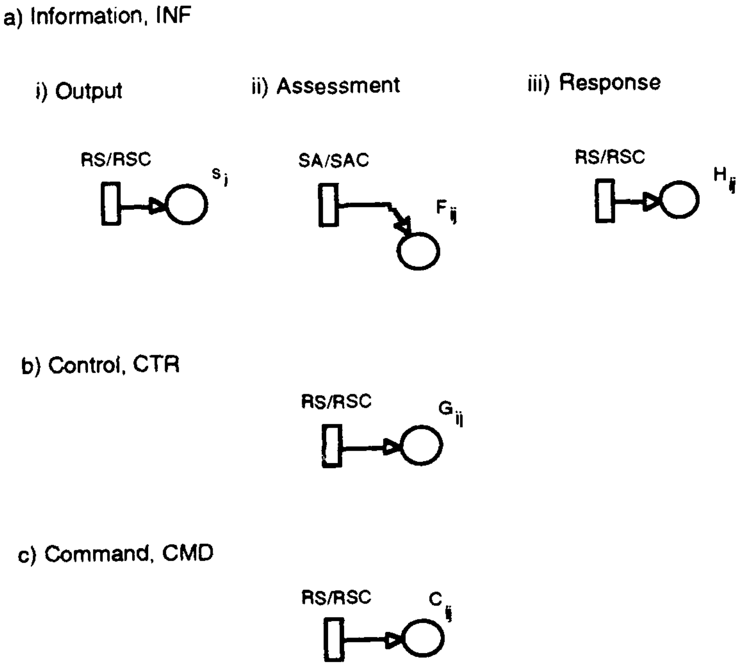


Figure 7.2 Classification of Output Interactions

A DMU with CTR input is considered at a higher echelon than the ones with CMD inputs. The DMUs with all other combinations of input interactions fall within these three echelons. Table 7.1 shows all the input interactions associated with a DMU and the corresponding ordering on the basis of the echelons to which they belong. A DMU with an order 1 is considered at the highest echelon as compared to all other DMUs with the identical set of output interactions. Note that the ordering defined in Table 7.1 is based on the input interactional structure of the DMU and is independent of the output interactional structure of the DMU.

TABLE 7.1 Ordering in terms of Input Messages

Input Interactions	**Corresponding Order**
INF	1
INF, CTR	2
CTR	3
INF, CMD	4
INF, CTR, CMD	5
CTR, CMD	6
CMD	7

7.3.2 Ordering in Terms of Outputs

The echelons for the DMUs in terms of their output interactional structure are defined reciprocally to the definition of the echelons presented in Subsection 7.3.1: The DMUs with INF outputs are taken at lower echelons than the ones with CTR or CMD outputs for a set of DMUs, where all the inputs to the DMUs are taken as constant throughout the set and only output interactions are considered. Similarly, DMUs with CTR outputs are at lower echelons than the ones with CMD. Table 7.2 presents DMUs with all possible output interactions and the corresponding ordering for the DMU based on their echelon definition. A DMU with an order 1 is considered at the highest echelon as compared to the all other DMUs with the same identical input interactions. Note, the order presented in Table 7.2 is defined in view of the output interactional structure of a DMU and is independent of the input interactions.

7.4 MULTIECHELON HIERARCHICAL ORGANIZATIONS

In this section, an echelon index is defined for a DMU based on both input and output interactional structures of the DMU. Table 7.3 shows a matrix between all the possible input and output interactions that a DMU can have along with the ordering defined for the input and output interactions. The rows and columns of the matrix are arranged according to the ordering defined for each element of the row/column in Subsections 7.3.1 and 7.3.2. A DMU is characterized as a 2-tuple, *(I, O)*, where 'I' corresponds to the order defined by the input interactions of the DMU (Table 7.1), and 'O' represents the order defined by the output interactions of the DMU (Table 7.2). The set of all the elements of the matrix is represented by Π. It is defined to be the set of DMUs with all the possible interactional structures and their associated input and output ordering. The classification of input and output interactions presented in the previous section yields 49 (7x7) elements for the set Π.

TABLE 7.2 Ordering in terms of Output Messages

Output Interactions	**Corresponding Order**
CMD	1
CTR, CMD	2
INF, CTR, CMD	3
INF, CMD	4
CTR	5
INF, CTR	6
INF	7

TABLE 7.3 Matrix Between Input and Output Orderings

Output / Input	CMD	CTR, CMD	INF, CTR, CMD	INF, CMD	CTR	INF, CTR	INF
INF	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)
INF, CTR	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)
CTR	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)	(3, 7)
INF, CMD	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)
INF, CTR, CMD	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)	(5, 7)
CTR, CMD	(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)	(6, 7)
CMD	(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)	(7, 6)	(7, 7)

Proposition 7.1

The set Π is partially ordered by the binary relation '«'.

The binary relation '«' has been defined in Chapter III. Proposition 7.2 is a direct consequence of the Proposition 7.1.

Proposition 7.2

The set Π is a **lattice**.

The lattice structure of Π is obtained in Figure 7.3 as a result of the partial ordering that exists between the elements of the set Π. The arrows represent the relation "is higher than", i.e., [A] → [B] means that 'A' is higher than 'B'. The echelon index for a DMU is now defined by the following equation.

$$\text{Echelon Index} = \mathrm{I} + \mathrm{O} - 2 \tag{7.1}$$

Figure 7.3 presents the multiechelon hierarchy that will be used in defining the echelon index associated with an organizational member. In the methodology, after unfolding a compound node to the next lower stratum, each of the DMUs of the compound nodes is identified as one of the elements of the set Π. Once the echelon indices associated with all the subsystems of the compound node are identified, a number of *connectivity rules* are specified to translate an interactional link defined in a higher stratum to its lower-stratum description, on the basis of the multiechelon hierarchy presented in this section.

7.5 RULES OF CONNECTIVITY

A number of simple rules are defined to resolve the lower-stratum connectivity of higher-stratum interactional links. As mentioned in Chapter V, an interactional link at a higher stratum will be translated into only one interactional link in the lower stratum. The restriction imposed by this condition and the rules that will follow are formulated as a result of ideas borrowed from Information Sciences.

Let us consider two compound nodes 'i' and 'j' in stratum 'k'. The compound nodes 'i' and 'j' themselves are DMUs of an organizational structure 'q' defined at stratum 'k-1'. The subsystems (DMUs) of compound node 'i' are given as 'a' and 'b', while compound node 'j' is composed of DMUs 'c' and 'd', Figure 7.4 (reproduction of Figure 5.7). The rules of connectivity now can be formulates as follows:

- Rule 7.1

An interactional link defined at stratum 'k' from a compound nodes 'i' to another compound node 'j' is translated into a single link at stratum 'k+1' from DMU 'a' or 'b' to DMU 'c' or 'd'.

- Rule 7.2

The translated lower stratum interactional link between the subsystems of the compound nodes 'i' and 'j' will connect the highest echelon-DMUs of the two suborganizational structures. The highest echelons identified for the subsystems of 'i' and 'j' need not necessarily be the same.

• Rule 7.3

If a compound node has two or more DMUs at the same highest echelon, the following rule applies:

- For an output interaction the DMU with higher 'O' index is selected
- For an input interaction the DMU with higher 'I' index is selected.
- For two or more DMUs with identical (I, O) indices, one of them is selected arbitrarily.

• Rule 7.4

If in following Rules 7.1 to 7.3 constraint R1 or R2 is violated, then the next highest echelon-DMU will be selected to participate in the interaction. The identification of the next highest echelon-DMU follows the procedure presented in Rules (7.2) and (7.3).

Example 7.1

Figure 7.4(a) shows a WDN Σ_{qk} in stratum 'k' of a compound node 'q' in stratum 'k-1'. The DMUs defined in stratum 'k' are 'i' and 'j'. The organizational structure shown in Figure 7.5(a) has an interactional link from RSC stage of DMU 'i' to CIC stage of DMU 'j'. The organizational structure in Figure 7.5(a) is unfolded to its stratum 'k+1' description in Figure 7.5(b). The DMU in stratum 'k+1' are identified as DMUs 'a', 'b', 'c' and 'd'. The dotted arcs in Figure 7.5(b) indicate all possible connections in stratum 'k+1' that may represent the interactional link defined in stratum 'k'. Let us consider that the echelons identified for the DMUs 'a', 'b', 'c' and 'd' are '0', '12', '6', and '6' respectively. Now, the interactional link in stratum 'k' will be described by a single interaction of the same kind between corresponding stages of the DMUs 'a' and 'c' since 'a' and 'c' are the highest echelon-DMUs defined within the suborganizational structures 'i' and 'j' respectively, Figure 7.5(c).

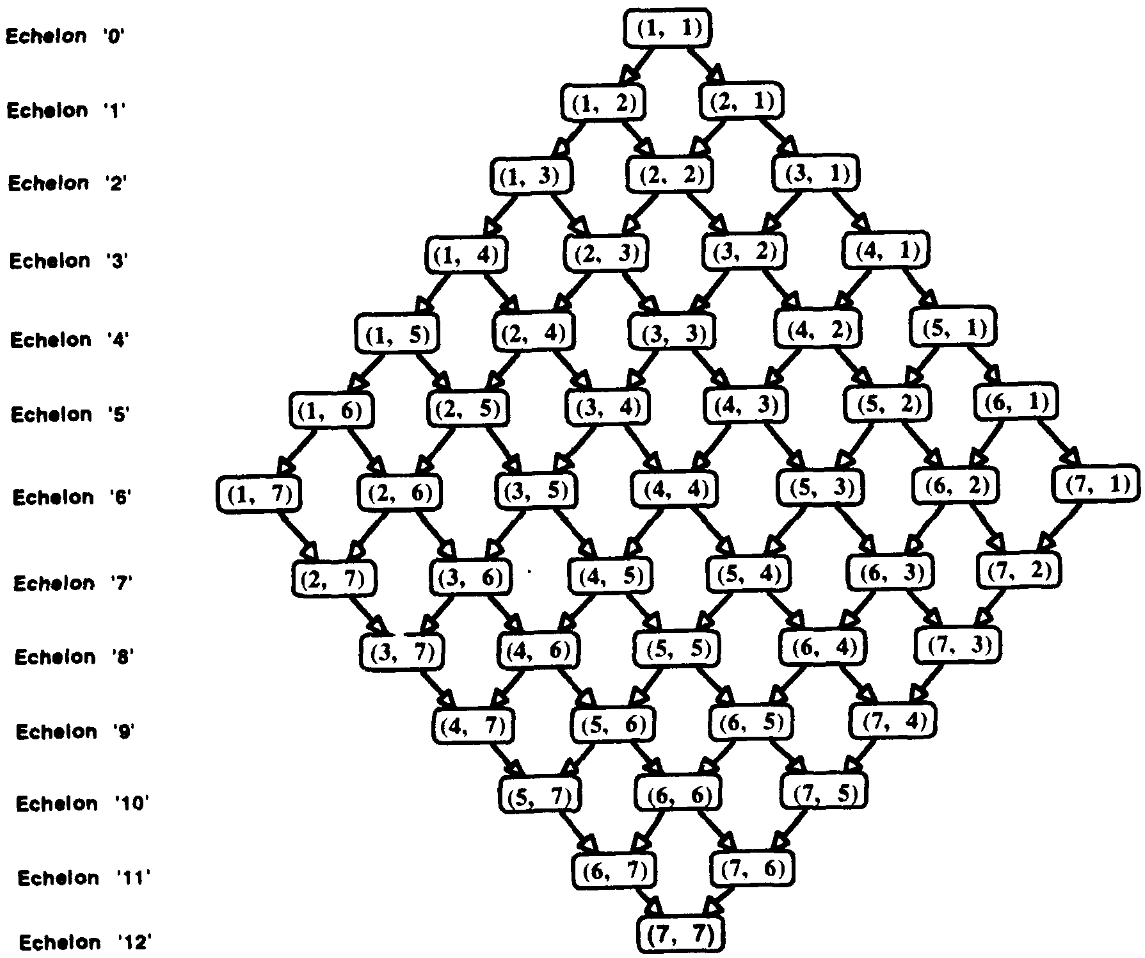


Figure 7.3 Multiechelon Hierarchy

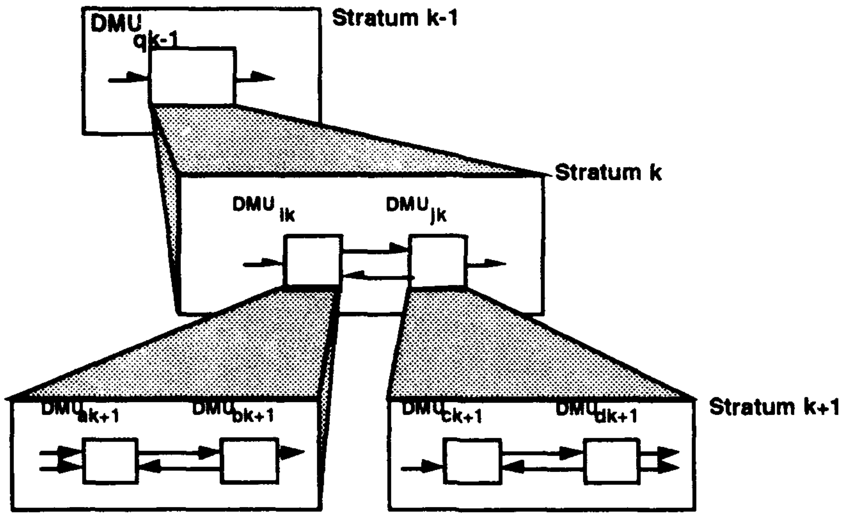
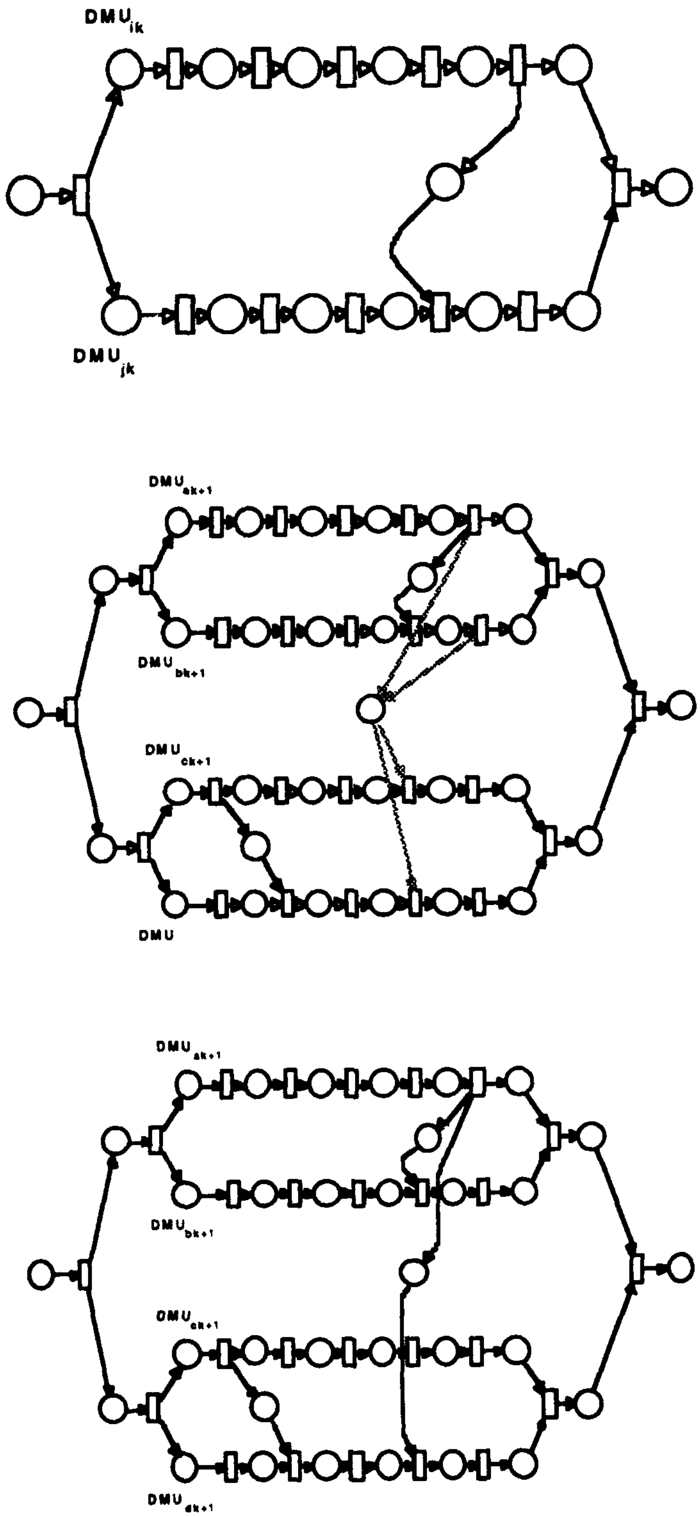


Figure 7.4 A Stratified Decision Making Organization

7.6 EXAMPLE

The connectivity rules presented in Section 7.5 are applied to the organizational structure of Example 5.3, Subsection 5.6.5. As it is mentioned in Chapter V, Figure 5.41 shows all the possible combinations of connections between DMUs in stratum '2' that may exist as a result of translating the interactional link between DMU_{11} and DMU_{21} in stratum '1' to its stratum '2' representation. However, according to Rule 7.1, only one interactional link between the subsystems in stratum '2' will represent the higher stratum (stratum '1') interactional link denoted by the interactional place p_{61241} in Figure 5.41. In order to identify the potential DMUs in stratum '2' that will have the desired single interactional link between them, the echelon index for each of the DMUs in stratum '2' is calculated. The echelon indices associated with the DMUs DMU_{12}, DMU_{22}, DMU_{32}, DMU_{42}, DMU_{52}, and DMU_{62} are identified as 0, 9, 6, 3, 3, and 9 respectively. DMU_{12} of the compound node '1' in stratum '1' has the highest echelon index (0) compared to the other subsystems of the compound node '1'. Therefore, the RS stage of DMU_{12} is selected as the input stage to the interactional place p_{61241} (Rule 7.2).



(a)

(b)

(c)

Figure 7.5 Lower Stratum Representation of a Higher Stratum Interactional Link

The highest echelon index identified for the subsystems of the compound node '2' in stratum '1' is '3' which is associated with two DMUs, DMU_{42} and DMU_{52}. However, the 2-tuple (I, O) representation, presented in Section 7.4, of the echelon index associated with DMU_{42} is given by (1, 4), while that of DMU_{52} is (2, 3). According to Rule 7.3, DMU_{42} is selected to participate in the interactions since it has the higher 'I' index. Application of the connectivity rules results in the organizational structure shown in Figure 7.6.

The incidence matrix of the net shown in Figure 7.6 is presented in Figure 7.7.

7.7 CONCLUSION

A set of connectivity rules to resolve the lower stratum representation of higher stratum interactional links was presented. The connectivity rules are based upon the multiechelon hierarchical relationship that may exist among the DMUs of an organizational structure. In order to define the multiechelon hierarchy among organizational members, the messages that flow in an organization are classified into different categories. An ordering was then defined on input and output messages to characterize the echelon type relationship among different organizational members. An echelon index was defined based on the orderings described for input and output messages. According to the methodology presented in this chapter, whenever it is desired to translate a higher stratum interactional link to its lower stratum description, all the subsystems defined in the lower stratum are identified in terms of their echelon indices and then the connectivity rules are applied to the structure. The methodology was applied to resolve the connectivity problem to an example presented in Chapter V.

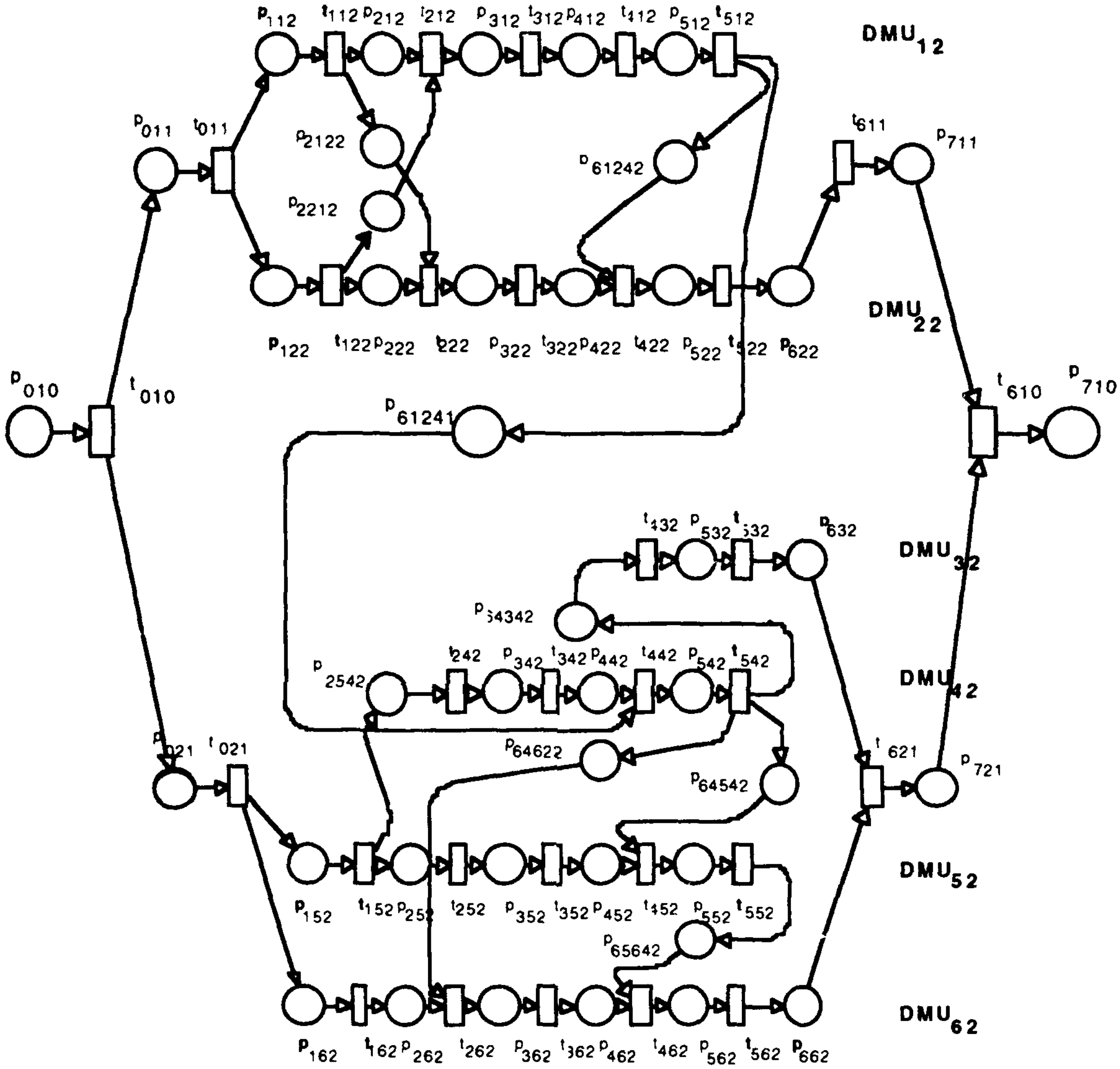


Figure 7.6 Application of the Connectivity Rules to the Organization in Example 5.3

$$
\Delta_{102} = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & C & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

Figure 7.7 Incidence Matrix of the Net in Figure 7.6

CHAPTER VIII

THE LATTICE ALGORITHM

A Feasible Organization (FO) has been defined as a Well Defined Net (WDN) in stratum 'k+1' of a compound node in stratum 'k', that satisfies both the structural and user-defined constraints. The compound node for which the WDN is defined can be an entire organization, a case where the compound node is located at the highest stratum (stratum '0'), or a subsystem defined for an arbitrary stratum 'k', where $1 \leq k < n$. The design methodology, introduced by (Remy, 1986), presented in this chapter determines the set of all Feasible Organizations, defined in stratum 'k+1', for a compound node 'i', in stratum 'k', corresponding to a specific set of constraints. It is assumed throughout this chapter that the user-defined constraints R_u are given.

