ANALYSIS AND SYNTHESIS OF DECISION-MAKING ORGANIZATIONS

by

Stamatios K. Andreadakis

Diploma in Mechanical and Electrical Engineering
National Technical University of Athens, 1979

S.M. in Mechanical Engineering
Massachusetts Institute of Technology, 1982

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1988

© Massachusetts Institute of Technology, 1988

Signature of Author

Department of Mechanical Engineering
January 8, 1988

Certified by

Dr. Alexander H. Levis
Thesis Supervisor

Accepted by

Professor Ain A. Sonin
Chairman, Departmental Committee on Graduate Students
Department of Mechanical Engineering
ANALYSIS AND SYNTHESIS OF DECISION-MAKING ORGANIZATIONS

by

Stamatios K. Andreadakis

Submitted to the Department of Mechanical Engineering on January 8, 1988 in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

The objective is to design distributed decision-making organizations. The synthesis problem is formulated as follows: Given a mission, design the decision-making organization that is accurate, timely, has task throughput rate higher than the task arrival rate, and whose decisionmakers are not overloaded. The fundamental feature of the methodology is the distinction between the design of the data flow structure and the design of the organization architecture. The data flow structure synthesis focuses on information processing schemata. A classification of these structures, suitable for decision-making, is introduced, and a procedure is developed for the generation of candidate structures. The organization architecture synthesis focuses on function allocation to decisionmakers, and on the selection of the supporting systems. The candidate designs are augmented by incorporating decision support systems and communication links. The performance and effectiveness of decision-making organizations is analyzed. An Information theoretic model of the human decisionmaker with bounded rationality is used to model the information processing and decision-making functions. The formalism of timed Petri Nets is employed to model the interaction of processes, and to compute time related measures of performance. The concepts of Timeliness and Survivability are investigated, and the respective measures are defined. The accuracy, response time, and maximum throughput rate, are computed. A Measure of Effectiveness is evaluated for each design, and the organization with the greatest measure is selected. The design of a Command and Control organization for a naval air-defense example is employed to illustrate the methodology.

Thesis Supervisor:  Dr. Alexander H. Levis
Title:  Senior Research Scientist
Acknowledgements

Dr. Alexander H. Levis, has guided the development of this thesis through his insightful suggestions, comments, and constructive critique. His high standards on both technical issues and technical writing have enhanced the quality of this work. Research under his supervision has been a rewarding experience; I am grateful for his advise and support.

Thanks are due to the readers of the thesis, Professors Thomas B. Sheridan, John N. Tsitsiklis, and Kamal Youcef-Toumi. They have provided helpful suggestions and have contributed to the clarity of the dissertation.

The selection of the naval air-defense example has been the result of several conversations with Lt. Chris Barrett, U.S. Navy. His interest in this work, and his help are greatly appreciated.

I wish to thank my friends and fellow students John Kyartzoglou and Vicky Jin. John has dedicated many hours listening to the formulation of several concepts of this thesis. His interaction has helped me clear my thoughts and is highly valued. Vicky has diligently proof-read the appendices, and verified the numerical results of the Information theoretic example. Her help is thankfully acknowledged.

Finally, I wish to express my gratitude to my parents for their constant support and encouragement during my studies at MIT.

The Petri Nets of the second chapter have been drawn with the “Petri Net CAD” software developed by John Kyartzoglou, while the Petri Nets of chapters four, five and six have been drawn with the “Design” software developed by Meta Software.

This research was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the U.S. Office of Naval Research under contract No. N00014-84-K-0519 (NR 649-003).
To my parents