8.1 CHARACTERIZATION OF THE SET OF FEASIBLE ORGANIZATIONS W(R)

As mentioned, once the set of special constraints is given, the notions of WDN and Well Defined Structure (WDS) are interchangeable. The term WDN will, therefore, be used throughout the chapter. The set of Feasible Organizations W(R) is a subset of the set of all WDNs W. Since, the set W is an ordered set, Chapter IV, according to Theorem 3.1 the set of Feasible Organizations (FO) is a partially ordered set with the same order (denoted $\leq$) defined on W, (Remy, 1986). From Theorem 3.2, the set of Feasible Organizations (FO) W(R) has at least one minimal and one maximal elements. Note that for the sake of simplicity the node and stratum indices are not shown.

Definition 8.1: *Universal and Kernel Nets*

The Universal Net associated with the constraints R_u - $\Omega(R_u)$ - is the WDN obtained by replacing all undetermined elements of {**e**, **s**, **F**, **G**, **H**, **C**} by 1.

Similarly the Kernel Net - $\omega(R_u)$ - is the WDN obtained by replacing the same undetermined elements by 0.

The set $W(R_u)$ of all Admissible Organization Forms is characterized by the following proposition.

Proposition 8.1

The set $W(R_u)$ is the subset of W that satisfies the following two conditions:

- Any element Σ of $W(R_u)$ is a subnet of the Universal Net $\Omega(R_u)$.
- The Kernel Net $\omega(R_u)$ is a subnet of any element Σ of $W(R_u)$.

Alternatively,

$$W(R_u) = \{ \Sigma \in W \mid \omega(R_u) \leq \Sigma \leq \Omega(R_u) \}$$

Proof

Since user-defined constraint R_u is convex on W, the convex subset $W(R_u)$ of set W satisfying R_u can be characterized by its minimal and maximal elements, Proposition 6.1. The proof is completed, if it is noted that $W(R_u)$ has a single maximal element (the Universal Net) and a single minimal element (the Kernel Net).

Corollary (Remy 1986)

$W(R_u)$ is a sublattice of W.

Definition 8.2: *Maximally (Minimally) Connected Organization*

A maximal element of the set W(R) of all Feasible Organizations will be called a Maximally Connected Organization (MAXO). The set of all MAXOs will be denoted as $W_{max}(R)$.

Similarly, a minimal element of W(R) will be called a Minimally Connected Organization (MINO). The set of all MINOs will be denoted as $W_{min}(R)$.

Maximally and minimally connected organizations can be interpreted as follows. A MAXO is a WDN such that it is not possible to add a single link without violating the set of constraints R, i.e., without crossing the boundaries of the subset W(R). Similarly, a MINO is a WDN such that it is not possible to remove a single link without violating the set of constraints R. The following proposition is a direct consequence of the definition of maximal and minimal elements.

Proposition 8.2 (Remy 1986)

For any given feasible Organization Σ, there is at least one MINO Σ_{min} and at least one MAXO Σ_{max} such that $\Sigma_{min} \leq \Sigma \leq \Sigma_{max}$. Alternatively,

$$\{ \Sigma \in W \mid \exists (\Sigma_{min}, \Sigma_{max}) \in W_{min}(R) \times W_{max}(R)\ \Sigma_{min} \leq \Sigma \leq \Sigma_{max} \} \supset W(R)$$

Note that the previous inclusion is not an equality in the general case. As mentioned earlier, the constraint R1 is not convex (Proposition 6.3) on the set W for a compound node with human DMs as DMUs. Therefore, there is indeed no guarantee that a WDN located between a MAXO and MINO will fulfill the constraint R1, since such a net need not be connected. To address this problem, the concept of simple path has been introduced by Remy (1986). The set of WDNs for compound nodes with subsystems other than human DMs do not have this problem, as the constraints C1 and C2 are convex and together they imply constraint R1, Proposition 6.4. Therefore, the set of Feasible Organizations for such a compound node can be completely characterized by the MAXOs and MINOs.

8.1.1 Simple Paths

Let Σ be a WDN that satisfies constraint R1 and whose source and sink have been merged together into a single external place. If the source and sink places of a Σ are merged together to form an external place, then a simple path of Σ is defined to be a directed elementary circuit which includes the external place.

According to Proposition 5.1, the Petri Net representing Σ is a marked graph. A simple path is therefore a minimal support S-invariant of Σ whose component corresponding to the external place is 1 (Hillion, 1986). Note that if the component corresponding to the external place is not 1 then the S-invariant is an internal loop of the net. The concept of S-component has been presented in Chapter II. An S-component of a WDN is itself a WDN whose places are exactly the places of the support of the S-invariant and transitions are the input and output transitions of these places. The matrix representation of the S-component is obtained by identifying the interactional places of the S-component. Consequently, the simple paths of a WDN are themselves WDNs. The set of all simple paths of the Universal Net $\Omega(R_u)$ are denoted as $Sp(R_u)$.

$$Sp(R_u) = \{sp_1, sp_2, ..., sp_i, ..., sp_r\} \qquad sp_i \leq \Omega(R_u)$$

8.1.2 Union of Simple Paths

If the cardinality of $Sp(R_u)$ is r, we can write $Sp(R_u) = \{sp_i, 1 \leq i \leq r\}$. Since simple paths are WDNs, the set $Sp(R_u)$ is included in the set of all WDNs, W. The set of all possible unions of elements of $Sp(R_u)$, augmented with the null element φ of W, is denoted as $USp(R_u)$. The null element φ is defined to be a WDN with all elements equal to zero.

$$USp(R_u) = \{\Sigma \in W \mid \exists \{sp_{i1}, ..., sp_{ir}\} \in Sp(R_u)^r \; \Sigma = sp_{i1} \cup ... \cup sp_{ir}\} \cup \{\varphi\}$$

The union of two elements of $Sp(R_u)$ is the WDN composed of all the simple paths included in either one of the two considered elements.

The following proposition justifies the introduction of simple paths.

Proposition 8.3 (Remy 1986)

Every WDN, element of the set $USp(R_u)$, satisfies the connectivity constraint R1. Reciprocally, a Feasible Organizational Form that fulfills the constraint R1 is an element of $USp(R_u)$. In formal language:

$$\{\Sigma \in W \mid R1[\Sigma] = 1\} \supset USp(R_u) \supset \{\Sigma \in W(R_u) \mid R1[\Sigma] = 1\}$$

$R1[\Sigma] = 1$ means that Σ satisfies the constraint R1.

8.1.3 Characterization of W(R)

The following proposition, Remy (1986), characterizes the set of all feasible organizations.

Proposition 8.4

Let Σ be a WDN of a compound node defined in a stratum 'k' of dimension 'm'. Σ will be a Feasible Organization if and only if

- Σ is a union of simple paths of the Universal Net $\Omega(Ru)$, i.e., $\Sigma \in USp(R_u)$.
- Σ is bounded by at least one MINO and one MAXO.

In formal language:

$$W(R) = \{\Sigma \in USp(R_u) \mid \exists (\Sigma_{min}, \Sigma_{max}) \in W_{min}(R) \times W_{max}(R)\ \Sigma_{min} \leq \Sigma \leq \Sigma_{max}\}$$

As mentioned earlier, the characterization of the set W(R) for the cases where organizational structure is comprised of the DMUs other than human decision makers is much simpler. The following proposition characterizes the set W(R) for such cases.

Proposition 8.5

Let Σ be a WDN of a compound node defined in a stratum 'k', where $k \neq N$, of dimension 'm'. Σ will be a Feasible Organization if and only if

- Σ is bounded by at least one MINO and one MAXO.

In formal language:

$$W(R) = \{\Sigma \in W(R_u) \mid \exists (\Sigma_{min}, \Sigma_{max}) \in W_{min}(R) \times W_{max}(R)\ \Sigma_{min} \leq \Sigma \leq \Sigma_{max}\}$$

Propositions 8.4 and 8.5 gives a characterization of the set W(R) just like Proposition 8.3 gives a characterization to the set $W(R_u)$. In the cases where the DMUs of the organizational structure are not human decision makers, a link is the incremental unit leading from a WDN to its immediate superordinate, while in the cases where human decision makers are defined as the DMUs of the organizational structure, the simple paths play the role of building unit. In generating organizational structures with simple paths, the connectivity constraint R1 is automatically satisfied. The following section illustrates the methodology by applying it to an example problem.

8.2 APPLICATION

Let us consider the set of user-defined constraints presented in Figure 8.1 corresponding to 2-dimensional WDNs associated with a compound node DMU_{10}, where the WDNs are defined in stratum 'n=N=1', the lowest stratum. The 'x' in the arrays of Figure 8.1 corresponds to the unspecified elements. The 0's and 1's indicate the forced absence or presence, respectively, of links. Note that all the diagonal elements are identically 0 as they represent the inadmissible links.

The organization under consideration has two DMUs. DMU_{11} acts as the sensor of the organization; it receives information from the external environment. DMU_{21} produces the organization's response with respect to the external environment. All other interactions between these two DMUs and the external environment are optional. The Universal Net $\Omega(R_u)$ is obtained by replacing all x's by 1's. The net $\Omega(R_u)$ is represented in Figure 8.2 with bold connectors representing the links imposed by user-defined constraints R_u. The Kernel Net $\omega(R_u)$ is given in Figure 8.3 obtained as a result of replacing all unspecified elements by 0's. The MAXO (M) and MINO (m) identified for the set W(R) are given in Figure 8.3. All the simple paths calculated are given in Figure 8.4. The lattice representation of W(R) is presented in Figure 8.5. The Hasse diagram presented in Figure 8.6 is constructed by taking the only MINO (m) found in the set $W(R_u)$ and by adding different simple paths to it. All the WDNs found as a result of adding these simple paths to the MINO (m) are the elements of the set W(R), thus representing all the Feasible Organizational structures for the given organization.

$$\Sigma_{11} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$$

$$e = [\,1 \quad x\,] \qquad F = \begin{bmatrix} 0 & 0 \\ x & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$s = [\,x \quad 1\,] \qquad H = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Figure 8.1 Matrix Representation Of Σ_{11}

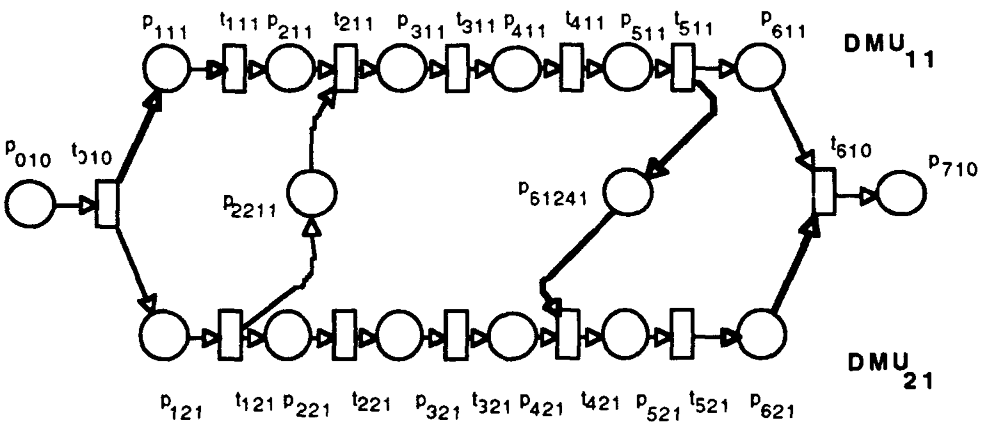


Figure 8.2 Universal Net $\Omega(R_u)$

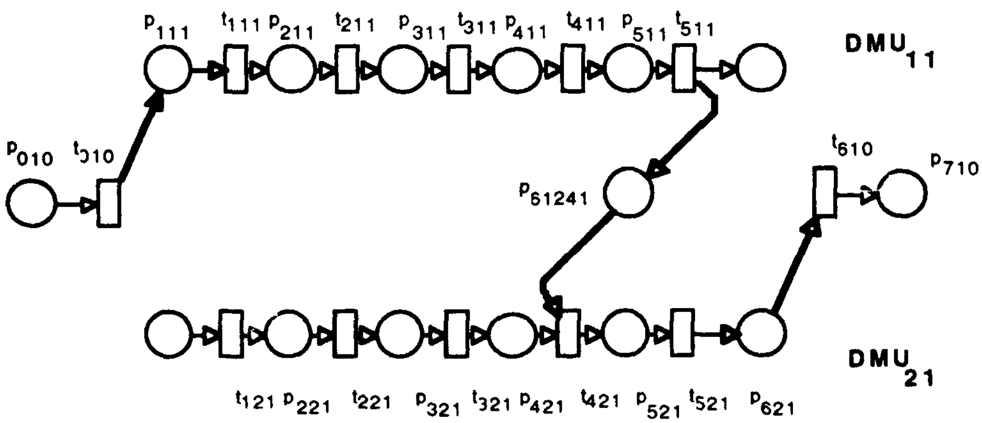


Figure 8.3 Kernel Net $\omega(R_u)$

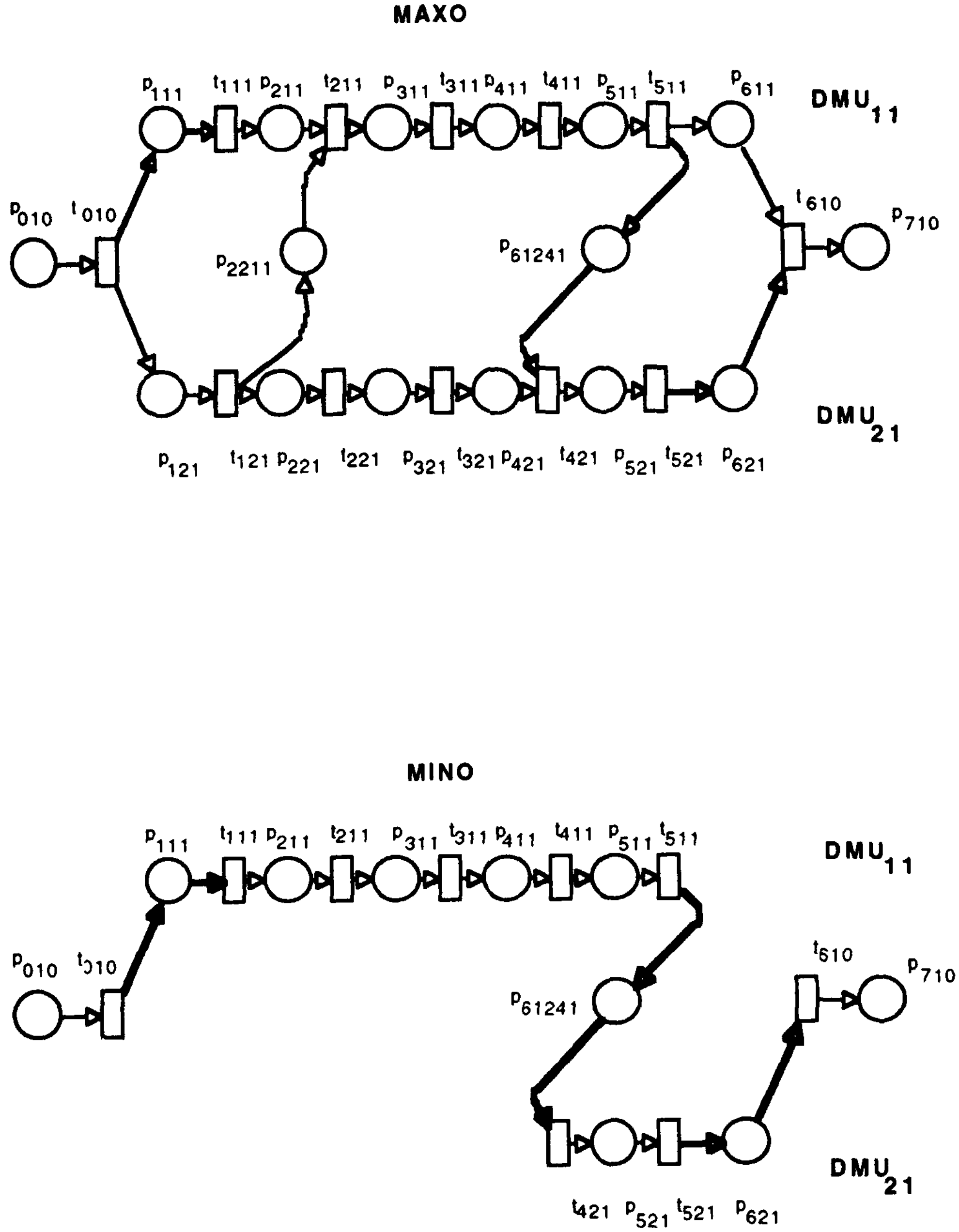


Figure 8.4 MAXO (M) and MINO (m) of Σ_{11}

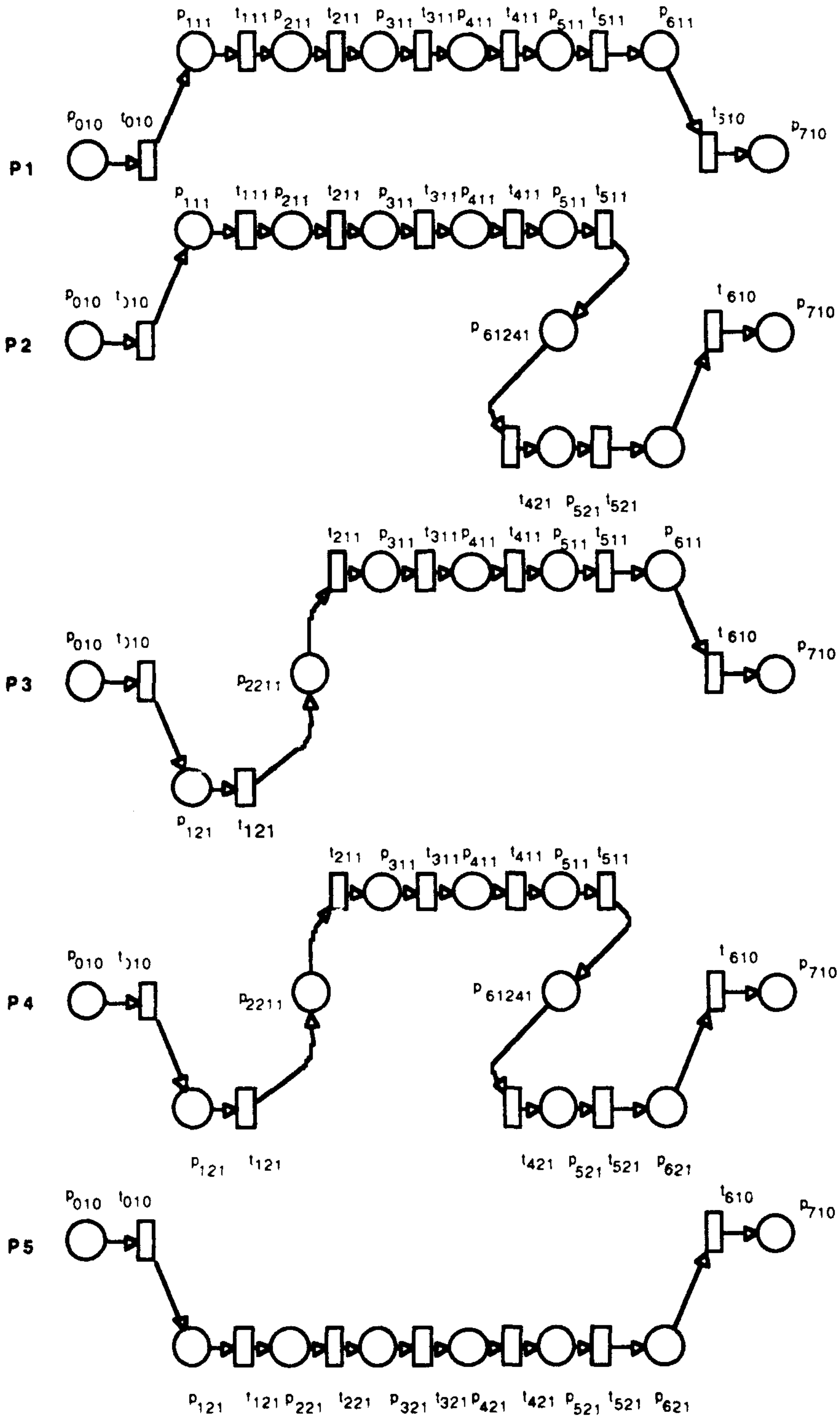


Figure 8.5 Simple Paths

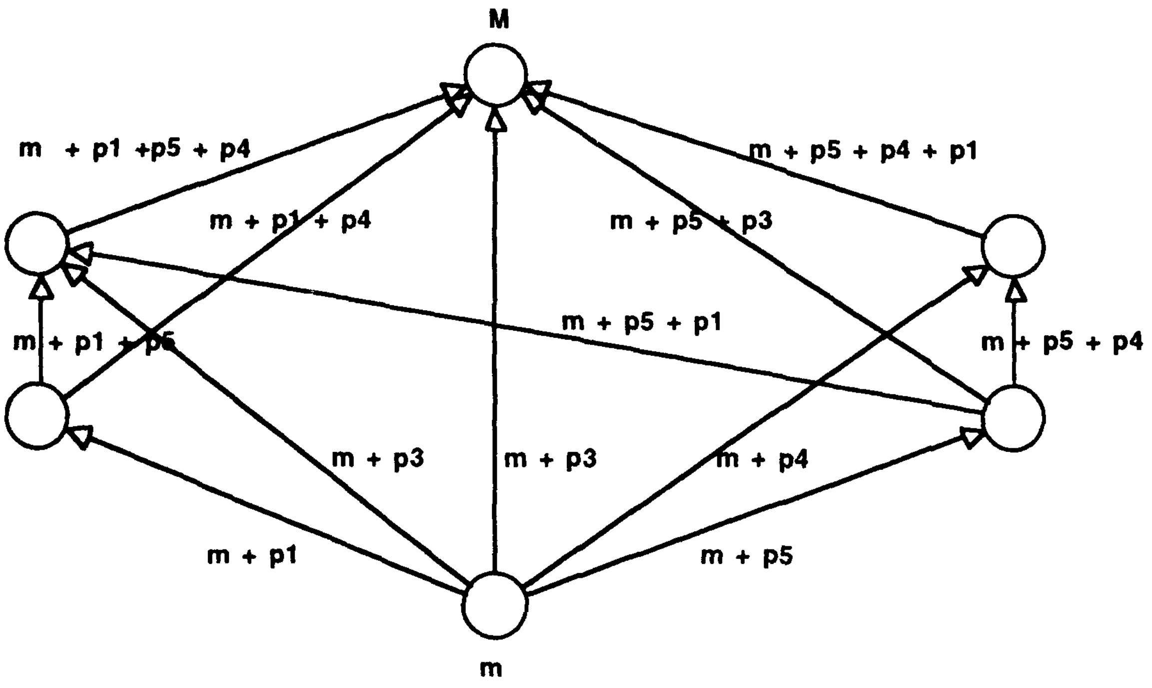


Figure 8.6 Hasse Diagram of the Lattice m-M

8.3 CONCLUSION

A review of the Lattice algorithm, introduced by Remy (1986), was presented in this chapter. A slight modification was presented in the algorithm for the organizational structures with DMUs other than human decision makers - a case not covered by Remy. It was found that the introduction of the compound node constraints (C1 and C2) has resulted in a mere simplification of the Lattice algorithm. A characterization of the set of all Feasible Organizational structures W(R) for a given set of user-defined constraints was presented for both cases (organizational structures with or without human DM). The algorithm was applied to an example problem to illustrate the methodology.

CHAPTER IX

APPLICATION OF THE ALGORITHM TO THE DESIGN PROBLEM

9.1 INTRODUCTION

This chapter presents the entire algorithm for generating Stratified Decision Making Organizational (SDMO) structures. The algorithm is developed by connecting the concepts and results presented in Chapters IV to VIII. A *flowchart* description and a *pseudocode*, a *Program Design Language (PDL)*, description of the algorithm provide the entire design procedure. Section 9.2 presents the algorithm for the *Bottom-Up* design approach, while the *Top-Down* design procedure is given in Section 9.3. A comparative study of the two approaches is presented in Section 9.3. Section 9.4 consists of the entire algorithm incorporating both approaches. An example is presented to illustrate the methodology developed.

A flowchart description is used to depict the logic, procedures, and elements of the algorithm. The *American National Standards Institute (ANSI)* has defined standard flowcharts symbols and their usage, ANSI (1970). Figure 9.1 presents the symbols that are used in the development of the algorithm presented in this chapter.

9.2 BOTTOM-UP APPROACH

In the bottom-up approach, the lower strata nodes are designed first; subsystems are designed prior to the systems to which they belong. In this approach the design procedure starts with the specification of the lowest stratum (stratum 'n') nodes, and then these nodes are mapped into the next higher stratum (stratum 'n-1') compound nodes. The WDNs of the next higher stratum (stratum 'n-1') compound nodes, defined in terms of the lower stratum subsystems, are generated by the lattice algorithm applied to each node. The procedure continues till the highest stratum (stratum '0') compound node is defined in terms of its subsystems (DMUs) in stratum '1'. Once all the FOs are generated for each compound node in stratum 'k', where $0 \leq k < n$, in terms of the subsystems (DMUs) in stratum 'k+1', then the organizational structure can be unfolded to any stratum description with the help of the procedure presented in Subsection 5.7.4. After unfolding the organizational structure to an arbitrary stratum 'k', all the higher strata interactional link are translated to their lower stratum, stratum 'k', representation by the application of the connectivity rules presented in Section 7.5. As mentioned in Chapter VII, the connectivity rules can not be directly applied to stratum 'k' if the stratum to which the interactional links belong is not the next higher stratum, stratum 'k-1'. The connectivity rules must be applied to all intermediate strata first. Then the interactional links, appearing as a result of this successive application of the rules, in the organizational structure, in stratum 'k-1', are translated to the next lower stratum (stratum 'k') by the rules. In case a higher stratum description is required, the folding procedure presented in Subsection 5.7.3 is applied to the organizational structure. A flowchart description of the approach is presented in Figure 9.2, while a pseudocode description of the design procedure is shown in Figure 9.3.

9.3 TOP-DOWN APPROACH

In contrast to the bottom-up approach, the top-down design procedure generates the organizational structures of the the higher strata nodes first; systems are designed prior to the subsystems comprising them. In this approach, the design procedure starts with the highest stratum (stratum '0') node, and then the subsystems for the node are specified for the next lower stratum (stratum '1'). Each node in stratum '1' is then designed in terms of its own subsystems in lower strata. The procedure continues till the lowest stratum (stratum 'n') nodes are defined. The folding and unfolding procedures are presented in Section 5.7. The connectivity issue follows the same scheme presented in the previous section. A flowchart description of the approach is presented in Figure 9.4, while a pseudocode description of the design procedure is shown in Figure 9.5.

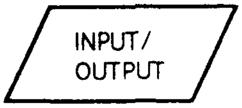
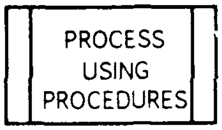
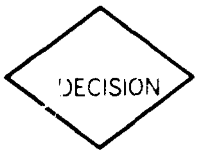
Symbol	Represents
	The input or output of data, where the medium of input or output is not specified.
	Manual input .
	Any manipulation or processing of data.
	Perform processing using a seperate procedure, function, or subprogram unit.
	The begining or end of an algorithm module.
	Taking of alternative actions based upon presence or absence of some condition. Often called a decision symbol.
	Annotation. Used for added comments. Connected to flowchart where helpful to provide additional information.
	Sequence and flow of logic.

Figure 9.1 Flowchart Symbols

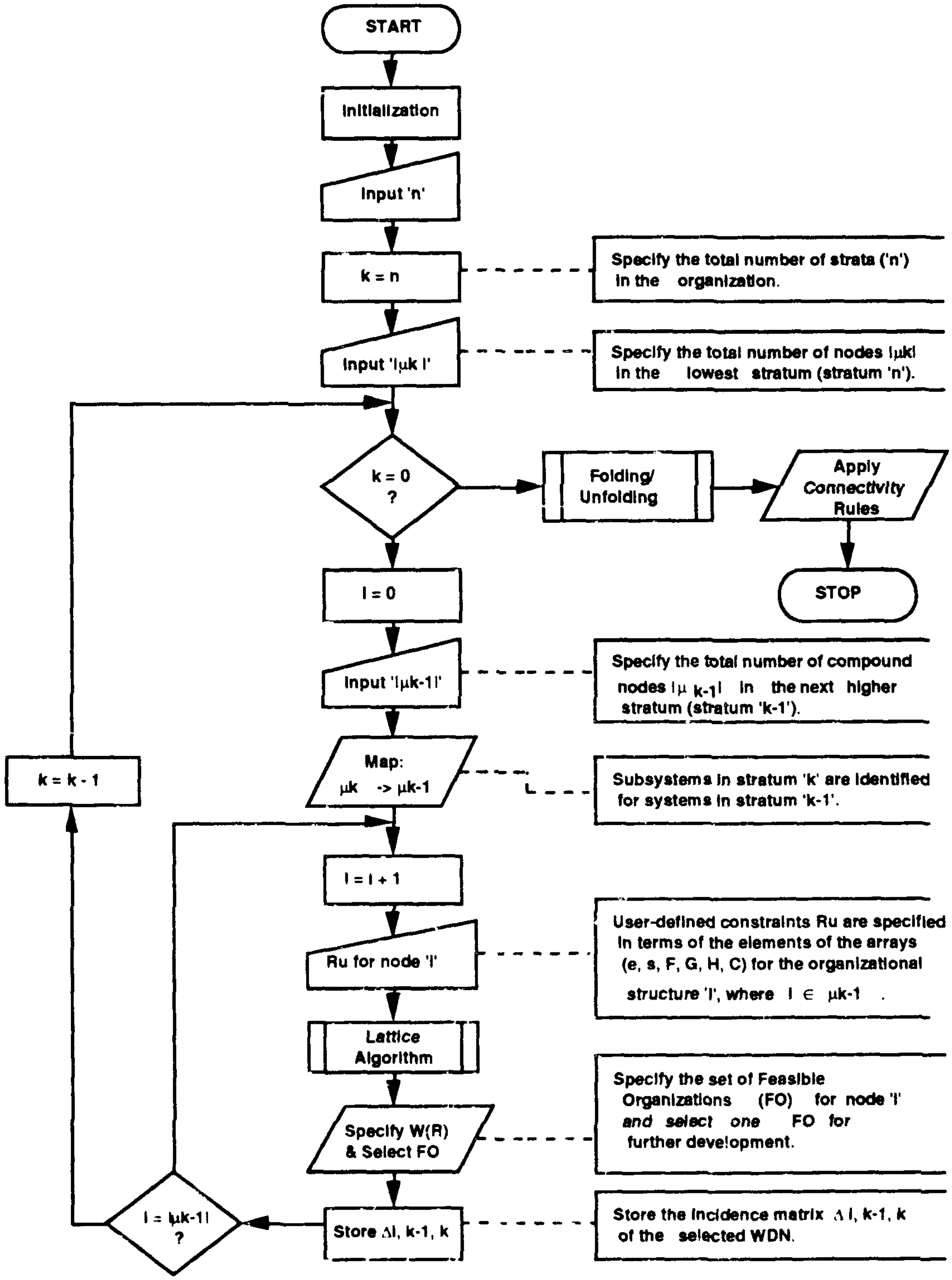


Figure 9.2 Flowchart for the Bottom-Up Approach

```
Begin
    initialization : var → int
    input (total # of strata 'n' in SDMO) : n → int
    k = n : k → int
    input (total # of DMUs in stratum 'k', |μk|) : |μk| → int

    for (k = n ; k > 0 ; -1)
    Begin for
        input (total # of DMUs in the next upper stratum, |μk-1|) :
                                                        |mk-1| → int
        map (|μk|, |μk-1|, m) : (|μk|, |μk-1|) → m: (int, int) → int

        for (i = 1 ; i ≤ |μk-1| ; +1)
        Begin for
            user-defined constraints Ru (e, s, F, G,H,C) :
                                        e, s, F, G, H, C → int
            lattice algorithm (Ru, R, W(R)) : (Ru, R) → W(R)
            select FO (W(R), Σik, Δi, k-1, k) : W(R) → (Σik, Δi, k-1, k)
            store (Σik, Δi, k-1, k)
        end for
    end for

    input (stratum 'l'' for which the description is required) : l → int
    folding/unfolding ((Σ11, ...,Σil), (Δ101, ...,Δikl), Δ10l ) :
                                (Σ11, ...,Σil), (Δ101, ...,Δikl) → Δ10l
    connectivity rules (R7.1, R7.2, R7.3, R7.4, Δ10l ) : Δ10l → Δ10l
end
```

Figure 9.3 Pseudocode Description of the Bottom-Up Design Procedure

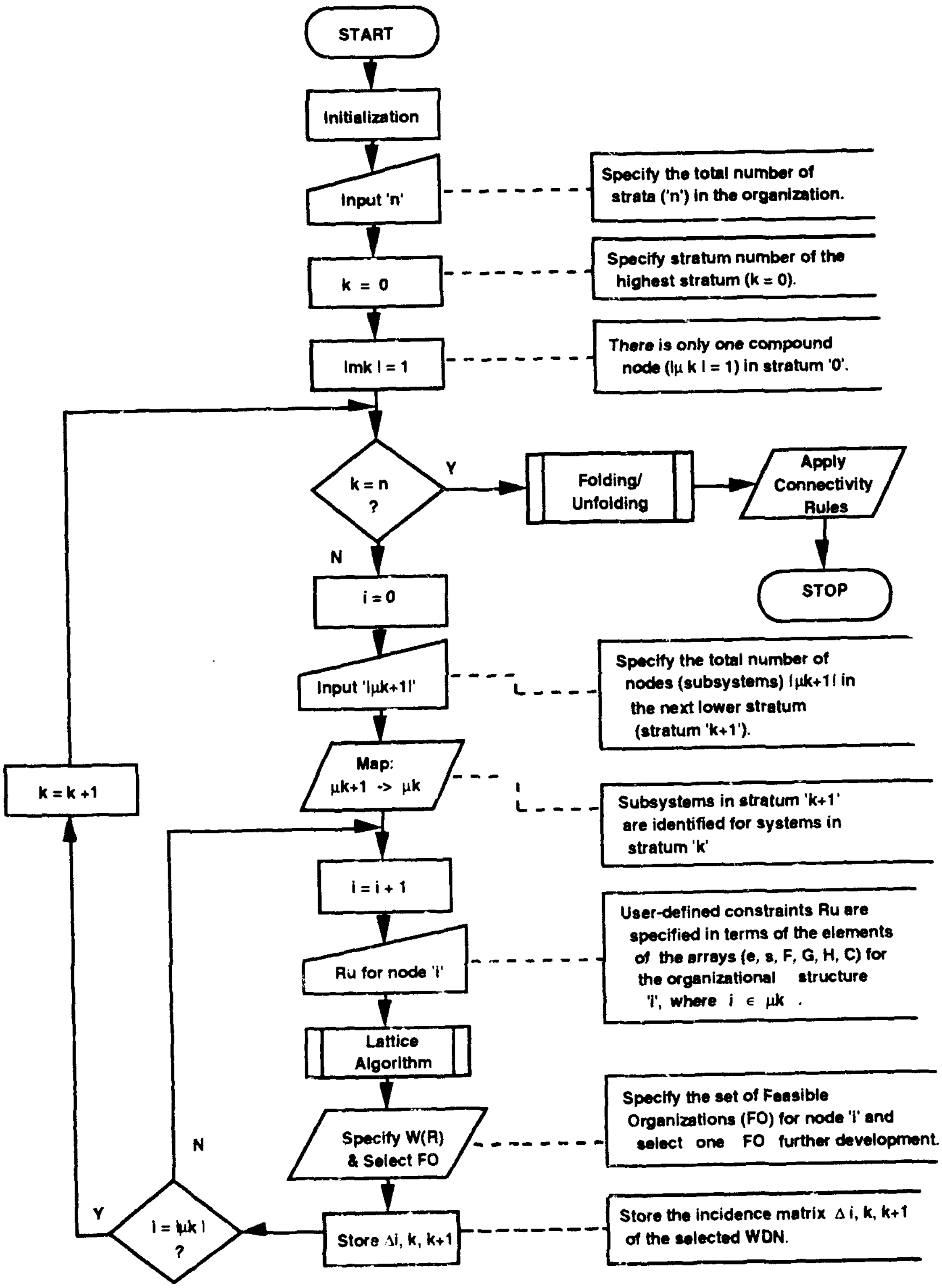


Figure 9.4 Flowchart for the Top-Down Approach

```
Begin
    initialization : var → int
    input (total # of strata 'n' in SDMO) : n → int
    k = 0 : k → int
    |mk| = 1 : |μk| → int

    for (k = 0 ; k < n ; +1)
    Begin for
        input (total # of DMUs in the next lower stratum, |μk+1|) :
                                                            |μk-1| → int
        map (|μk+1|, |μk|, m) : (|μk+1|, |μk|) → m; (int, int) → int

        for (i = |μk| ; i > 0 ; -1)
        Begin for
            user-defined constraints Ru (e, s, F, G,H,C) :
                                              e, s, F, G, H, C → int
            lattice algorithm (Ru, R, W(R)) : (Ru, R) → W(R)
            select FO (W(R), Σik+1, Δi, k, k+1) :
                                              W(R) → (Σik+1, Δi, k, k+1)
            store (Σik+1, Δi, k, k+1)
        end for
    end for

    input (stratum 'l' for which the description is required) : l → int
    folding/unfolding ((Σ11, ...,Σil), (Δ101, ...,Δikl), Δ10l ) :
                                   (Σ11, ...,Σil), (Δ101, ...,Δikl) → Δ10l
    connectivity rules (R7.1, R7.2, R7.3, R7.4, Δ10l ) : Δ10l → Δ10l
end
```

Figure 9.5 Pseudocode Description of the Top-Down Design Procedure

9.4 COMPARATIVE STUDY OF THE TWO APPROACHES

The two approaches presented in the previous sections provide identical results if the same Feasible Organizational structures (FOs) generated for each node are considered. Since the generation of WDNs for a particular node in an arbitrary stratum 'k', where $0 \leq k < n$, is independent of the generation of WDNs for any other node in any other stratum 'l' ($0 \leq l < n$), the end result of the design procedure is independent of the approach taken. However, each approach not only provides a systematic way of designing an organizational structure, but also gives the designer an option to choose the sequence in which the results are produced.

In case the emphasis of the designer is on the subsystems of the entire organization then the Bottom-Up approach should be adopted in order to generate the organizational structure. As mentioned earlier, in this approach, the subsystems are modeled before the system itself. Therefore, the design procedure produces information about the subsystems and the interactions among them even before the entire organization is modeled. If it is required to investigate the details of the subsystems of the organization at any point of the design procedure, the process can be interrupted at that point without even exercising the entire algorithm, and then all subsystems defined at the time of interruption can be investigated in terms of their lower strata descriptions thus saving a lot of time and computational effort, especially for very large systems. Note that the behavior of the subsystems in the entire organization can not be determined just by investigating their individual organizational structures since a number of interactions among them may be defined at a higher stratum.

On the other hand, if the emphasis is on the entire system instead of its subsystems and the manner in which it evolves, then the Top-Down approach is the one to be considered. In this approach, the design process starts with the entire system designed at the highest abstraction (highest stratum description) desired, then each subsystem is modeled in terms of its higher stratum description Therefore, the design procedure produces information about the system at a relatively higher degree of abstraction as the process evolves. At any point, the design process can be interrupted to investigate the entire organizational structure to a degree of abstraction that has been defined at the time of interruption without generating all the suborganizational structures at the lowest stratum.

9.5 OVERALL STRUCTURE OF THE ALGORITHM

The entire design procedure is presented in terms of a Flowchart description in Figure 9.6 by joining the two Flowcharts presented in Figures 9.2 and 9.4. The shaded boxes represent the processes that are different in each approach. No further explanation is required as both the approaches has been described in Sections 9.2 and 9.3.

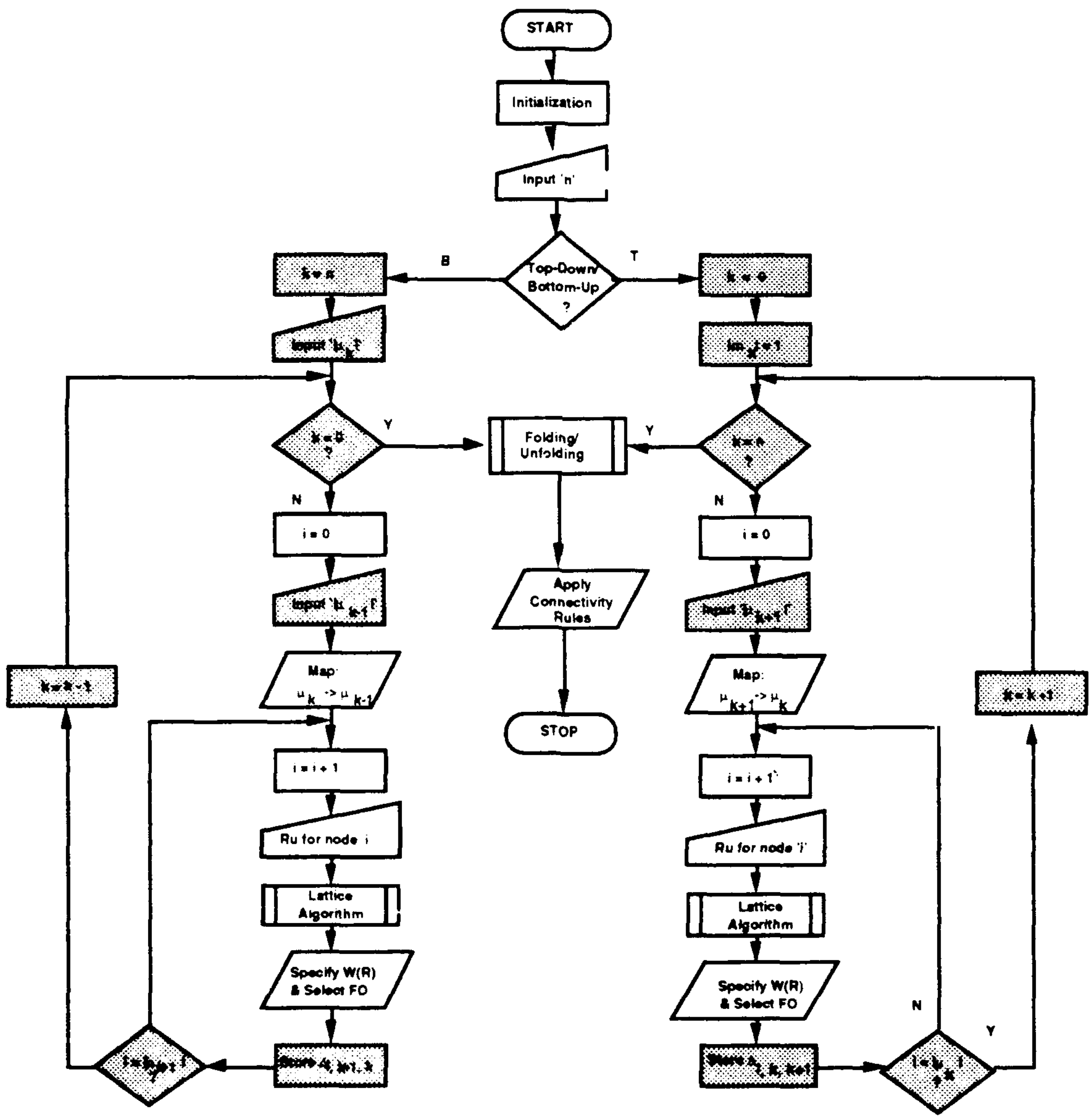


Figure 9.6 Overall Structure of the Algorithm

9.6 APPLICATION

In this section, the design methodology presented in this thesis is applied to a fairly simple and illustrative example. Figure 9.7 presents a block description of a 2-strata Stratified Decision Making Organization (SDMO) with n = N = 2.