Kallistratos and Polyzeni
Contents

Abstract 2

Acknowledgements 3

List of Figures 10

List of Tables 17

1 INTRODUCTION 19

1.1 Decision-making Organizations ............................ 19

1.2 Contribution of this Thesis ................................... 21

1.3 Overview of the Thesis ....................................... 24

2 REVIEW OF PETRI NET THEORY 25

2.1 Introduction .................................................... 25

2.2 Definitions ................................................... 25

2.3 Incidence matrix ............................................... 27

2.4 Duality ....................................................... 27

2.5 Subnet ......................................................... 28

2.6 Marking and Firing Sequence ................................ 28
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7 Petri Net Properties</td>
<td>30</td>
</tr>
<tr>
<td>2.8 Graph-Theoretic Definitions</td>
<td>31</td>
</tr>
<tr>
<td>2.9 S and T Invariants</td>
<td>31</td>
</tr>
<tr>
<td>2.10 The Switch</td>
<td>35</td>
</tr>
<tr>
<td>2.11 Macro-nodes</td>
<td>36</td>
</tr>
<tr>
<td>2.12 Information Flow Paths</td>
<td>36</td>
</tr>
<tr>
<td>2.13 Timed Petri Nets</td>
<td>40</td>
</tr>
<tr>
<td>2.14 Resource Availability Place</td>
<td>44</td>
</tr>
<tr>
<td>2.15 Synchronization Loop</td>
<td>46</td>
</tr>
<tr>
<td>2.16 Lines and Slices of Occurrence Nets</td>
<td>48</td>
</tr>
<tr>
<td>2.17 Chapter Summary</td>
<td>50</td>
</tr>
<tr>
<td>3 INFORMATION THEORY FUNDAMENTALS</td>
<td>51</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>51</td>
</tr>
<tr>
<td>3.2 Entropy and Transmission</td>
<td>51</td>
</tr>
<tr>
<td>3.3 Partition Law of Information</td>
<td>53</td>
</tr>
<tr>
<td>3.4 Computation of Total Activity</td>
<td>56</td>
</tr>
<tr>
<td>3.5 Modeling of Input Tasks</td>
<td>58</td>
</tr>
<tr>
<td>3.6 Chapter Summary</td>
<td>59</td>
</tr>
<tr>
<td>4 ANALYSIS OF C² ORGANIZATIONS</td>
<td>60</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>60</td>
</tr>
</tbody>
</table>
4.2 Model of the Command and Control Process ......................................... 60
4.3 Model of Command and Control Organizations ....................................... 64
4.4 Decision Strategies .............................................................................. 66
4.5 Workload Computation ......................................................................... 70
4.6 Computation of Processing Time ............................................................. 71
4.7 Timeliness .............................................................................................. 72
4.8 Computation of Throughput Rate ........................................................... 76
4.9 Computation of Accuracy ...................................................................... 80
4.10 Effectiveness of $C^2$ Organizations ...................................................... 81
  4.10.1 Organization Locus and Requirements Locus ..................................... 81
  4.10.2 Measure of Effectiveness ................................................................. 83
4.11 Chapter Summary ................................................................................ 84

5 SYNTHESIS OF $C^2$ ORGANIZATIONS ..................................................... 85
  5.1 Introduction .......................................................................................... 85
  5.2 Synthesis Problem Formulation ............................................................. 85
  5.3 Synthesis Approach .............................................................................. 87
  5.4 Description of the Synthesis Methodology .......................................... 87
  5.5 Classification of Data Flow Structures ................................................. 89
  5.6 Data Flow Structures Design ............................................................... 98
  5.7 Design Parameters .............................................................................. 103
5.8 Examples ................................................................. 104
5.9 Evaluation of Data Flow Structures ........................................ 110
5.10 Organization Architecture Design ......................................... 113
5.11 Modification of Candidate Designs and Selection ....................... 115
5.12 Chapter Summary ........................................................ 117

6 APPLICATION OF THE SYNTHESIS METHODOLOGY ....................... 118
6.1 Introduction ................................................................ 118
6.2 Problem Description ........................................................ 118
6.3 Outline of Synthesis ......................................................... 120
6.4 Generation of Data Flow Structures ....................................... 120
6.5 Function and Algorithm Description ....................................... 130
6.6 Data Flow Structure Evaluation and Selection ......................... 136
6.7 Organization Design ....................................................... 142
6.8 Model of Input Tasks and Cost Function Selection .................... 147
6.9 Decision Strategies ........................................................ 149
6.10 Design Evaluation ........................................................ 149
6.11 Chapter Summary ........................................................ 167

7 A GLOBAL VIEW ............................................................. 168
7.1 Introduction ................................................................ 168
7.2 Survivability ............................................................... 168
7.3 Computation of Survivability Measure ........................................ 169
7.4 Reconfigurability ................................................................. 172
7.5 Summary and Conclusions ...................................................... 173
7.6 Directions for Future Research ................................................ 173

REFERENCES ................................................................. 175

APPENDICES

A ALTERNATE ALGORITHMS' ACTIVITY COMPUTATION ...................... 178
  A.1 Algorithm Selection Independent of Input Value ....................... 178
  A.2 Algorithm Selection Dependent on Input Value ....................... 183

B ACTIVITY COMPUTATION EXAMPLE ............................................. 185
  B.1 Activity Computation of First Algorithm ................................ 185
  B.2 Activity Computation of Second Algorithm ............................ 191
  B.3 Activity Computation for Alternate Algorithm Use .................. 193
List of Figures