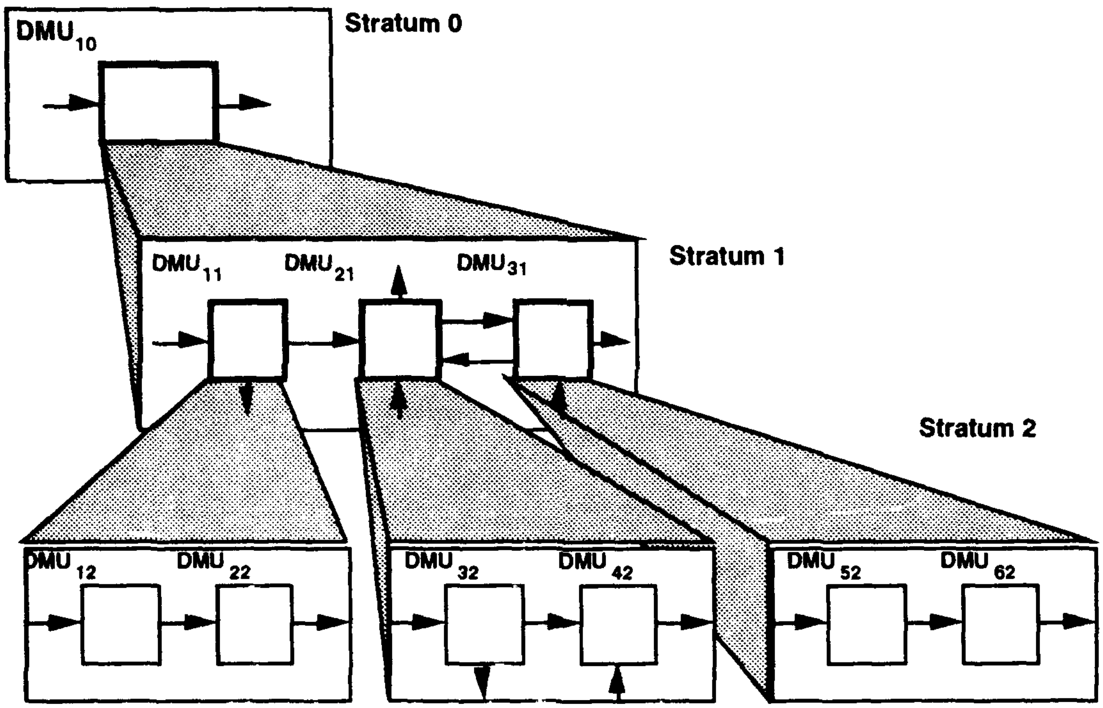


Figure 9.7 A 2-Strata SDMO

The Bottom-Up approach is used to generate the organizational structure. Stratum '1' contains three subsystems, DMU_{11}, DMU_{21}, and DMU_{31} of the organization. The design methodology is applied to the problem in the Bottom-Up manner. The steps of the design procedure are given as follows.

Step 1

- Specification of total number of strata: n = 2
- Specification of the current (lowest) stratum, stratum 'k': k = 2
- Identification of DMUs in stratum 'k' (stratum '2'): DMU_{12}, DMU_{22}, DMU_{32}, DMU_{42}, DMU_{52}, DMU_{62}
- Identification of compound node(s) in stratum 'k-1' (stratum '1'): DMU_{11}, DMU_{21}, DMU_{31}
- Mapping compound node(s) in stratum 'k-1' (stratum '1') to its/their Subsystems in stratum 'k' (stratum '2'): The subsystems in stratum '2' of the compound nodes in stratum '1' are given in Table 9.1.

TABLE 9.1 Compound Nodes in Stratum '1' and their Subsystems in Stratum '2'

Compound Nodes in Stratum '1'	**Subsystems in Stratum '2'**
DMU_{11}	DMU_{12}, DMU_{22}
DMU_{21}	DMU_{32}, DMU_{42}
DMU_{31}	DMU_{52}, DMU_{62}

Step 2

- Generation of organizational structure for compound node DMU_{11} in terms of its subsystems and their mutual interactions: The user-defined constraints for subsystem DMU_{11} are given in Figure 9.8. The Petri Net representation of Σ_{12} and the incidence matrix Δ_{112} of the selected WDN for DMU_{11} are shown in

Figures 9.9 and 9.10. The Universal Net, Kernel Net, MAXOs, MINOs and all the simple paths for Σ_{12} are given in Appendix A.

$$\Sigma_{12} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$$

$$e = [1 \quad x] \qquad F = \begin{bmatrix} 0 & 1 \\ x & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$s = [0 \quad x] \qquad H = \begin{bmatrix} 0 & x \\ x & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Figure 9.8 Matrix Representation of Σ_{12}

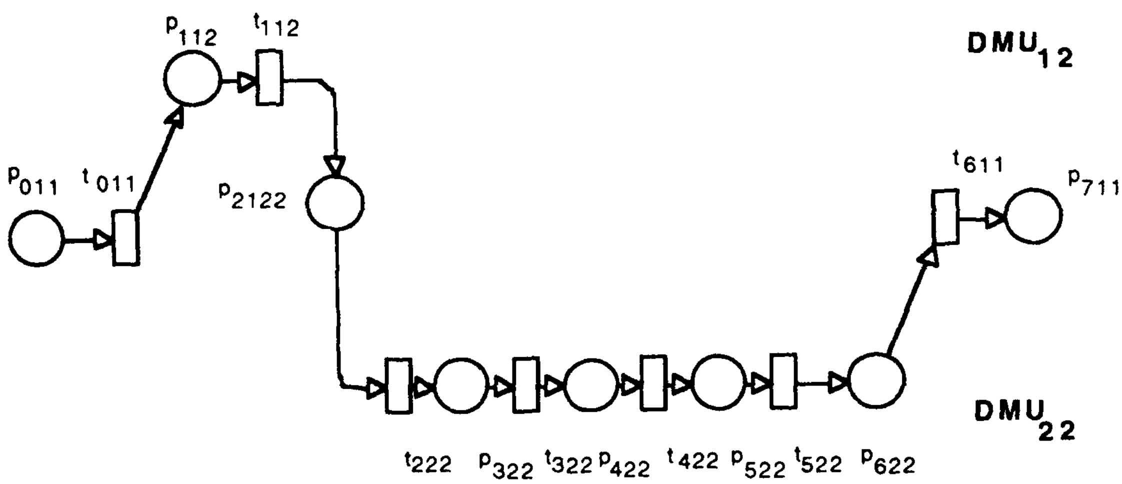


Figure 9.9 Petri Net Representation of the Selected WDN Σ_{12}

$$\Delta_{112} = \begin{array}{c|ccccccc|} & t_{011} & t_{112} & t_{222} & t_{322} & t_{422} & t_{522} & t_{611} \\ p_{011} & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ p_{112} & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ p_{2122} & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ p_{322} & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ p_{422} & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ p_{522} & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ p_{622} & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ p_{711} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Figure 9.10 Incidence Matrix Δ_{112} of Σ_{12}

- Generation of organizational structure for compound node DMU_{21} in terms of its subsystems and their mutual interactions: User-defined constraints for the next subsystem DMU_{21} in stratum '1' are shown in Figure 9.11. The Petri Net representation and the incidence matrix of the selected WDN Σ_{22} are given in Figures 9.12 and 9.13.

$$\Sigma_{22} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$$

$$e = [1 \quad 1] \qquad F = \begin{bmatrix} 0 & x \\ x & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$s = [1 \quad 1] \qquad H = \begin{bmatrix} 0 & x \\ x & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Figure 9.11 Matrix Representation of Σ_{22}

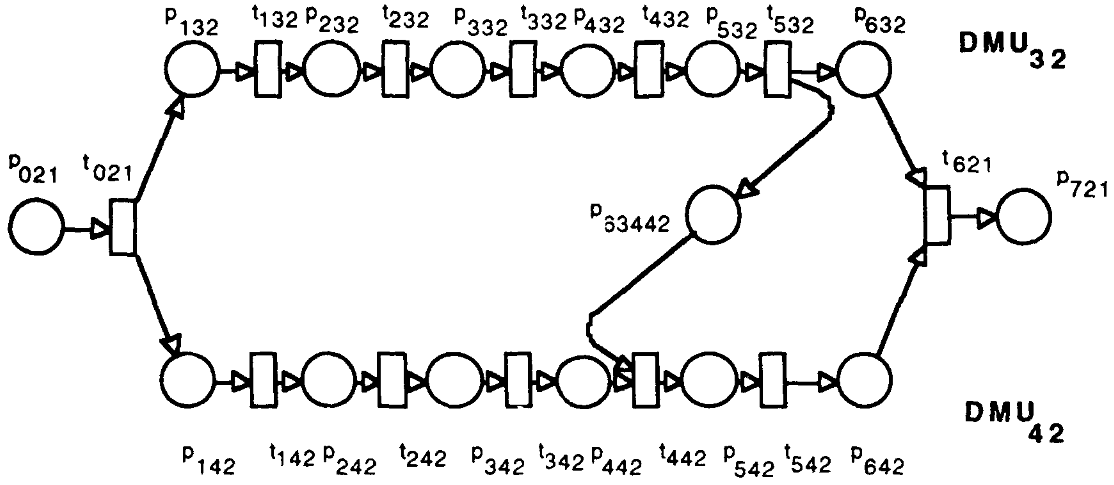


Figure 9.12 Peri Net Representation of Selected WDN Σ_{22}

- Generation of organizational structure for compound node DMU_{31} in terms of its subsystems and their mutual interactions: The last subsystem DMU_{31} in stratum '1' also has two DMUs, DMU_{52} and DMU_{62}, in stratum '2'.The interactional structure desired by the designer between DMU_{52} and DMU_{62} is given in Figure 9.14 in terms of the matrix representation. One of the generated WDNs fulfilling user and structural constraints is presented in Figure 9.15 in terms of the Petri Net representation. The incidence matrix Δ_{32} of the net in Figure 9.15 is given in Figure 9.16.

	t_{021}	t_{132}	t_{142}	t_{232}	t_{242}	t_{332}	t_{342}	t_{432}	t_{442}	t_{532}	t_{542}	t_{621}
p_{021}	-1	0	0	0	0	0	0	0	0	0	0	0
p_{132}	1	-1	0	0	0	0	0	0	0	0	0	0
p_{142}	1	0	-1	0	0	0	0	0		0	0	0
p_{232}	0	1	0	-1	0	0	0	0	0	0	0	0
p_{242}	0	0	1	0	-1	0	0	0	0	0	0	0
p_{332}	0	0	0	1	0	-1	0	0	0	0	0	0
p_{342}	0	0	0	0	1	0	-1	0	0	0	0	0
p_{432}	0	0	0	0	0	1	0	-1		0	0	0
p_{442}	0	0	0	0	0	0	1	0	-1	0	0	0
p_{532}	0	0	0	0	0	0	0	1	0	-1	0	0
p_{542}	0	0	0	0	0	0	0	0	1	0	-1	0
p_{63442}	0	0	0	0	0	0	0	0	-1	1	0	0
p_{632}	0	0	0	0	0	0	0	0	0	1	0	-1
p_{642}	0	0	0	0	0	0	0	0	0	0	1	-1
p_{721}	0	0	0	0	0	0	0	0	0	0	0	1

Figure 9.13 Incidence Matrix Δ_{212} of Σ_{22}

$$\Sigma_{32} = \{ \mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C} \}$$

$$e = [1 \quad x] \qquad F = \begin{bmatrix} 0 & x \\ 0 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & x \\ x & 0 \end{bmatrix}$$

$$s = [x \quad 1] \qquad H = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Figure 9.14 Matrix Representation of Σ_{32}

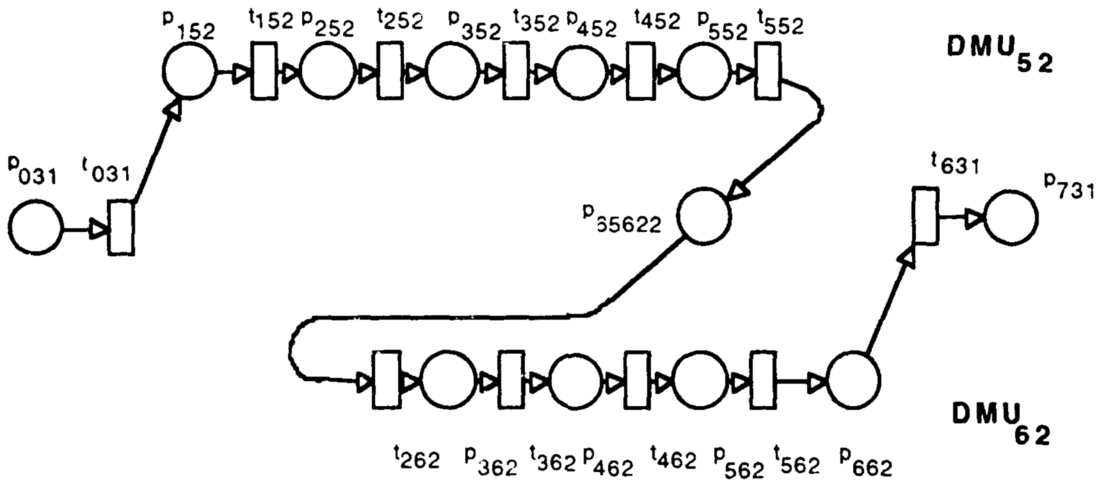


Figure 9.15 Petri Net Representation of Selected WDN Σ_{32}

	t_{031}	t_{152}	t_{252}	t_{262}	t_{352}	t_{362}	t_{452}	t_{462}	t_{552}	t_{562}	t_{631}
p_{031}	-1	0	0	0	0	0	0	0	0	0	0
p_{152}	1	-1	0	0	0	0	0	0	0	0	0
p_{252}	0	1	-1	0	0	0	0	0	0	0	0
p_{352}	0	0	1	0	-1	0	0	0	0	0	0
p_{362}	0	0	0	1	0	-1	0	0	0	0	0
p_{452}	0	0	0	0	1	0	-1	0	0	0	0
p_{462}	0	0	0	0	0	1	0	-1	0	0	0
p_{552}	0	0	0	0	0	0	1	0	-1	0	0
p_{562}	0	0	0	0	0	0	0	1	0	-1	0
p_{65622}	0	0	0	0	0	0	0	-1	1	0	0
p_{662}	0	0	0	0	0	0	0	0	0	1	-1
p_{731}	0	0	0	0	0	0	0	0	0	0	1

Figure 9.16 Incidence Matrix Δ_{312} of Σ_{32}

Step 3

- Specification of next higher stratum: k = 1
- Identification of compound node(s) in stratum 'k-1' (stratum '0'): DMU10
- Mapping compound node(s) in stratum 'k-1' (stratum '0') to its/their Subsystems in stratum 'k' (stratum '1'): All the nodes in stratum '1' are the subsystems of DMU_{10}.

Step 4

- Generation of organizational structure for compound node DMU_{10} in terms of its subsystems and their mutual interactions: The interactional structure as desired by the designer for the subsystems in stratum '1' of the organization is given in Figure 9.17. The Petri Net representation of the entire organization in stratum '1' is presented in Figure 9.18, while the incidence matrix Δ_{101} of the net is shown in Figure 9.19.

$$\Sigma_{11} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$$

$$\mathbf{e} = [1 \quad 1 \quad 1] \qquad \mathbf{F} = \begin{bmatrix} 0 & x & 0 \\ x & 0 & x \\ 1 & 0 & 0 \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{s} = [1 \quad 1 \quad 1] \qquad \mathbf{H} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & x \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 9.17 Matrix Representation of Σ_{11}

Step 5

- Unfolding the organization DMU_{10} to stratum '2': The unfolded organizational structure in stratum '2' is presented in Figure 9.20 after the connectivity rules are applied to the interactional links among DMU_{11}, DMU_{21}, and DMU_{31}. Finally, the unfolded incidence matrix of the organizational structure is given in

Figure 9.21. Again, all the Universal Nets, Kernel Nets, MAXOs, MINOs and all the simple paths for Σ_{11}, Σ_{12}, Σ_{22}, and Σ_{32} are given in Appendix A.

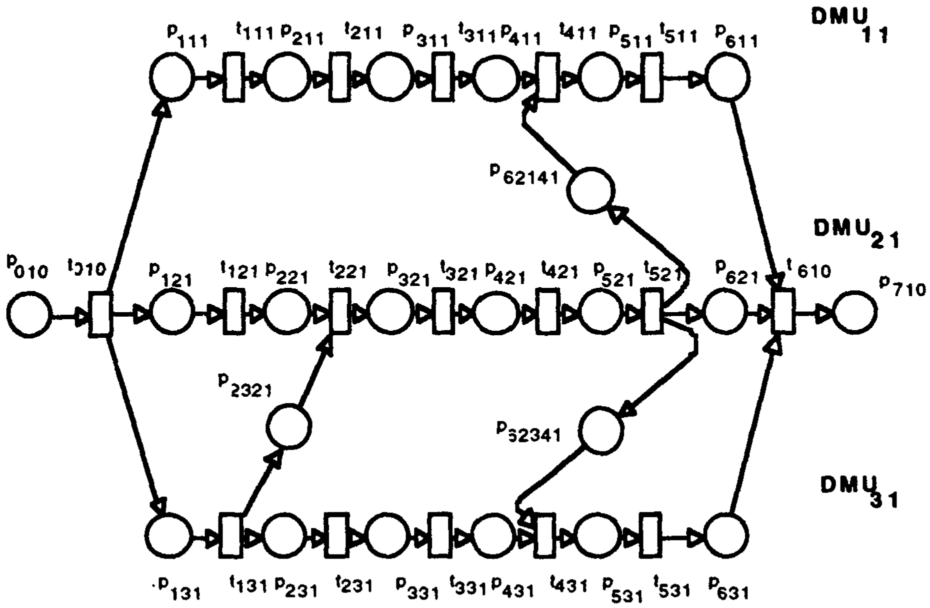


Figure 9.18 Petri Net Representation of Selected WDN Σ_{11}

$$
\Delta_{101} = \begin{bmatrix}
-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&-1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&-1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&-1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&-1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&-1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&-1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&-1\\
0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}
$$

(columns $t_{010} \ldots t_{610}$; rows $p_{010} \ldots p_{710}$)

Figure 9.19 Incidence Matrix Δ_{101} of Σ_{11}

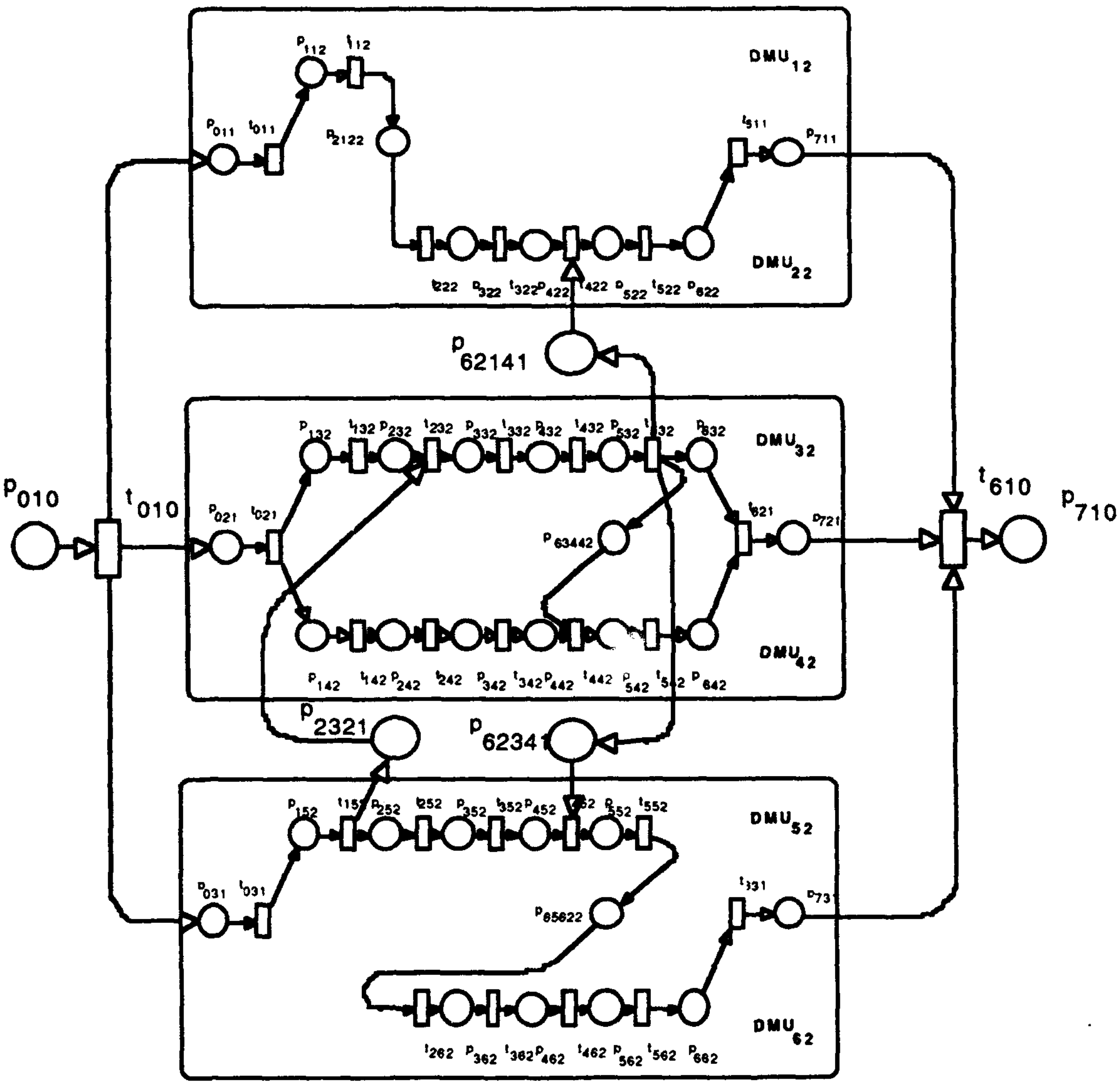


Figure 9.20 Unfolded Organizational Structure in Stratum '2'

9.7 SUBMARINE EMERGENCY CONTROL

In this section, the design methodology is applied to the ship control party of a submarine performing an emergency control task. The description of the problem follows from Weingaertner (1989) who considered a five member decision making organization.

$t_{010} \cdots t_{610}$ (columns); $p_{010} \cdots p_{710}$ (rows)

$$\Delta_{102} = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Figure 9.21 Unfolded Incidence Matrix of the Organizational Structure in Stratum '2'

9.7.1 Overview of Submarine Emergency Control

Submarine emergency control has been broadly defined as those actions taken to counteract the effects of any and all system failures which impede the normal operation of the submarine and the accomplishment of its mission. Although submarines vary, any submarine must be able to submerge to and maintain a commanded depth, maneuver precisely at depth, and rise rapidly to the surface without broaching, in the event of an emergency or in the conduct of its mission. The failures which may befall a submarine range from those of little direct consequences to those threatening catastrophe. They may arise from a variety of sources including design flaws, human error, and battle damage. The gravity of casualties is magnified by the high speed of modern submarines, especially those of the attack classes. The range of operating depths is on the order of only five times the length of the vessel. A distressed vessel may therefore, within tens of seconds, plunge to dangerous depths where the hull may crush, or ascend to and broach the surface, giving away its position and potentially exacerbating the casualty or even colliding with another vessel.

All control decisions, both normal and emergency, are the responsibility of the five member ship control party (SCP). The SCP relies upon several effectors for exercising this control: main and variable ballast tanks for aiding in depth and trim control, external control surfaces (rudder, stern planes, fairwater planes) for controlling trajectory, and naturally, a propeller. (see Figure 9.22).

To detect and diagnose an emergency, the members of the ship control party have available a number of sources of information. Figure 9.23 depicts the SCP positions before the ship and ballast control panels. On the ship control panel are indicators of ship state (speed, depth, heading, trim and roll conditions) and control surface positions displayed with pointer and dial meters and auxiliary plane indications provided by lights located along the dial perimeters. Also on the ship control panel are the control mode buzzer and lights. When electrical power or normal hydraulic power to a set of planes is lost, the control mode shifts automatically from normal mode (electrical-servo control) accompanied by the sounding of the buzzer and the activation of a light corresponding to the affected plane.

The ballast control panel provides information about ship's depth and trim conditions, the status of its ballast tanks and pressurized air banks, as well as information and alarms corresponding to all other vital non-weapon ship systems, e.g., water sensor alarms, gyroscope alarms, and life support system status. The ballast control panel is also equipped with a telephone for communicating with all other ship compartments. This telephone bears reports of flooding casualties.

A final source of information is a loudspeaker providing information about surfaced and submerged sonar contacts and tactical situations which may affect the response to an emergency.

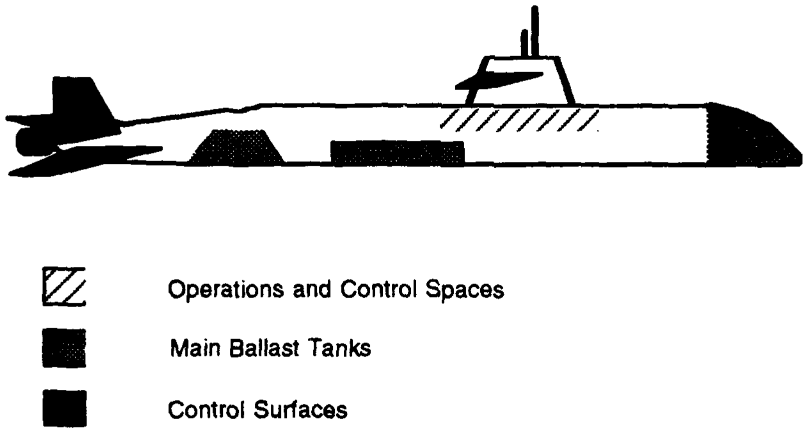


Figure 9.22 Submarine Control Configuration

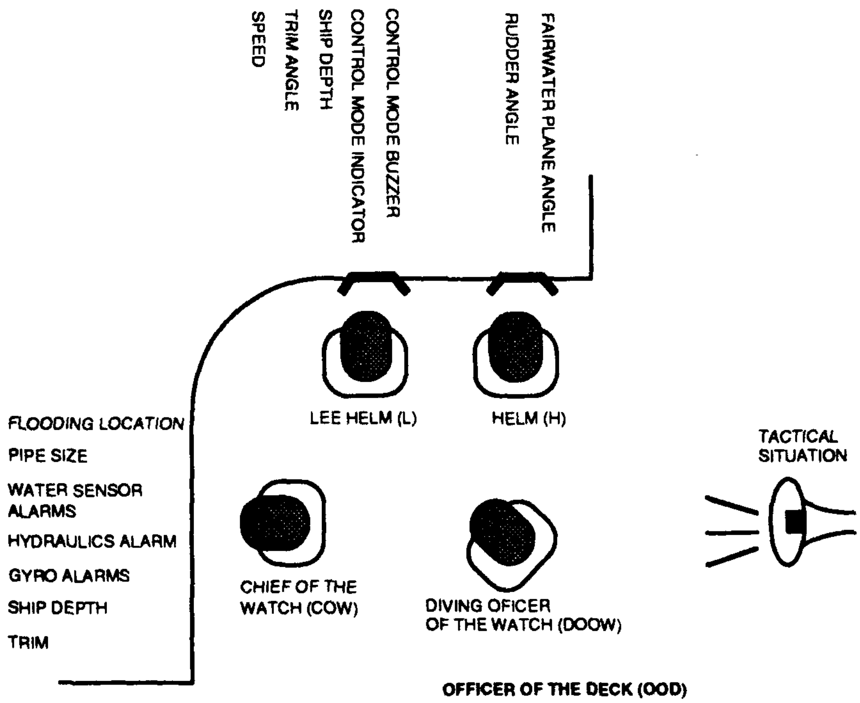


Figure 9.23 The Layout of the Ship Control Party and the Ship Control Panel

9.7.2 Organization Modeling

The ship control party consists of five decision makers: the Officer of the Deck (OOD or O), the Diving Officer of Watch (DOOW or D), the Chief of the Watch (COW or C), the Lee Helm (L), and the Helm (H). The organization has hierarchical and parallel aspects as shown in Figure 9.24. At the top of the structure is the OOD, who has the responsibility for integrating the ship control process with the other aspects of the ship's mission. For emergency control, his job is essentially to decide whether certain aspects of the emergency response should be restricted because of the existence of a sensitive tactical situation. Second in command is the DOOW whose task in the emergency context is to direct and monitor the actions of his subordinates responding to the casualty, subject to any restrictions placed by the OOD. The COW and the helmsmen comprise the bottom tier of the organization, immediately under the DOOW. The COW receives all information on flooding casualties and hydraulic failure, which he shares with the DOOW. He is also in charge of controlling the ship ballast system for aiding in the control of depth. The Lee Helm, L, drives the ship's stern planes, the control surface that modulates the vehicle's trim angle and thus its depth. In performing this task, L receives information about the plane angle and the control mode (see Weingaertner, 1989) as well as ship state information. Finally, the Helm, H, controls the ship's rudder and fairwater planes based on plane angle information, control mode, and ship state information - the same information that is available to L.

SCP as a Stratified Decision Making Organization

Figure 9.25 presents the SCP as a Stratified Decision Making Organization (SDMO), where DOOW, COW, L, and H comprise a suborganization in stratum '2'. The OOD is taken as another subsystem in stratum '1'. The interactions between the OOD and the rest of the SCP are defined in stratum '1' as interactions between the OOD and the compound node representing the suborganization of DOOW, COW, L, and H in stratum '2'. The unfolded organizational structure in stratum '2' of the node in stratum '0' will show the detailed interactional structure of the SCP.

The design methodology is now applied to the SDMO in Figure 9.25. The members of the SCP, OOD, DOOW, COW, L, and H are denoted by DMU_{52}, DMU_{12}, DMU_{32}, and DMU_{42} respectively. The compound node representing the suborganization consisting of DOOW, COW, L, and H is denoted by DMU_{11} in stratum '1', while the compound node representation of OOD is denoted by DMU_{21}. The compound node in stratum '0' representing the SCP is denoted by DMU_{10}.

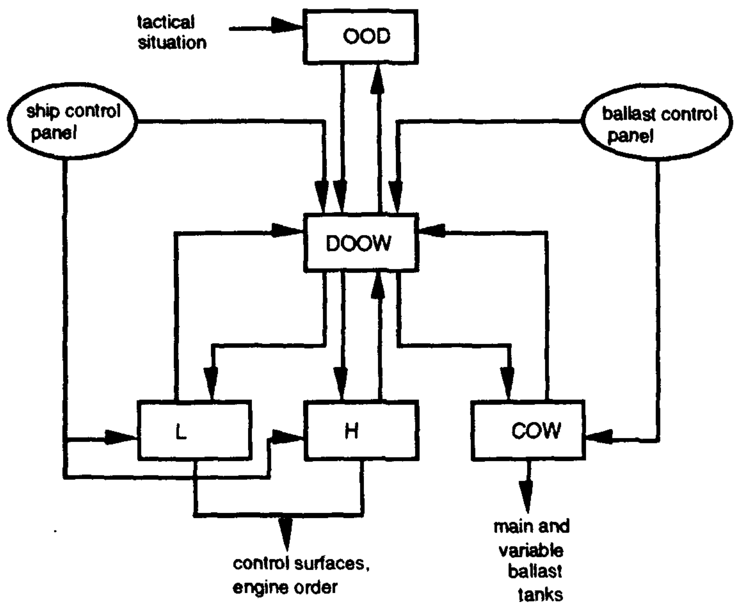


Figure 9.24 The Ship Control Party

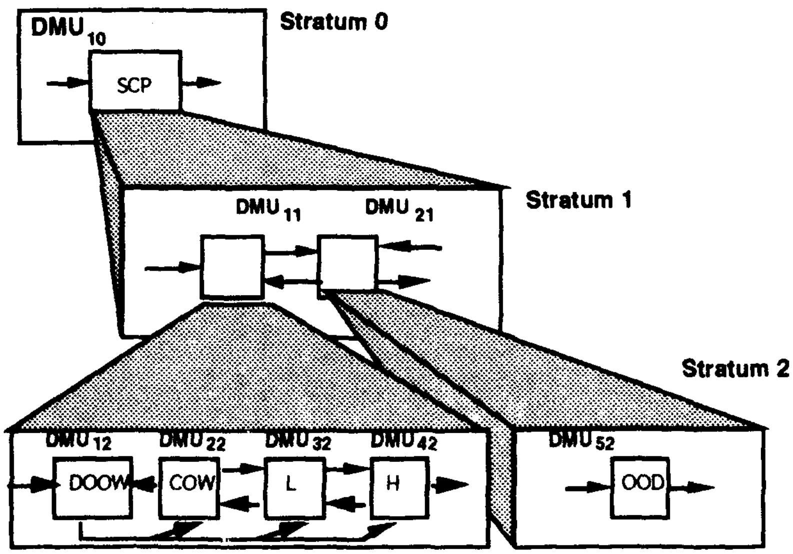


Figure 9.25 SCP as a 2-Strata SDMO

Application of the Methodology

The Bottom-Up approach is adopted to generate the organizational structures. Stratum '1' contains two subsystems, DMU_{11}, and DMU_{21}. The steps of the design procedure are given as follows.

Step 1

- N = 2
- Specification of total number of strata: n = 2
- Specification of the current (lowest) stratum, stratum 'k': k = 2
- Identification of DMUs in stratum 'k' (stratum '2'):

  DMU_{12}, DMU_{22}, DMU_{32}, DMU_{42}, DMU_{52}

  These DMUs are actually DMs.
- Identification of compound node(s) in stratum 'k-1' (stratum '1'): DMU_{11}, DMU_{21}
- Mapping compound node(s) in *stratum* 'k-1' (*stratum* '1') to its/their Subsystems in stratum 'k' (stratum '2'): The subsystems in stratum '2' of the compound nodes in stratum '1' are given in Table 9.2.

TABLE 9.2 Compound Nodes in Stratum '1' and their Subsystems in Stratum '2'.

Compound Nodes in Stratum '1'	**Subsystems in Stratum '2'**
DMU_{11}	DMU_{12}, DMU_{22} DMU_{32}, DMU_{42}
DMU_{21}	DMU_{52}

Step 2

- Generation of organizational structure for compound node DMU_{11} in terms of its subsystems and their mutual interactions: The user-defined constraints for subsystem DMU_{11} are given in Figure 9.26. The user-defined constraints are derived from the discussion in Section 9.7.2 and Figure 9.24. A special constraint has also been implemented by a inserting an interactional link between the SA stages of COW and DOOW, as this link represents the situation where the DOOW's SA stage selects an appropriate algorithm for filtering out the extraneous information from the external inputs by using assessed information from COW. The special constraint is implemented by the defining an interactional place P_{22112}. The Petri Net representation of Σ_{12} and the incidence matrix Δ_{112} of the selected WDN for DMU_{11} are shown in Figures 9.27 and 9.28. The MAXOs, MINOs and all the simple paths for Σ_{12} are given in Appendix A.

$$\Sigma_{12} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$$

$$e = [x \quad 1 \quad 1 \quad 1] \qquad F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & x \\ 1 & 0 & x & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$s = [0 \quad 1 \quad 1 \quad 1] \qquad H = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 9.26 Matrix Representation of Σ_{12}

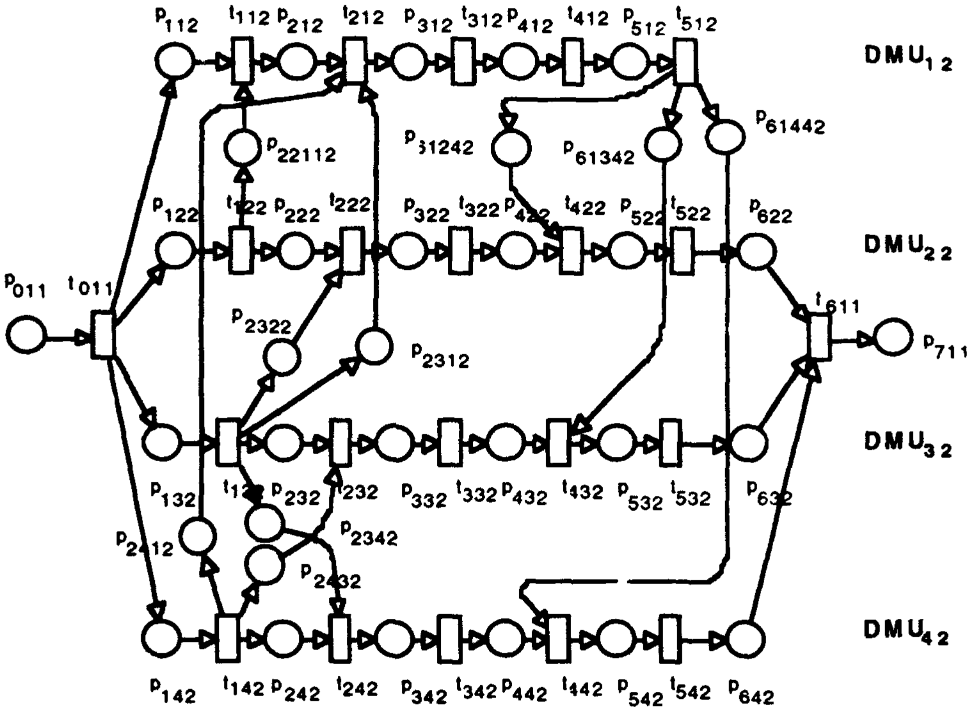


Figure 9.27 Petri Net Representation of the Selected WDN Σ_{12}

- Generation of organizational structure for compound node DMU_{21} in terms of its subsystems and their mutual interactions. As there is only one DM in stratum '2', the structure of DMU_{21} will trivially be represented by the structure of DMU_{52}.

Step 3

- Specification of next higher stratum: k = 1
- Identification of compound node(s) in stratum 'k-1' (stratum '0'): DMU10
- Mapping compound node(s) in stratum 'k-1' (stratum '0') to its/their Subsystems in stratum 'k' (stratum '1'): All the nodes in stratum '1' are the subsystems of DMU_{10}.

$t_{011} \cdots \qquad\qquad \cdots t_{611}$

$$
\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{matrix} p_{011} \\ \vdots \\ \\ \vdots \\ p_{711} \end{matrix}
$$

Figure 9.28 Incidence Matrix Δ_{112} of Σ_{12}

Step 4

- Generation of organizational structure for compound node DMU_{10} in terms of its subsystems and their mutual interactions: The interactional structure as described in Section 10.2 for the subsystems in stratum '1' of the organization is given in Figure 9.29. The Petri Net representation of the entire organization in stratum '1' is presented in Figure 9.30, while the incidence matrix Δ_{101} of the net is shown in Figure 9.31.

$$\Sigma_{11} = \{\mathbf{e}, \mathbf{s}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{C}\}$$

$$\mathrm{e} = [1 \quad 1] \qquad \mathrm{F} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \mathrm{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathrm{s} = [1 \quad 1] \qquad \mathrm{H} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \mathrm{C} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

Figure 9.29 Matrix Representation of Σ_{11}

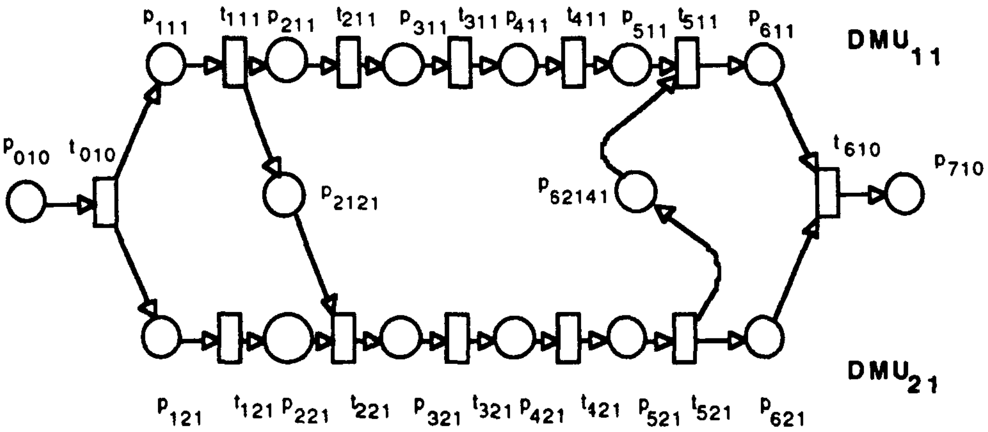


Figure 9.30 Petri Net Representation of Selected WDN Σ_{11}

	t_{010}	t_{111}	t_{121}	t_{211}	t_{221}	t_{311}	t_{321}	t_{411}	t_{421}	t_{511}	t_{521}	t_{610}
p_{010}	-1	0	0	0	0	0	0	0	0	0	0	0
p_{111}	1	-1	0	0	0	0	0	0	0	0	0	0
p_{121}	1	0	-1	0	0	0	0	0	0	0	0	0
p_{211}	0	1	0	-1	0	0	0	0	0	0	0	0
p_{2121}	0	1	0	0	-1	0	0	0	0	0	0	0
p_{221}	0	0	1	0	-1	0	0	0	0	0	0	0
p_{311}	0	0	0	1	0	-1	0	0	0	0	0	0
p_{321}	0	0	0	0	1	0	-1	0	0	0	0	0
p_{411}	0	0	0	0	0	1	0	-1	0	0	0	0
p_{421}	0	0	0	0	0	0	1	0	-1	0	0	0
p_{511}	0	0	0	0	0	0	0	1	0	-1	0	0
p_{521}	0	0	0	0	0	0	0	0	1	0	-1	0
p_{611}	0	0	0	0	0	0	0	0	0	1	0	-1
p_{621}	0	0	0	0	0	0	0	0	0	0	1	-1
p_{62141}	0	0	0	0	0	0	0	0	0	-1	1	0
p_{710}	0	0	0	0	0	0	0	0	0	0	0	1

Figure 9.31 Incidence Matrix Δ_{101} of Σ_{11}

Step 5

- Unfolding the organization DMU_{10} to stratum '2': The unfolded organizational structure in stratum '2' is presented in Figure 9.32 after the connectivity rules are applied to the interactional links among DMU_{11}, and DMU_{21}. The interactional links between the compound nodes in stratum '1' are translated in terms of interactions among their subsystems in stratum '2'. Finally, the unfolded incidence matrix of the organizational structure is given in Figure 9.33. Again, all the Universal Nets, Kernel Nets, MAXOs, MINOs and all the simple paths for Σ_{11}, and Σ_{12} are given in Appendix A.

9.8 CONCLUSION

The entire algorithm for generating Stratified Decision Making Organizations (SDMO) was presented. The two approaches, Top-Down and Bottom-Up, were emphasized and a comparative study of the two approaches was done. The two approaches mentioned produce identical results if applied to a particular problem but differ in the manner in which the system's architecture evolves under the two design approaches. The proposed algorithm was applied to to two illustrative examples with all the steps of the design procedure explicitly outlined.