2.1 Petri Net ........................................... 26
2.2 Dual Petri Net ..................................... 28
2.3 S component \([X_1]\) .................................. 34
2.4 S component \([X_2]\) .................................. 34
2.5 Representation of switch ............................ 35
2.6 Macro Nodes ......................................... 37
2.7 Source and sink added to Petri Net .................. 38
2.8 Feedback Loop added to Petri Net .................. 38
2.9 Simple Paths ........................................ 39
2.10 Petri Net with switches ............................ 40
2.11 Complete path \# 1 .................................. 41
2.12 Complete path \# 2 .................................. 41
2.13 Complete path \# 3 .................................. 42
2.14 Complete path \# 4 .................................. 42
2.15 Complete path \# 5 .................................. 43
2.16 Complete path \# 6 .................................. 43
2.17 Transition equivalence for timed Petri Nets ....... 44
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.18</td>
<td>Resource availability place</td>
<td>45</td>
</tr>
<tr>
<td>2.19</td>
<td>Resource availability place replaced by transition and two places</td>
<td>45</td>
</tr>
<tr>
<td>2.20</td>
<td>Resource availability place for a decisionmaker</td>
<td>46</td>
</tr>
<tr>
<td>2.21</td>
<td>Petri Net without synchronization of observations</td>
<td>47</td>
</tr>
<tr>
<td>2.22</td>
<td>Petri Net with synchronization of observations</td>
<td>47</td>
</tr>
<tr>
<td>2.23</td>
<td>Lines of Petri Net</td>
<td>49</td>
</tr>
<tr>
<td>2.24</td>
<td>Slices of Petri Net</td>
<td>49</td>
</tr>
<tr>
<td>3.1</td>
<td>System and internal variables</td>
<td>53</td>
</tr>
<tr>
<td>3.2</td>
<td>Function implemented by alternative algorithms</td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>Model of the Command and Control process</td>
<td>61</td>
</tr>
<tr>
<td>4.2</td>
<td>Interaction of processing channels</td>
<td>63</td>
</tr>
<tr>
<td>4.3</td>
<td>Petri Net of alternate procedure use</td>
<td>65</td>
</tr>
<tr>
<td>4.4</td>
<td>Petri Net of decision aid - database access</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>Petri Net with decision switch selecting decision aid - database access</td>
<td>66</td>
</tr>
<tr>
<td>4.6</td>
<td>Organization with two functions implemented by alternate algorithms</td>
<td>69</td>
</tr>
<tr>
<td>4.7</td>
<td>Window of opportunity</td>
<td>73</td>
</tr>
<tr>
<td>4.8</td>
<td>Computation of pdf of maximum of two stochastic delays</td>
<td>75</td>
</tr>
<tr>
<td>4.9</td>
<td>Event Graph</td>
<td>77</td>
</tr>
<tr>
<td>4.10</td>
<td>Directed Elementary Circuits</td>
<td>77</td>
</tr>
<tr>
<td>4.11</td>
<td>Elementary circuits with common transition</td>
<td>80</td>
</tr>
</tbody>
</table>
4.12 Computation of accuracy ........................................... 81
4.13 Decision strategy space and mapping to MOP space ............. 82
4.14 Intersection of Organization locus and Requirements locus ........ 83

5.1 Flowchart of the Design Methodology ......................... 88
5.2 Information Flow Types ......................................... 92
5.3 Class 1 Data Flow Structure ................................. 92
5.4 Class 2 or Class 3 Data Flow Structure ..................... 93
5.5 Class 12 Data Flow Structure .................................. 93
5.6 Class 13 Data Flow Structure .................................. 94
5.7 Class 123 Data Flow Structure ................................ 94
5.8 Class 2 DFS with r=2, c=2 .................................. 96
5.9 Class 2 DFS with r=3, c=3 ................................... 97
5.10 Block form of Incidence Matrix ............................... 101
5.11 Class 1 DFS: Incidence Matrix ................................ 106
5.12 Class 1 DFS: Petri Net ....................................... 107
5.13 Class 13 DFS: Incidence Matrix .............................. 108
5.14 Class 13 DFS: Petri Net ....................................... 109
5.15 Class 123 DFS: Incidence Matrix ............................ 111
5.16 Class 123 DFS: Petri Net ..................................... 112
5.17 Synchronous Protocol ....................................... 114
6.20 Local Courses of Action Development; Algorithm 2 ............... 138
6.21 Global Response Selection Algorithm .................................. 139
6.22 Local Response selection Algorithm .................................... 140
6.23 Data Flow Structure A ..................................................... 141
6.24 Data Flow Structure B ..................................................... 141
6.25 First Function Allocation for DFS A ................................... 143
6.26 Second Function Allocation for DFS A .................................. 143
6.27 First Function Allocation for DFS B ................................... 144
6.28 Second Function Allocation for DFS B .................................. 144
6.29 Organization 1 ............................................................... 145
6.30 Organization 2 ............................................................... 145
6.31 Organization 3 ............................................................... 146
6.32 Organization 4 ............................................................... 146
6.33 Organization 1, MOP locus ............................................... 151
6.34 Organization 1, MOP locus projection on the J - T plane .......... 151
6.35 Organization 1, MOP locus projection on the J - R plane .......... 152
6.36 Organization 1, MOP locus projection on the R - T plane .......... 152
6.37 Organization 2, MOP locus ............................................... 153
6.38 Organization 2, MOP locus projection on the J - T plane .......... 153
6.39 Organization 2, MOP locus projection on the J - R plane .......... 154
6.40 Organization 2, MOP locus projection on the R - T plane .......... 154
6.41 Organization 3, MOP locus ............................................. 155
6.42 Organization 3, MOP locus projection on the $J - T$ plane ........ 155
6.43 Organization 3, MOP locus projection on the $J - R$ plane .......... 156
6.44 Organization 3, MOP locus projection on the $R - T$ plane .......... 156
6.45 Organization 4, MOP locus ............................................. 157
6.46 Organization 4, MOP locus projection on the $J - T$ plane ........ 157
6.47 Organization 4, MOP locus projection on the $J - R$ plane .......... 158
6.48 Organization 4, MOP locus projection on the $R - T$ plane .......... 158
6.49 Organization 1, MOE versus $J_o - T_o$ requirements ............... 160
6.50 Organization 1, MOE versus $J_o - R_o$ requirements ............... 160
6.51 Organization 1, MOE versus $T_o - R_o$ requirements ............... 161
6.52 Organization 2, MOE versus $J_o - T_o$ requirements ............... 161
6.53 Organization 2, MOE versus $J_o - R_o$ requirements ............... 162
6.54 Organization 2, MOE versus $T_o - R_o$ requirements ............... 162
6.55 Organization 3, MOE versus $J_o - T_o$ requirements ............... 163
6.56 Organization 3, MOE versus $J_o - R_o$ requirements ............... 163
6.57 Organization 3, MOE versus $T_o - R_o$ requirements ............... 164
6.58 Organization 4, MOE versus $J_o - T_o$ requirements ............... 164
6.59 Organization 4, MOE versus $J_o - R_o$ requirements ............... 165
6.60 Organization 4, MOE versus $T_o - R_o$ requirements ............... 165
6.61 Organization 2, MOE versus $T_o - R_o$ requirements, $J_o = 1.2$ .... 166
6.62 Organization 4, MOE versus $T_o - R_o$ requirements, $J_o = 1.2$ ........ 166

B.1 Flowchart of first algorithm .......................................... 186

B.2 Flowchart of second algorithm ........................................ 191
### List of Tables

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Classification of Data Flow Structures</td>
<td>91</td>
</tr>
<tr>
<td>5.2</td>
<td>Transition Sets</td>
<td>98</td>
</tr>
<tr>
<td>5.3</td>
<td>Place Sets</td>
<td>99</td>
</tr>
<tr>
<td>5.4</td>
<td>Design Modification Cases</td>
<td>116</td>
</tr>
<tr>
<td>6.1</td>
<td>Computation of the number of DF transitions, class 2 DFS</td>
<td>121</td>
</tr>
<tr>
<td>6.2</td>
<td>Computation of the number of RF transitions, class 1 DFS</td>
<td>122</td>
</tr>
<tr>
<td>6.3</td>
<td>Computation of the number of RF transitions, class 12 DFS</td>
<td>122</td>
</tr>
<tr>
<td>6.4</td>
<td>Input alphabet</td>
<td>147</td>
</tr>
<tr>
<td>6.5</td>
<td>Cost Function $C(Y_i,Y_{di})$</td>
<td>148</td>
</tr>
<tr>
<td>6.6</td>
<td>Measures of Performance</td>
<td>150</td>
</tr>
<tr>
<td>B.1</td>
<td>Input alphabet and corresponding probability</td>
<td>185</td>
</tr>
<tr>
<td>B.2</td>
<td>Output $Y$ of first algorithm</td>
<td>186</td>
</tr>
<tr>
<td>B.3</td>
<td>Input alphabet</td>
<td>189</td>
</tr>
<tr>
<td>B.4</td>
<td>Output $Y$ of second algorithm</td>
<td>191</td>
</tr>
<tr>
<td>B.5</td>
<td>Function Output $Z$, and Algorithm Output $Y_1,Y_2$</td>
<td>194</td>
</tr>
<tr>
<td>B.6</td>
<td>Probability mass function of output $Z$, and entropy $H(Z)$</td>
<td>195</td>
</tr>
<tr>
<td>B.7</td>
<td>Coordination $G_c$ as a function of $p_1 = p(u = 1)$</td>
<td>195</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