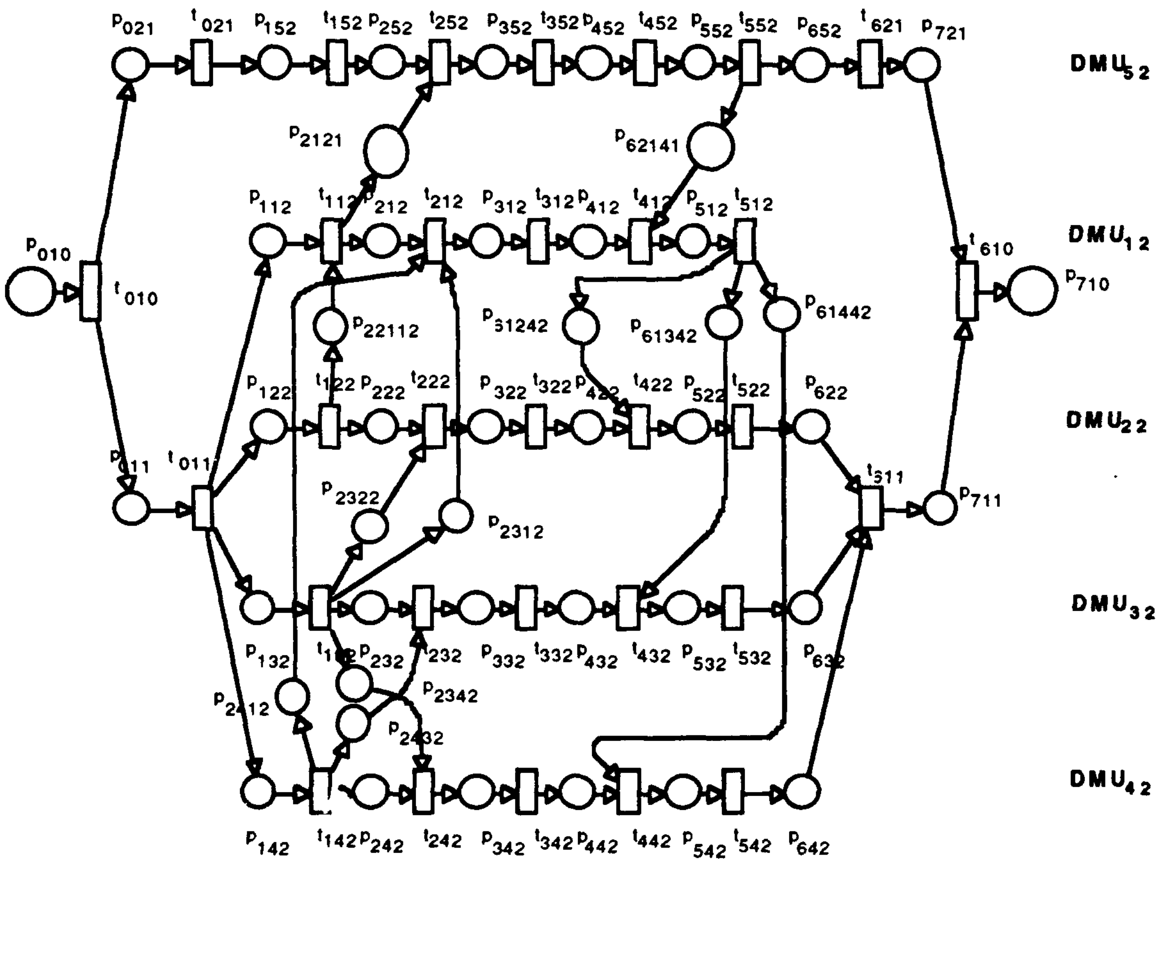


Figure 9.32 Unfolded Organizational Structure of SCP in Stratum '2'

$t_{010} \cdots \quad \cdots \ t_{610}$

$$
\begin{bmatrix}
-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}
\begin{matrix} p_{010} \\ \vdots \\ \\ \vdots \\ p_{710} \end{matrix}
$$

Figure 9.33 Unfolded Incidence Matrix of the Organizational Structure of SCP in Stratum '2'

CHAPTER X

CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

10.1 CONCLUSIONS

This thesis described a methodology to model and generate multilevel hierarchical distributed systems. A system which is comprised of a vertical arrangement of its subsystems is considered as a multilevel hierarchical system. The higher level subsystems of such a system are characterized with their priority of action and right of intervention. The performance of the system, however, depends upon the lower level subsystems. The type of multilevel hierarchical systems modeled in this thesis are *stratified systems,* where the system is described by a family of structures each concerned with the behavior of the system as viewed from a different level of abstraction, the *stratum*.. The basic concepts of multilevel hierarchical organizational structures are presented in Chapter IV. The objects from which the system may be built are also presented in Chapter IV. In this model, a decision making process described by a five stage process, and interactions among these decision making processes are identified. It is shown that the organizational structure of a decision making organization or sub-organization can be represented in matrix form. The notion of a *Compound Node* is introduced to formalize the concept of higher level description of organizational or sub-organizational structures. The decision making entities in the least abstracted description of an organization are human decision makers (DMs).

In Chapter V, the methodology is formulated using the language of Hierarchical Petri Net theory. The concept of having a family of organizational structures for a system where each member of the family describes the system's behavior at a different degree of abstraction is realized by *folding* and *unfolding* the organizational structures. The processes of folding and unfolding allow one to have a description of an organization at any desired degree of abstraction. The two processes generate all the relevant descriptions of a system's architecture. Folding process yields organizational structures with lower degree of detail, while unfolding results in more elaborated description of an organization. The processes of folding and unfolding an organizational structure are translated into both matrix and Hierarchical Petri Net representations.

In Chapter VI, the class of structures that must be considered, given a design problem, is described. The structural and user-defined constraints are identified for different levels of abstraction used to describe the organization. A set of modified structural constraints have been imposed to define the set of organizational structures that make physical sense. The set of user-defined constraints is introduced to allow the designer of a system to translate his knowledge of the specific application into the formalism of the design procedure. The notion of convexity is used to analyze the properties of different constraints.

The problem of interpreting higher level interactions in lower levels arises when an organizational structure is unfolded to its lower level description. Chapter VII addresses this issue of connectivity for the higher level interactions when defined at lower levels. A scheme to identify the echelon type hierarchical relationship among DMUs is presented and a *set of connectivity* rules is formulated on the basis of multiechelon hierarchy present in the system's architecture. The connectivity rules are used to translate interactions among subsystems of the organization defined at a given level to their lower level representations.

In Chapter VIII, a review of the Lattice algorithm is presented. The set of all allowable organizational structures of the given system or subsystem is characterized by its boundaries. Lattice theoretic results are used to define a partial order among all allowable organizational structures belonging to a system or a subsystem. The process includes the application of the Lattice Algorithm (Remy, 1986) iteratively at different levels in the organization with redefined structural requirements for the particular levels and for the particular system or subsystem under consideration.

The algorithmic implementation of the overall methodology is presented in Chapter IX. An appropriate user interface is defined. It allows the designer to go step by step through the entire design methodology. Two simple examples illustrate the design procedure. It is seen that by defining a team of organizational members in a multilevel environment and then generating the organizational structure has resulted in a substantial reduction of computational effort which is required in generating all the feasible solutions.

The methodology provides a structured and a modular way of solving a design problem. An organization with hundreds of lower level subsystems can be modeled with lesser computational effort by carefully defining the higher level subsystems of the organization in terms of the lower level subsystems. The entire organization can then be modeled only in terms of the higher level subsystems. The higher level subsystems are modeled in terms of the lower level subsystems. Finally, all the structures are integrated to produce a family of structures for the organization each describing the organization at different degree of detail.

10.2 DIRECTIONS FOR FURTHER RESEARCH

Research can be pursued in many directions to improve and extend the methodology developed in this thesis.

- The hierarchical relationship among Decision Making Units (DMUs) is established by investigating the input and output interactions of each DMU. There is a need of an algorithm which identifies the hierarchical relationship among DMUs by looking at pattens in the arrays or incidence matrix defining an organization.

- A natural extension to the current effort would be to achieve a more relaxed set of connectivity rules from those presented in Chapter VII. Especially, the translation of a higher stratum interactional link into a single interactional link at a lower stratum should be relaxed to multiple interactional links in the lower stratum representation of the organization. This would also require a careful investigation of the set of connectivity rules and it seems that an extended set of connectivity rules will be achieved as a result. A situation where an executive broadcasts his commands to all of his subordinates can be easily modeled in the extended version.

- The present model should include new protocols of interactions between DMUs. The present model permits only three interactional links between two DMUs. This paucity of interactions prevents the DMUs from having elaborate protocols of interactions at several stages of their processing. For example, as mentioned by Demaël (1989), if A has already received a message from B, it can not ask B to send more information. New interactions should be defined and be interpreted in terms of their physical relevance. The connectivity rules presented in Chapter VII can then be modified and made an integral part of the design methodology.

- It would be particularly interesting to work on a methodology for modeling and generating variable-structure multilevel hierarchical distributed intelligence systems. In this kind of structures, not only the interactions among DMUs in a given stratum will change according to the input they process, but the interactional structure of their subsystems in lower strata will also change. This can be achieved by using the Stochastic Timed Colored Petri Net formalism.

- Lastly, coordination issues in a multilevel hierarchical environment have not been addressed yet. There is need for an analytical model to measure the *coordination* among subsystems of an organization defined at different levels of abstraction, and the system's performance against all possible coordination strategies used in the organizational structure.

REFERENCES

American National Standard Institute, X3.5-1970, "Flowchart Symbols and their Usage in Information Processing."

Andreadakis, S. K., and Levis A. H. (1988) "Synthesis of Distributed Command and Control for the Outer Air Battle" *Proc. 1988 Symposium on Command and Control Research.* Science Applications International Corporation, McLean, Virginia.

Balbes, R. and Dwinger, P. (1974). *Distributive Lattices* University of Missouri Press, Columbia, Missouri.

Birkhoff, G. (1948). *Lattice Theory.* American Mathematical Society, Providence, RI.

Boettcher, K. L. and Levis, A. H. (1982). "Modeling the Interacting Decisionmaker with Bounded Rationality." *IEEE Trans. on Systems, Man, and Cybernetics,* SMC-12, No.3.

Brams, G. W. (1983). *Réseaux de Petri: Théorie et Pratique Tomes 1 et 2.* Masson, Paris.

Carré, B. (1979). *Graphs and Networks.* Oxford University Press, New York.

Demaël, J. J (1989). "On the Generation of Variable Structure Distributed Architectures." LIDS-TH-1869, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Demaël, J. J. and Levis, A. H. (1990). "On the generation of a Variable Structure Airport Surface Traffic Control System." *Proc. IFAC World Congress,* Tallinn, Estonia, USSR.

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

Grätzer, G. (1971). *Lattice Theory, First Concepts and Distributed Lattices.* W. H. Freeman and Company, San Francisco.

Hall, S. A. (1982). "Information Theoretic Models of Storage and Memory.", LIDS-TH-1232, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Hillion, H. P. (1986) "Performance Evaluation of Decisionmaking Organizations using Timed Petri Nets." LIDS-TH-1590, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Jensen, K. (1987). "Colored Petri Nets" *in Advanced Course on Petri Nets 1986, Lecture Notes in Computer Science*, Springer Verlag, Berlin, Germany.

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, England.

Levis, A. H. (to appear in 1992). "Modeling and Design of Distributed Intelligence Systems" in *Introduction to Autonomous and Intelligent Control,* P. J. Antsaklis and K. M. Passino, Eds., Kluwer Publishers, Boston, MA.

Mesarovic, M. D., Macko, D., and Takahara, Y. (1970). *Theory of Hierarchical, Multilevel Systems.* Academic Press, New York.

Minsky, M. (1986). *The Society of Mind.* Simon and Schuster, New York.

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

Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems,* Prentice Hall, Englewood Cliffs, NJ.

Reisig, W. (1985). *Petri Nets, An Introduction.* Springer Verlag, Berlin, Germany.

Remy, P. (1986). "On the Generation of Organizational Architectures Using Petri Nets." LIDS-TH-1630, Laboratory for Information Decision Systems, MIT, Cambridge, MA.

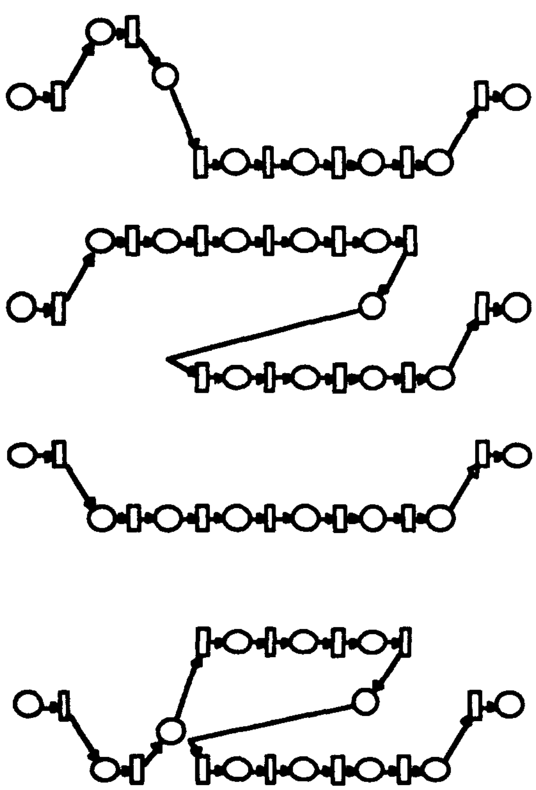
Remy, P. and Levis, A. H. (1988). "On the Generation of Organizational Architectures Using Petri Nets." in *Advances in Petri Nets 1988, Lecture Notes in Computer Science,* G.Rozenberg Ed. Springer Verlag, Berlin, Germany.

Stabile, D. A. and Levis, A. H. (1984) "The Design of Information Structures: Basic Allocations Strategies for Organizations." in *Large Scale Systems,* Vol.6, No 2, North-Holland.

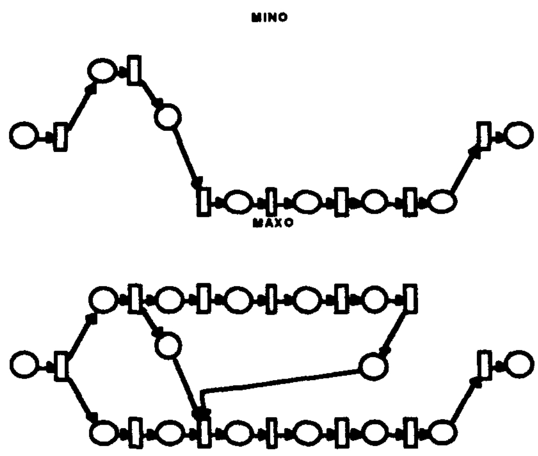
Weingaertner, S. T., and A. H. Levis, 1989, "Evaluation of Decision Aiding in Submarine Emergency Decisionmaking" *Automatica,* Vol. 25, No 3.

APPENDIX A

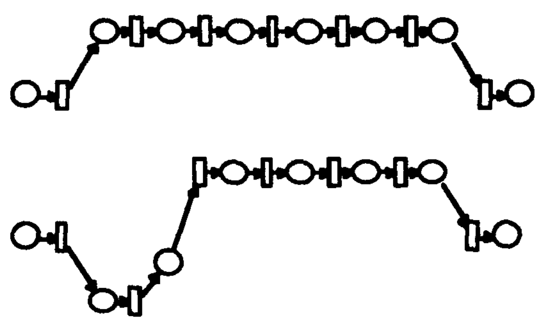
This appendix contains the simple paths, MAXOs and MINIOs for the WDNs presented in the two illustrative examples in Chapter IX. The simple paths, MAXOs and MINOs listed in this appendix are produced with the help of *ARCGEN*, a software application which implements the lattice algorithm. Figures A-1 to A-10 list the nets belonging to the first example with six DMUs in stratum '3' DMUs in stratum '1'. The rest of the figures belong to the second example in which the organizational structure of the ship control party of a submarine is modeled.



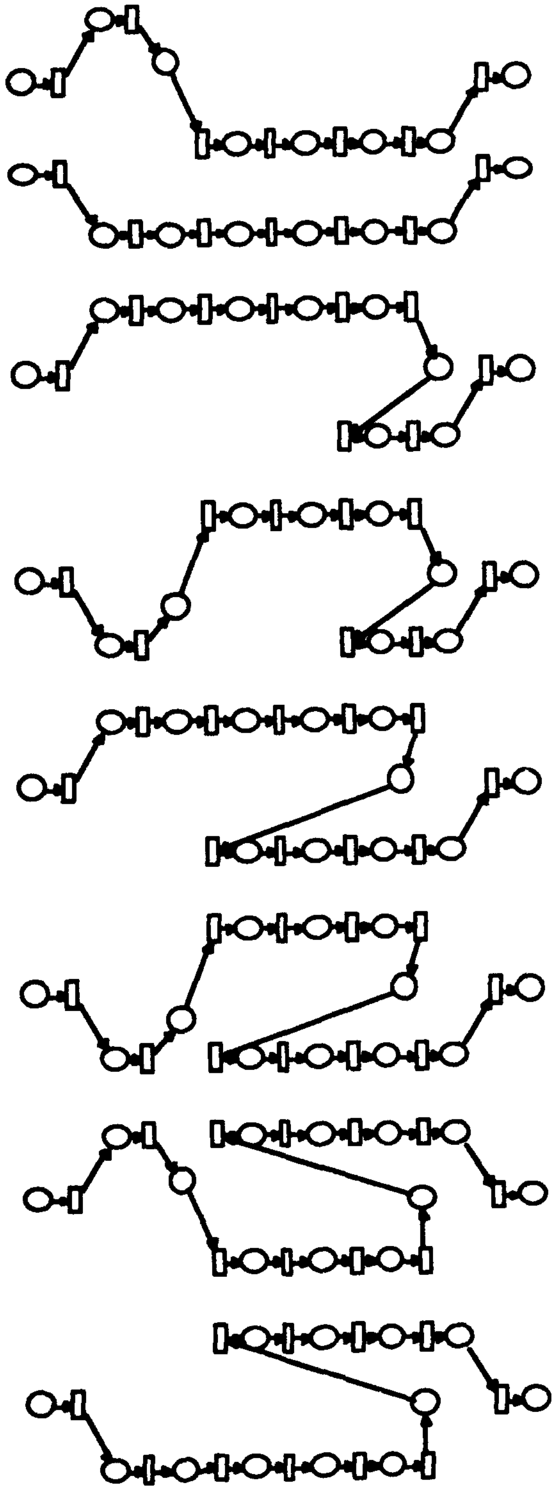
A-1 Simple Paths of Σ_{12} in Chapter IX



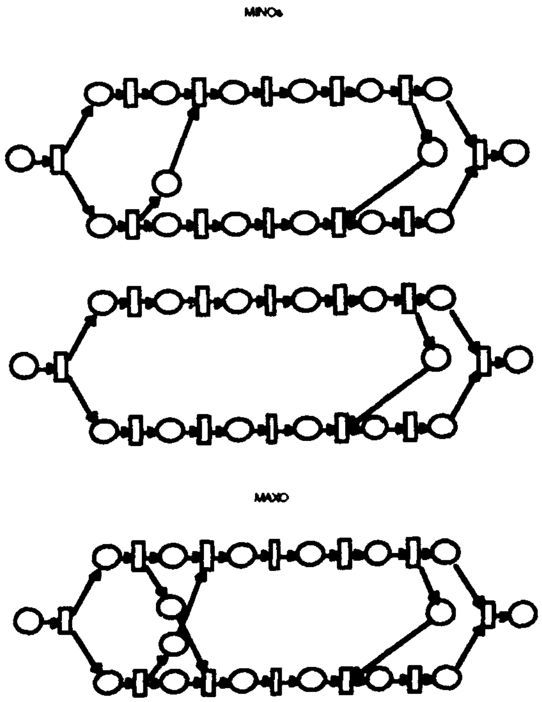
A-2 MINO and MAXO of Σ_{12} in Chaptr IX



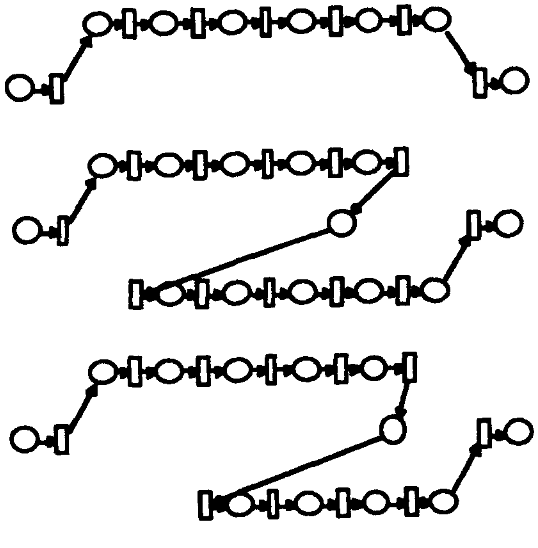
A-3 Simple Paths of Σ_{22} in Chapter IX



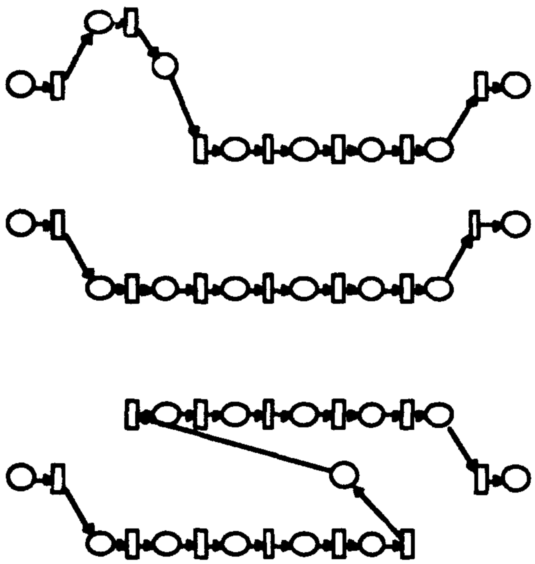
A-3 Simple Paths of Σ_{22} Continued



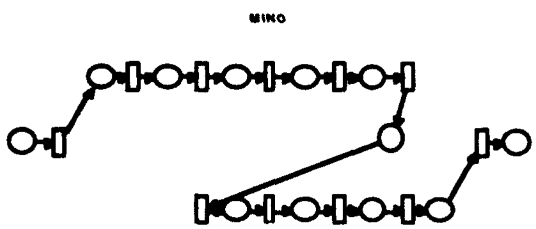
A-4 MINOs and MAXO of Σ_{22} in Chapter IX