1.1 Decision-making Organizations

The need for decision-making organizations emerges when the amount of information processing required to reach a decision exceeds the processing capacity of one decisionmaker. Similarly, decision-making organizations are formed when a decision requires knowledge or information not available to one decisionmaker, and when training and specialization of the decisionmakers can improve the accuracy and timeliness of the decision.

A decision-making organization is defined here as a team of decisionmakers. The decisionmaking process is partitioned into subtasks (sub-processes) and each subtask is performed by a properly trained member of the team. Decision aids and databases may be introduced to help decisionmakers process the information and select a response. Rules for the information exchange among the decisionmakers must be established, so that they can process the information effectively and produce an accurate response in a timely manner.

A specific class of decision-making organizations implements the Command and Control process. Command and Control is the process through which the chief executive officer of an organization directs his staff and deploys his resources in order to achieve his objective. In the military context, the Command and Control process has been defined as follows: “Command and Control is the exercise of authority and direction by a properly designated commander over assigned forces in the accomplishment of his mission” (Joint Chiefs of Staff, Publication 1).

A Command and Control ($C^2$) Organization is a decision-making organization that
implements the Command and Control process. Examples of $C^2$ Organizations are air-
traffic control centers of the air transportation system, command centers of military
forces, the control centers of nuclear power plants, and the trading desks of financial
institutions.

Command, Control, and Communications ($C^3$) Systems are the systems that support
$C^2$ Organizations. $C^3$ Systems are the nervous systems of decision-making organ-
izations; they are designed to help decisionmakers carry out their tasks.

The analysis of decision-making / $C^2$ organizations evaluates their performance and
effectiveness. The decision-making process is implemented by humans and systems;
thus, the analysis of decision-making organizations incorporates concepts and methods
from several disciplines which include Human Factors (Man-Machine systems) Engi-

The design of a $C^2$ organization and the design of the $C^3$ system that supports
it are two parts of the same problem. Specification of the functions allocated to the
different decisionmakers determines the $C^3$ system components and the connectivity
that is required. Conversely, specification of the $C^3$ system imposes a structure on the
$C^2$ organization and restricts the allocation of the information processing and decision-
making functions to the members of the $C^2$ organization.

The synthesis of decision-making / $C^2$ organizations is a complex process that must
address a multitude of questions: specifically how to partition the task, how many
decisionmakers to select, how to allocate the subtasks to the decisionmakers, how to
select the schema of information exchange among the decisionmakers (protocols), what
kind of communication hardware is required for timely transmission of information and
data in a given operating environment, what the structure of the required databases
and the specifications of the respective hardware should be, and how to design decision
aids and allocate them to the decisionmakers. Consequently, it is necessary to develop
a methodology so that the design of decision-making / $C^2$ Organizations become a
structured process.
1.2 Contribution of this Thesis

The objective of this thesis is twofold: to develop a structured methodology for the synthesis of distributed decision-making / $C^2$ organizations and a systematic procedure for their analysis.

The major contribution of this thesis is the development of a synthesis methodology for decision-making / $C^2$ organizations. The methodology tackles the design problem at two levels: the data flow structure level and the organization architecture level. The data flow structure design focuses on the generation of information processing schemata, while the organization design addresses the allocation of tasks to decisionmakers and the selection of the supporting systems. The significance of this differentiation is the ability to generate data flow structures suitable for the information processing task without consideration of the organization design constraints. The resulting design flexibility allows the fine-tuning of the decision-making organization at two levels: the data flow structure level and the organization architecture level.

A novel approach for the classification and generation of data flow structures, suitable for decision-making, has been introduced. The notions of data fusion and results fusion are pivotal in the formulation of the data flow structures classification. A procedure has been developed for the synthesis of structures parameterized by the complexity and redundancy of the information processing required by the task.

The organization synthesis develops organization architectures from each data flow structure through the allocation of functions to organization members. The function allocation is performed on the basis of the specialization required, and the physical location where each task is performed. Thus, the number of organization members is derived by the design, as opposed to being a design parameter. The addition of communication links, decision aids and databases transforms each design organization to the corresponding $C^2$ organization.

The analysis of decision-making / $C^2$ organizations evaluates their performance and effectiveness. The measures of performance include the response time, and the throughput rate.
The second contribution of the thesis is a formulation of the reconfigurability problem of $C^2$ organizations, and the investigation of their survivability. The concept of survivability has been explored and a measure of survivability has been defined. The framework for the computation of the measure of survivability has been developed. The reconfigurability problem has been formulated as a set of synthesis problems of $C^2$ organizations, parameterized by the number of available assets.

The third contribution of this thesis is the computation of the response time and the throughput rate of decision-making organizations, based on estimates of the processing time that is required by the human decisionmakers to perform their assigned tasks. Specifically, the processing time of a decision-making function is estimated by the ratio of the information theoretic total activity (workload) of the function divided by the rationality threshold (maximum processing rate) of the human decisionmaker. The best response time and throughput rate for a given organizational architecture correspond to the shortest processing times of the individual processing functions. These are obtained when the decisionmakers operate at their maximum processing rate (rationality threshold). The rationality threshold varies among decisionmakers. Consequently, the response time and processing rate of the organization vary when the information processing tasks are performed by different decisionmakers.

Two approaches to the computation of response time and throughput rate have been formulated. In the first one, the maximum response time and minimum throughput rate, corresponding to the minimum value of the rationality threshold, are computed. In the second, the probability density functions of the response time and of the throughput rate, based on the probability density function of the rationality threshold are computed and two measures of performance have been defined: a measure of timeliness, and a measure of processing capacity.

Information theory and Petri Net theory are employed to analyze the performance of decision-making / $C^2$ organizations. Specifically, the Information theoretic model of the human decisionmaker, developed by Boettcher (1981), Boettcher and Levis (1982), Levis (1983), and the Petri Net model of decision-making organizations, introduced by Tabak and Levis (1985), are used.
The Information theoretic model of the human decisionmaker computes the workload of individual members of the organization. The model postulates that a decisionmaker is well trained and can select among several procedures in order to process the available information. In this thesis, this model is recast in order to compute the workload (total activity) of individual information processing and decision-making functions, which are implemented by the decisionmakers.

The Petri Net model of decision-making organizations has been used by Jin (1985, 1986) to compute the time delay of the organizations, for arbitrary deterministic delays of the processing functions. The same model has been used by Hillion (1986) to compute the throughput rate of organizations for deterministic processing delays.

This approach has been refined and expanded in this thesis, in order to compute the probability density functions of the response time and of the throughput rate of organizations for stochastic delays. Communications and decision support systems are an integral part of the distributed decision-making process. The delays across communication links and the access times of decision support systems affect the response time and throughput rate.

In this formulation, it is assumed that the probability density function of the rationality threshold is known, and that the probability density functions of the communication delays and access times of support systems are given. An analytic procedure is developed for the computation of the probability density functions of the response time and of the throughput rate.

The synthesis methodology evaluates the candidate designs on the basis of their effectiveness. The measure of effectiveness, is a global measure, which quantifies the degree to which an organization satisfies its requirements. The measures of performance and the measure of effectiveness of each candidate design are computed. Heuristic rules for the improvement of the measure of effectiveness have been formulated, based on the comparison of the individual measures of performance to the corresponding requirements. Finally, a design is selected on the basis of the highest measure of effectiveness.
1.3 Overview of the Thesis

The thesis is structured as follows: first, the mathematical tools employed in the modeling and the analysis of organizations are presented. Second, the analysis of organizations is described, and the measures of performance and effectiveness are defined. Next, the synthesis problem is formulated and the synthesis methodology is developed. Then, the methodology is applied to an example. Finally, the reconfigurability and survivability concepts are presented and the directions for future research are given.

In Chapter 2, the concepts of Petri Net theory are reviewed, and the models used in the analysis of decision-making organizations are described.

In Chapter 3, the fundamental definitions and identities of Information Theory are presented, as well as the algorithms for workload computation.

Chapter 4 develops the analysis of decision-making organizations. The model of the $C^2$ process is described, and the Measures of Performance and the Measure of Effectiveness are defined. Finally, procedures for their computation are presented.

Chapter 5 describes the synthesis of decision-making organizations. The problem formulation and the design methodology are presented. A classification of data flow structures is introduced. A procedure for the generation of data flow structures, and the process for the transformation of these structures to Command and Control organizations are developed. Finally the heuristic rules for the modification and fine-tuning of the candidate designs are presented.

In Chapter 6 the methodology is applied to the design of $C^2$ organizations for naval air defense.