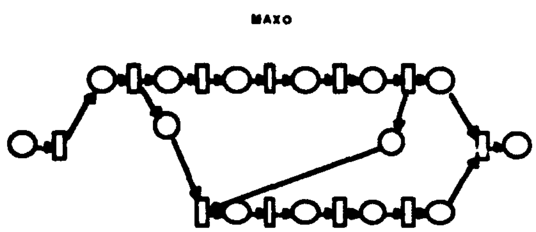
A-5 Simple Paths of $\Sigma 32$ in Chapter IX



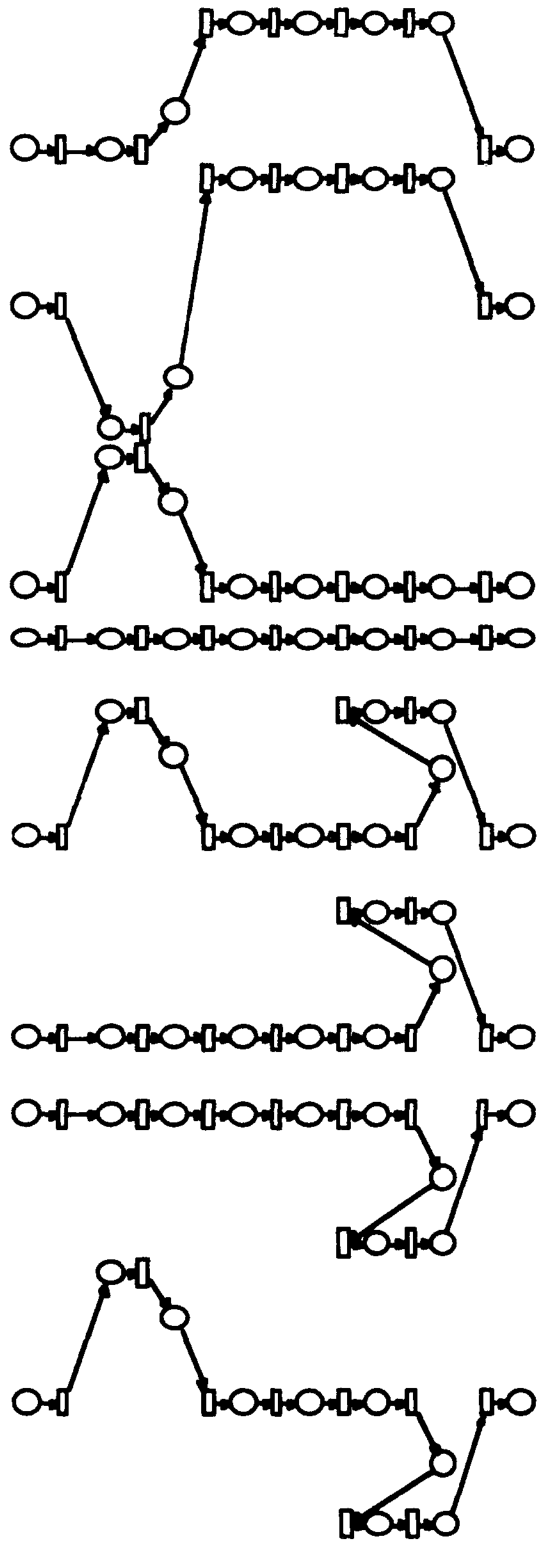
A-5 Simple Paths of Σ_{32} in Chapter IX Continued



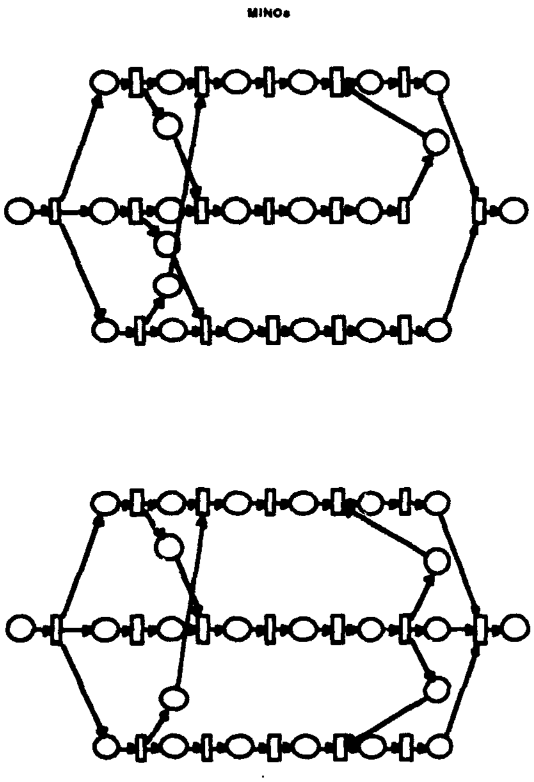
A-6 MINO of Σ_{32} in Chapter IX



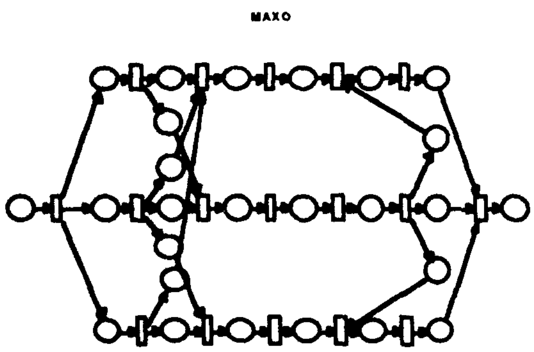
A-7 MAXO of Σ_{32} in Chapter IX



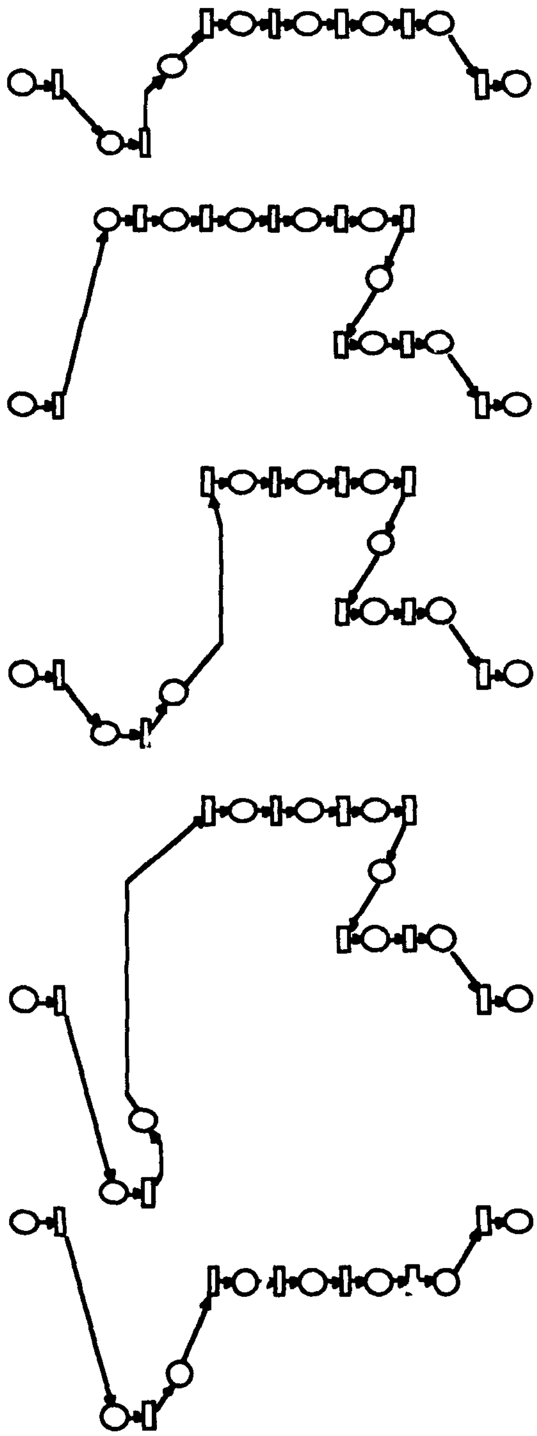
A-8 Simple Paths of Σ_{11} in Chapter IX



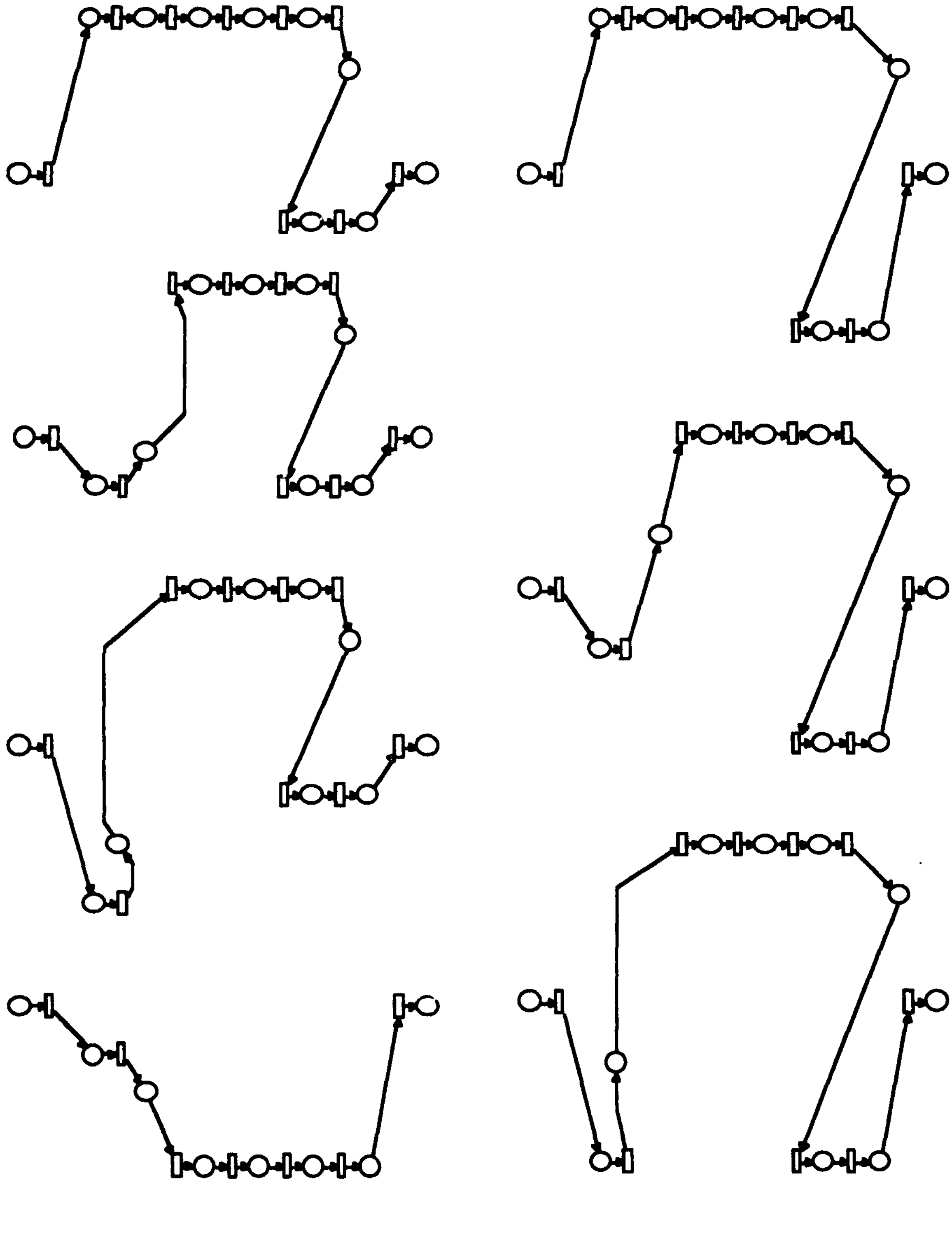
A-9 MINOs of Σ_{11} in Chapter IX



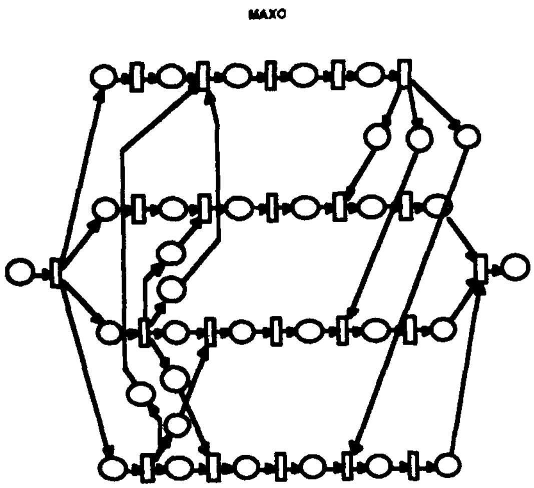
A-10 MAXO of Σ_{11} in Chapter IX



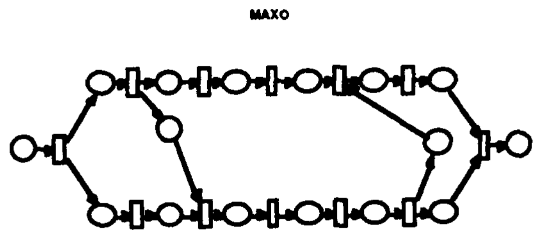
A-11 Simple Paths of Σ_{12} in Chapter IX (SCP)



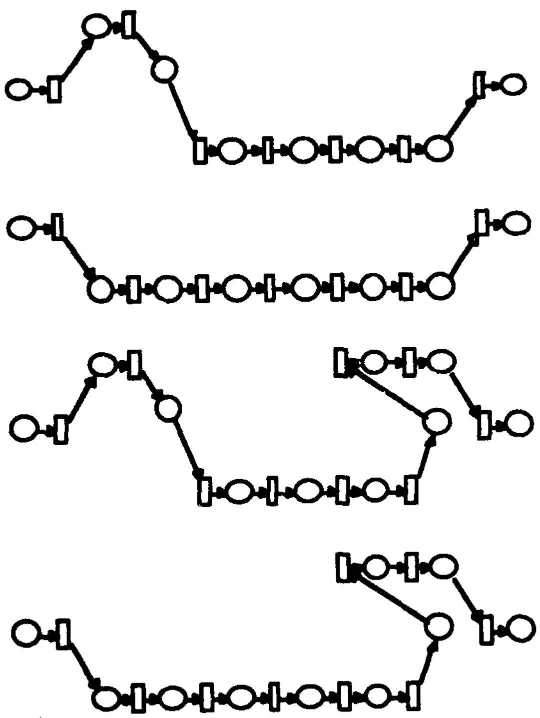
A-11 Simple Paths Continued



A-12 MAXO of Σ_{12} in Chapter IX (SCP)



A-14 MAXO/MINO of Σ_{11} in Chapter IX (SCP)



A-13 Simple Paths of Σ_{11} in Chapter IX (SCP)

APPENDIX B

PROCEDURES FOR FOLDING AND UNFOLDING THE ORGANIZATIONAL STRUCTURES USING *DESIGN CPN™*

INTRODUCTION

In order to carry out a folding and/or unfolding procedure one has to be familiar with the software *Design/CPN™*. In the following discussion, only those procedures and steps are outlined that are used to fold and unfold a particular organizational structure given at a specified stratum. The reader is, therefore, requested to go through the manuals of the software, in case, he or she is not familiar with the software itself. The following discussion might lead to a non-executable Petri Net representation of the organization if the instructions in the above mentioned manuals are not followed carefully. Note, the software implementation discussed in this Appendix is valid for **Macintosh™** and **Sun Microsystems** implementations of the *Design/CPN™* released by **Meta Software Corporation**.

The processes of folding and unfolding are presented with the help of an example Petri Net model of an organization defined in stratum '2'. One has to have a Petri Net representation of an organization in a given stratum in order to fold it to the strata above the current one. The folded structures can be unfolded up to the stratum at which the Petri Net representation is defined. As a result, one can not unfold an organizational structure below the one in which the structure of the organization has been constructed. In order to have an even lower - if theory permits - stratum representation, the designer has to construct the Petri Net representation for the required stratum with the help of the incidence matrix constructed as a result of the methodology. It is now made clear that the software does not support the methodology for generating multilevel hierarchical organizational structures as a result of the algorithm presented in Chapter IX. Rather, it supports the graphical (Petri Net) representation of the organization structures generated analytically by the methodology. It is, however, important to note that once an organizational structure is represented in terms of its Petri Net model, the Petri Net structures can be folded and unfolded in accordance to the theory developed in the previous chapters. The hierarchical nature of the software makes it a perfect choice for representing organizational structures and their folded and unfolded versions.

FOLDING

Figure B-1 represents a Petri Net model of an organization in stratum '2' with four DMUs defined in the stratum. The place and transition labels are suppressed for the sake of simplicity, only the labels associated with interactional places are shown as they play a key role in deciding whether or not a particular place should be folded at a given stratum.

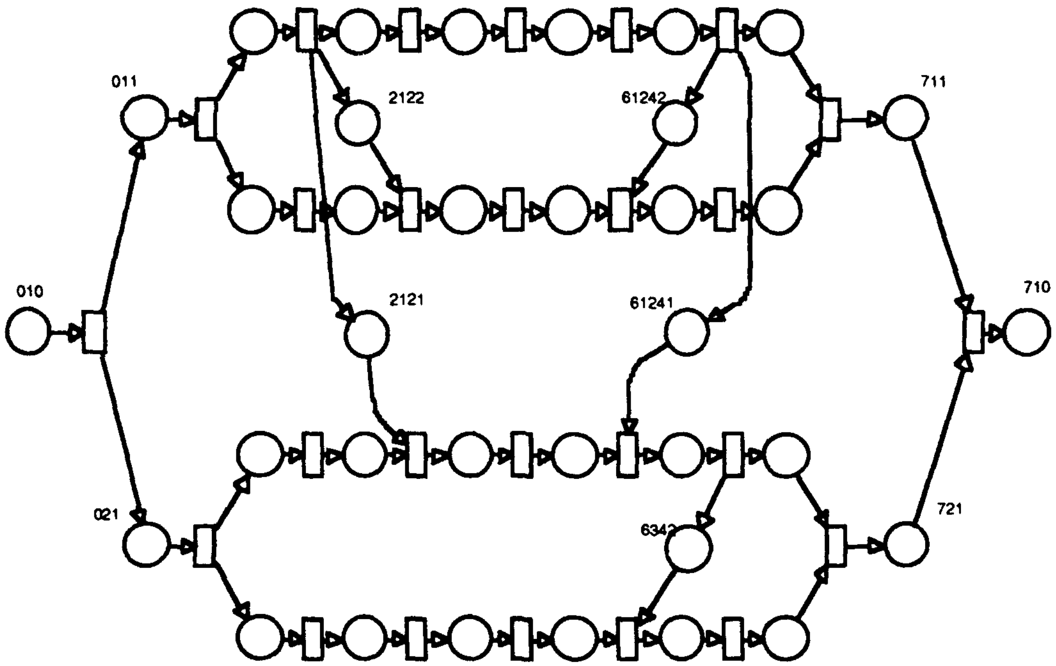


Figure B-1

In Figure B-2, the shaded rounded boxes represents the boundaries of the subnets that are to be replaced by compound transitions as a result of folding the structure in stratum '2' to obtain the structure in stratum '1'. All the nodes inside a particular box are selected together by invoking the command **Enter Group Mode** in the **Group** menu.

Once a correct group is formed, the command **Move to Subpage ...** is invoked in the **CPN** menu. As a result, an option window will appear. Click **OK**. A compound transition labelled with **'HS'** will replace the selected nodes. The said compound transition can be placed at a desired location by pushing the **shift** button and dragging the transition by mouse movement. All the subnets outlined by the boxes are treated in this manner sequentially.

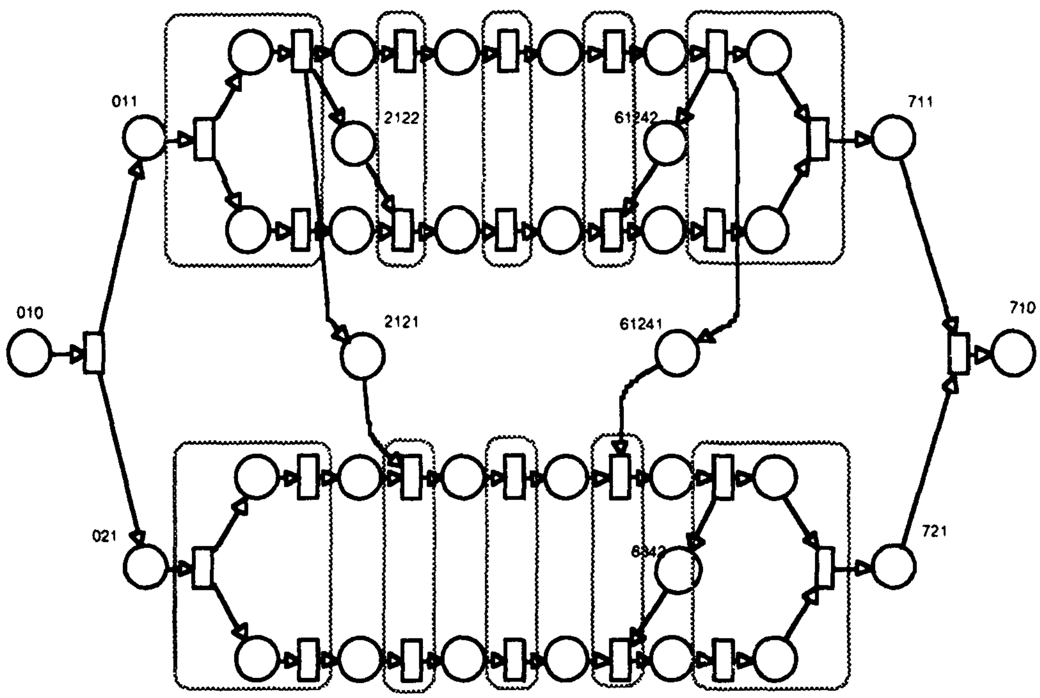


Figure B-2

If all the subnets are replaced by their compound transition representation, one should get a structure shown in Figure B-3. Following the discussion in Subsections 2.3.4 and 5.7.3, the redundant places between any two transitions are omitted. Similarly, the interactional places that are defied in stratum '2' are also deleted as they have no representation in stratum '1'. Note that the place labels shown in the figures will help in deciding which ones are to be deleted. The justification of the statement made is given in Subsection 5.7.3.