In Chapter 7, the concept of survivability of $C^2$ organizations is investigated. A measure of survivability is defined and a procedure for its computation is developed. The reconfigurability of $C^2$ organizations is examined. Finally the conclusions and directions for further research are presented.
Chapter 2

REVIEW OF PETRI NET THEORY

2.1 Introduction

Petri Nets are bipartite directed multigraphs [Peterson (1981), Reisig (1985)]. The two types of nodes are: places, denoted by circles representing signals or conditions; and transitions, denoted by bars, representing processes or events. Places can only be connected to transitions, and transitions can only be connected to places. Tokens are markers that are deposited in places to denote that the corresponding conditions are satisfied. If all the places that are input to a transition are marked, then all the conditions that are prerequisites for the event represented by the transition are satisfied. Then the transition is enabled and it can "fire", i.e., the event can occur or the process can begin. When a transition fires, one token is consumed from each input place to the transition, and one token is deposited to each output place of the transition.

2.2 Definitions

A Petri Net is a bipartite directed graph, represented by a quadruple \([P,T,I,O]\).

- \(P\) is a finite set of places: \(P = \{p_1,p_2,\cdots,p_n\}\)
- \(T\) is a finite set of transitions: \(T = \{t_1,t_2,\cdots,t_m\}\)
- \(I\) is a mapping \(P \times T \rightarrow \{0,1\}\) corresponding to the set of directed links (arcs) from places to transitions.
- \(O\) is a mapping \(T \times P \rightarrow \{0,1\}\) corresponding to the set of directed links (arcs) from transitions to places.
The set of all output places of transition $t$ is denoted by $t^*$:

$$t^* = \{p \in P : O(t, p) = 1\}$$

The set of all input places of transition $t$ is denoted by $^*t$:

$$^*t = \{p \in P : I(t, p) = 1\}$$

Similarly, the set $p^*$ of all input transitions, and the set $^*p$ of all output transitions of place $p$ are defined by:

$$p^* = \{t \in T : I(t, p) = 1\}$$

$$^*p = \{t \in T : O(t, p) = 1\}$$

Figure 2.1 depicts a Petri Net. In the example of Figure 2.1,

$$t_1^* = \{2, 4\} \quad p_1^* = \{1\}$$

$$^*t_1 = \{1\} \quad ^*p_1 = \{4\}$$
2.3 Incidence matrix

The structure of a Petri Net can be represented by an integer matrix $C$, called incidence or flow matrix. For a Petri Net with $n$ places and $m$ transitions, $C$ is a $n \times m$ matrix whose elements are defined as follows:

$$ C_{ij} = \begin{cases} 
-1 & \text{if } p_i \text{ is an input place of } t_j \\
+1 & \text{if } p_i \text{ is an output place of } t_j \\
0 & \text{if there no link between } p_i \text{ and } t_j 
\end{cases} $$  \hspace{1cm} (2.1)

It can be verified that

$$ C_{ij} = O(t_j, p_i) - I(p_i, t_j) $$  \hspace{1cm} (2.2)

The incidence matrix of the Petri Net depicted in Figure 2.1 is:

$$ C = \begin{bmatrix} 
-1 & 0 & 0 & 1 \\
1 & -1 & 0 & 0 \\
0 & 1 & 0 & -1 \\
1 & 0 & -1 & 0 \\
0 & 0 & 1 & -1 
\end{bmatrix} $$  \hspace{1cm} (2.3)

2.4 Duality

The Petri Net $PN'$, obtained from a Petri Net $PN$, by exchanging places and transitions, and reversing the direction of the links, is called the dual of $PN$. Thus if $PN=(P,T,O,I)$ and $PN'=(P',T',O',I')$, then

$$ P' = T $$  \hspace{1cm} (2.4)

$$ T' = P $$  \hspace{1cm} (2.5)

$$ I'(p'_j, t'_i) = I(p_i, t_j) $$  \hspace{1cm} (2.6)

$$ O'(t'_i, p'_j) = O(t_j, p_i) $$  \hspace{1cm} (2.7)
Figure 2.2: Dual Petri Net

Figure 2.2 depicts the dual of the Petri Net of Figure 2.1.

2.5 Subnet

A subnet of a Petri Net $\text{PN}=(P,T,O,I)$ is a Petri Net $\text{PN}_s=(P_s,T_s,O_s,I_s)$, such that

$$P_s \subseteq P \tag{2.8}$$

$$T_s \subseteq T \tag{2.9}$$

and $I_s$ and $O_s$ are restrictions of $I$ and $O$ to $P_s \times T_s$ and $T_s \times P_s$ respectively.