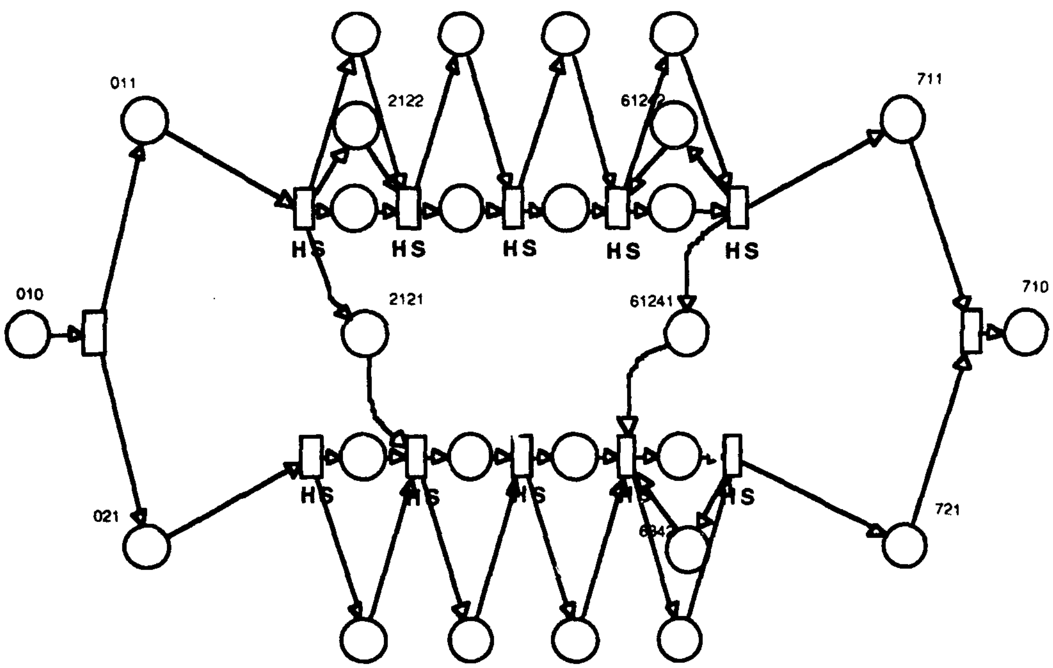


Figure B-3

Figure B-4 presents the organizational structure in stratum '1'. Again all compound transitions are identified by the label '**HS**'. Note the changes in place labels. These changes follow the scheme presented in Subsection 5.7.3. The designer of the organization has to make these changes in order to be consistent with the methodology. Once the labeling scheme is implemented by the designer, the software will keep track of the labels.

Figure B-5 shows the structure in stratum '1' with the subnets outlined by dashed boxes. The folding of this structure will yield the stratum '0' representation of the organization under consideration.

The folding procedure discussed before is again applied to each subnets. One obtains as a result the structure shown in Figure B-6

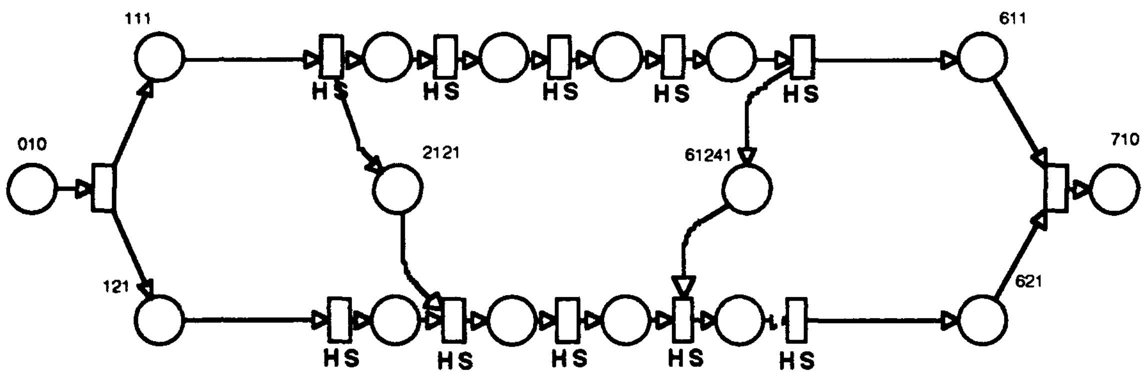


Figure B-4

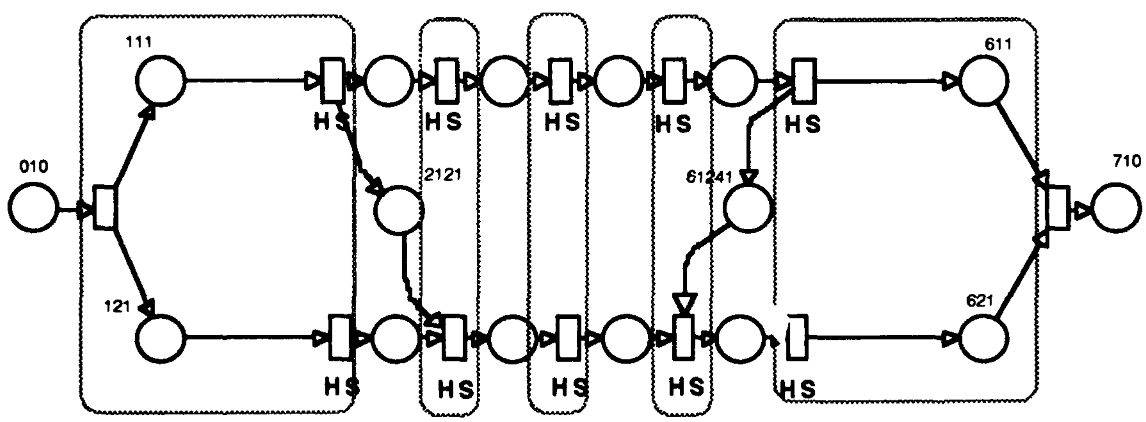


Figure B-5

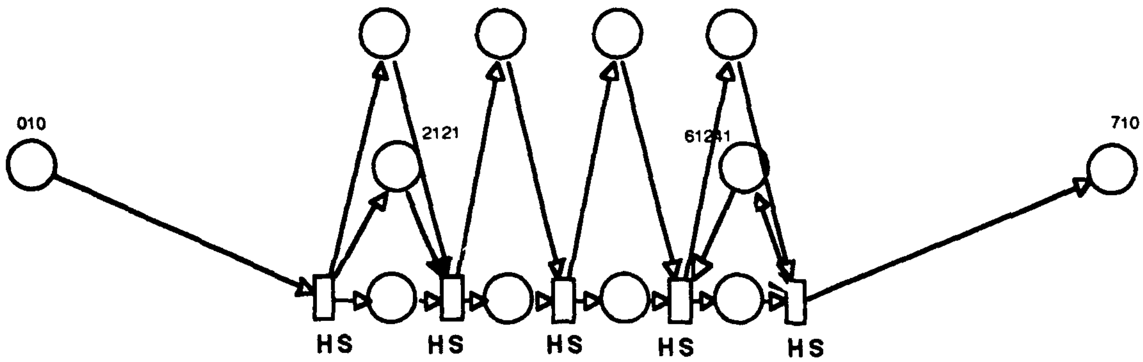


Figure B-6

The redundant places and interactional places defined in stratum '1' are omitted and a structure shown in Figure B-7 is obtained. The structure in Figure B-7 represents the compound node representation of an entire organization. Again, note the changes in place labels. The designer has to be careful in order to be consistent with the labelling scheme.

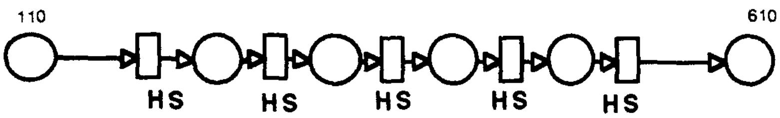


Figure B-7

Once an organizational structure is folded to its highest stratum, it is recommended to save the stratum '0' representation of the entire organization. One now can unfold and fold the structure to any desired description.

UNFOLDING

As mentioned before, a folded structure can be unfolded to its lower strata descriptions. In this section those procedures are outlined which are necessary to be carried out in order to unfold an organizational structure in stratum '0' to its lowest stratum (stratum '2') description. For illustration, the folded structure in Figure B-7 is considered.

The leftmost transition shown in Figure B-7 is selected and the command **Replace by Subpage** is invoked in **CPN** menu. An options window will appear as a result. Disable the option **Delete Port Nodes with No Port Assignments**. Click **OK**. The lower stratum (stratum '1') representation of the compound transition will replace the compound transition and the Petri Net structure obtained as a result will be similar to one shown in Figure B-8. Note that the interactional place **2121** has reappeared as a result of unfolding. This place along with other places was deleted in stratum '0' during the folding process, but the lower stratum description preserved the lower stratum connectivity.

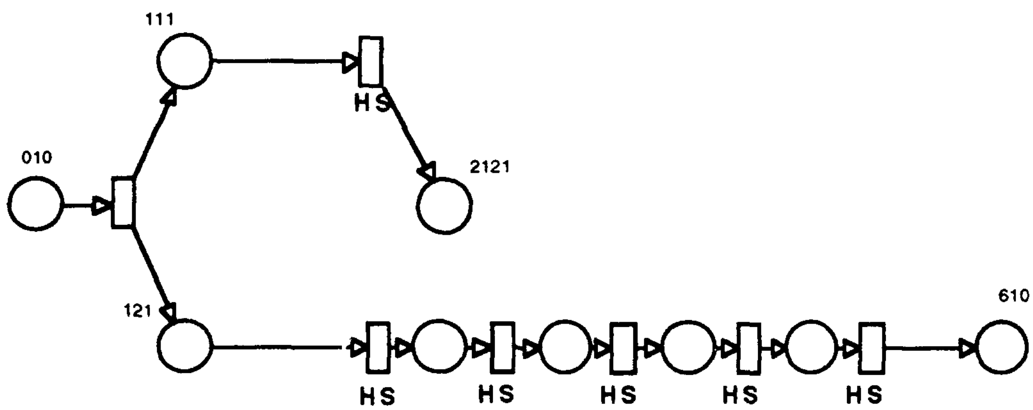


Figure B-8

Similarly, another compound transition (one representing the SAC stage) is unfolded and the structure shown in Figure B-9 is obtained. The interactional link defined in stratum '1' can now be seen with its input and output connections. Also, note the changes in the labels. This time, the designer does not have to do anything; the software has kept track of all labels.

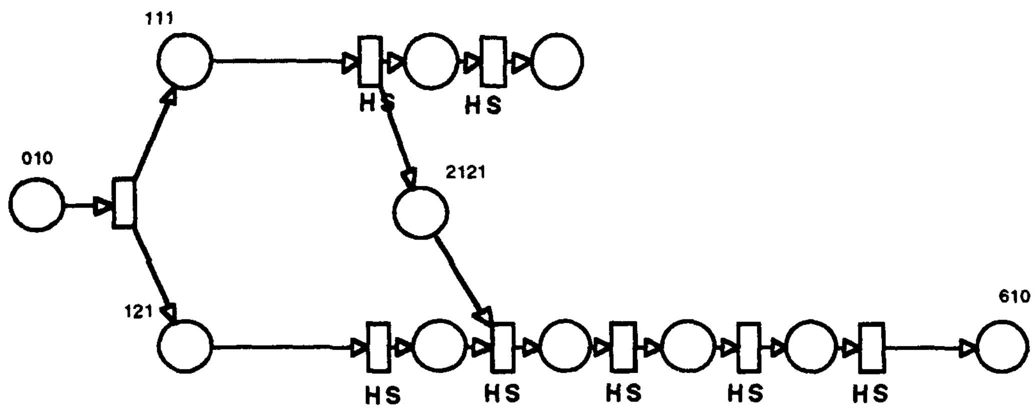


Figure B-9

At this point, it is necessary to describe a problem in the software. In Figure B-9, the interactional place **2121** is shown connected to its input and output transitions. But the place represented by **2121** is actually two places placed on top of each other, one with the input arc and the other with the output arc. The software does remember the precise location of a place that has been folded but loses its connectivity. The situation is depicted in Figure B-10.

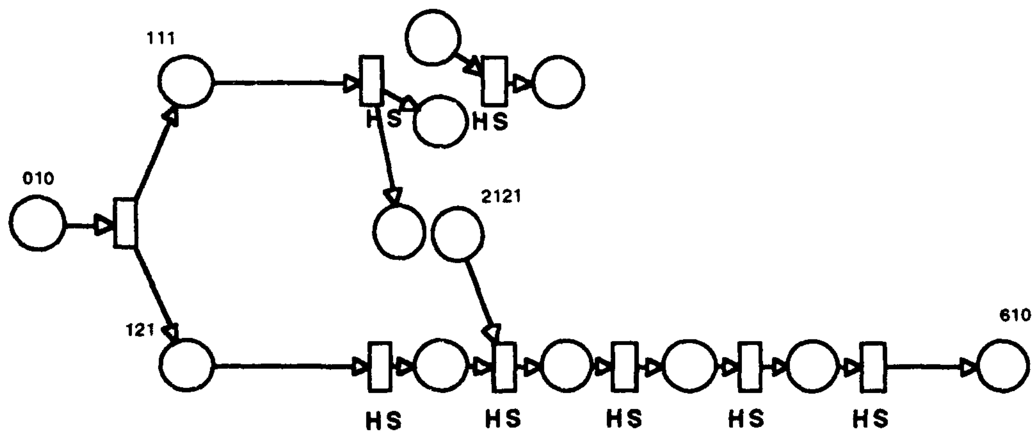


Figure B-10

The only solution to this problem at this stage is to maintain the connectivity manually. An arc of one of the two places is selected and then connected to the other place by dragging it to the place. The place with no input and output arcs can now be deleted. This procedure is shown in Figure B-11

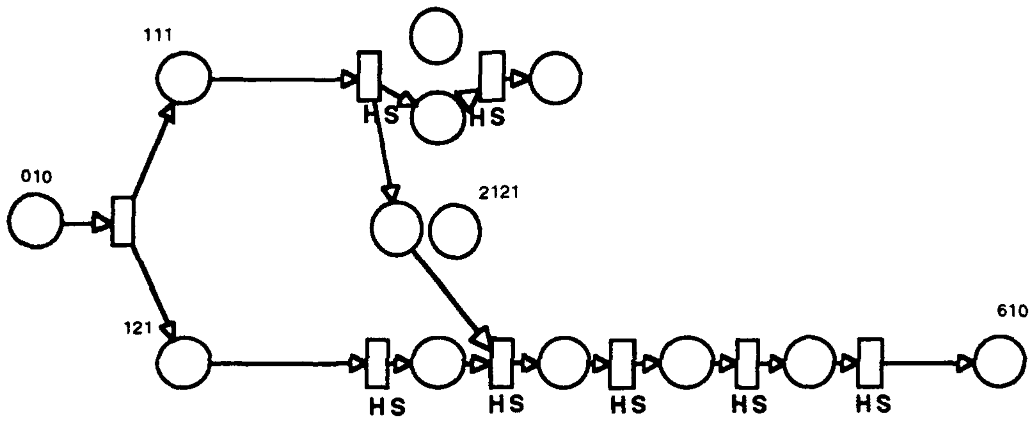


Figure B-11

Figure B-12 shows the organizational structure in stratum '1'. The structure in Figure B-12 is obtained as a result of unfolding all the compound transitions of the net in Figure B-7 and connecting all the unconnected (though not visually) places.

Figure B-13 presents the organizational structure in stratum '2' which is obtained by unfolding the compound transitions of the net in Figure B-12. The unfolding is done according to the procedure outlined before. Again, the nets presented in this discussion need not be executable on computer. One has to program them according to the procedures presented in the *Design /CPN™* manuals.

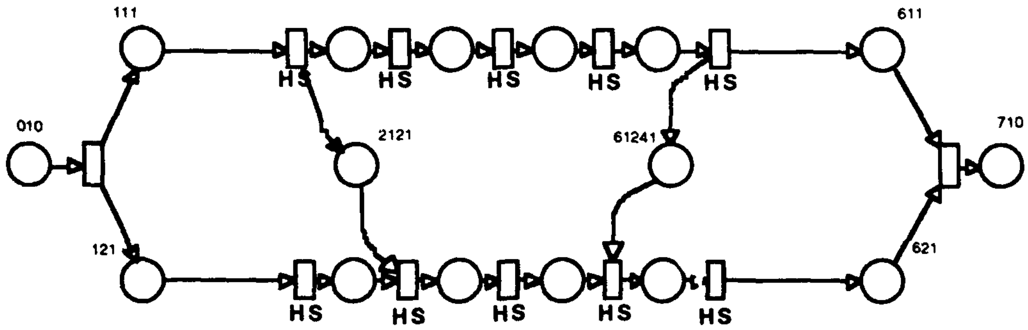


Figure B-12

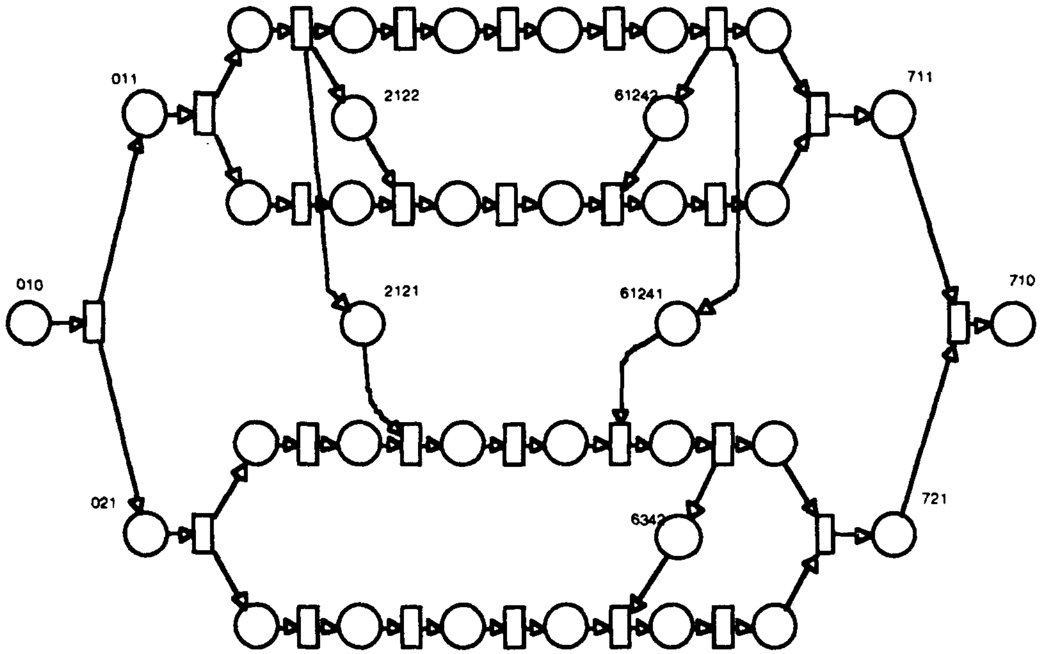


Figure B-13

APPENDIX C

GLOSSARY

ANSI	American National Standard Institute
$\mathbf{C_{ij}}$	Interaction from RS/RSC stage of DMU 'i' to CI/CIC stage of DMU 'j'
CI/CIC	Command Interpretation
CMD	Command
CTR	Control
DM	Decision Maker (human)
DMO	Decision Making Organization
DMSO	Decision Making Sub-Organization
DMU	Decision Making Unit
$\mathbf{DMU_{ik}}$	Decision Makining Unit 'i' in Stratum 'k'
$\mathbf{d[\Sigma_{ik+1}]}$	Size of WDN Σ_{ik+1}
$\mathbf{e_i}$	External input to SA/SAC stage of DMU 'i'
$\mathbf{F_{ij}}$	Interaction from SA/SAC stage of DMU 'i' to IF/IFC stage of DMU 'j'
FO	Feasible Organization
$\mathbf{G_{ij}}$	Interaction from RS/RSC stage of DMU 'i' to SA/SAC stage of DMU 'j'
g.l.b	Greatest Lower Bound
$\mathbf{H_{ij}}$	Interaction from RS/RSC stage of DMU 'i' to IF/IFC stage of DMU 'j'
I	Order defined on Inputs
IF/IFC	Information Fusion
INF	Information
$\mathbf{\Delta_{q,k-1,\mathit{l}}}$	Incidence matrix in Stratum 'l' of the Organizational Structure of Node 'q' in Stratum 'k-1'
$\mathbf{(L_{qk})_{max}}$	Maximum number of Links in a WDN of a Compound Node
l.u.b	Least Upper Bound

MAXO	Maximal element of the set W(R)
MINO	Minimal element of the set W(R)
$(M_{qk})_{max}$	Maximum number of Transitions in the WDN of Node 'q'
$M_{k\ max}$	Maximum number of Transitions in the Organizational Structure defined in Stratum 'k'
μ_k	Set of all Nodes in Stratum 'k'
N	Lowest degree of Abstraction possible
n	Lowest degree of Abstraction desired
$(N_{qk})_{max}$	Maximum number of Places in the WDN of Node 'q'
$N_{k\ max}$	Maximum number of Places in the Organizational Structure defined in Stratum 'k'
O	Order defined on Outputs
PDL	Program design Language
Π	Set of all DMUs represented in a 2-tuple form (I, O)
R	Set of all Constraints
R_f	User-defined Constraints given by 1s and 0s in the arrays {**e**, **s**, **F**, **G**, **H**, **C**}
R_p	Special Constraints
R_s	Structural Constraints
RS/RSC	Response Selection
R_u	User-defined Constraints
SA/SAC	Situation Assessment
SDMO	Stratified Decision Making Organization
s_i	External Output from RS/RSC stage of DMU 'i'
S_p	Simple Path
$S_p(R_u)$	Set of all Simple Paths of the Universal Net
TP/TPC	Task Processing
$US_p(R_u)$	Set of all possible Unions of elements of $S_p(R_u)$
W	Set of WDNs
WDS	Well Defined Structure
WDN	Well Defined Net
W(R)	Set of all WDNs satisfying R
$W(R_u)$	Set of all WDNs satisfying Ru
$W_{max}(R)$	Set of all MAXOs

$\mathbf{W_{min}(R)}$	Set of all MINOs
Ω	Greatest Element
ω	Least Element
Σ_{ik+1}	WDN in Stratum 'k+1' of a Node 'i' in Stratum 'k'
$\Omega(\mathbf{R_u})$	Universal Net
$\omega(\mathbf{R_u})$	Kernel Net