2.6 Marking and Firing Sequence

A Marking of a Petri Net is a mapping of the places on the set of non-negative integers: $P \rightarrow \{0,1,2,...\}$. The marking is represented by a n-dimensional vector, whose components correspond to the places of the Petri Net.
A token is an entity, represented by a marker (dot), that can be deposited in the places of the Petri Net. The marking denotes how many tokens exist in the places of the Petri Net.

A transition is enabled by a marking \( M \) if each input place of the transition contains at least one token, i.e.

\[
M(p) \geq 1 \text{ for all } p \in ^*t
\]  

(2.10)

If a transition is enabled by a marking \( M \), it can fire. When a transition fires, a token is consumed (removed) from each of its input places and a token is deposited to each of its output places. The new marking of the Petri Net, \( M^* \), that results from the firing of transition \( t \) is obtained by

\[
M^*(p) = \begin{cases} 
M(p) - 1 & \text{if } p \in ^*t \\
M(p) + 1 & \text{if } p \in t^* \\
M(p) & \text{otherwise}
\end{cases}
\]  

(2.11)

or equivalently, when transition \( t_j \) fires

\[
M^*(p_i) = M(p_i) + O(t_j, p_i) - I(p_i, t_j)
\]  

(2.12)

A sequence of transition firings, called a firing sequence, is denoted by:

\[
\sigma_s = t_{j_1}, t_{j_2}, ..., t_{j_s}
\]  

(2.13)

The marking \( M_s \) which is reached from the initial marking \( M \) by firing the sequence \( \sigma_s \), is called a reachable marking and is denoted by:

\[
M \xrightarrow{\sigma_s} M_s
\]  

(2.14)

The set of all possible reachable markings from an initial marking \( M_o \), denoted by \( \hat{M}_o \), is called the forward marking class.
Assume that the initial marking of the Petri Net of Figure 2.1 is:

\[ M_0 = [1 \ 0 \ 0 \ 0 \ 0] \] (2.15)

For this marking there exist two firing sequences:

\[ \sigma_1 = t_1, t_2, t_3, t_4 \] (2.16)
\[ \sigma_2 = t_1, t_3, t_2, t_4 \] (2.17)

The markings reached under the firing sequence \( \sigma_1 = t_1, t_2, t_3, t_4 \) are

\[ M_1 = [0 \ 1 \ 0 \ 1 \ 0] \] (2.18)
\[ M_2 = [0 \ 0 \ 1 \ 1 \ 0] \] (2.19)
\[ M_3 = [0 \ 0 \ 0 \ 1 \ 1] \] (2.20)
\[ M_4 = [1 \ 0 \ 0 \ 0 \ 0] = M_0 \] (2.21)

whereas the markings reached under the firing sequence \( \sigma_2 = t_1, t_3, t_2, t_4 \) are

\[ M_1 = [0 \ 1 \ 0 \ 1 \ 0] \] (2.22)
\[ M_2 = [0 \ 1 \ 0 \ 1 \ 0] \] (2.23)
\[ M_3 = [0 \ 0 \ 0 \ 1 \ 1] \] (2.24)
\[ M_4 = [1 \ 0 \ 0 \ 0 \ 0] = M_0 \] (2.25)

The set of all reachable markings \( \hat{M}_0 \) from the initial marking \( M_0 \) is

\[ \hat{M}_0 = [M_0 \ M_1 \ M_2 \ M_3 \ M_4] \] (2.26)

2.7 Petri Net Properties

Liveness: A marking \( M_0 \) is live if, for any transition \( t \), and for every reachable marking \( M \), there exists a firing sequence from \( M \) which fires \( t \). This property guarantees that the firing process is deadlock free.

Boundedness: A marking \( M_0 \) is bounded if there exists a positive integer \( N \), such that for every reachable marking \( M \), the number of tokens in each place of the Petri Net is bounded by \( N \). If \( N \) equals one, the marking is said to be safe.
2.8 Graph-Theoretic Definitions

Connectivity: A Petri Net is connected if there exists a path from any node to any other node.

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

Directed Circuit: A directed circuit is a directed path from any node to itself.

Directed elementary circuit: A directed elementary circuit is a directed circuit in which no node appears more than once.

Event Graph: An event Graph is a connected Petri Net, in which each place has exactly one input and one output transition.

2.9 S and T Invariants

Definition: A n-dimensional non-negative integer vector $X$ is called an S-invariant [Memmi and Roucairol (1979)] if and only if

$$X^T C = 0$$  \hspace{1cm} (2.27)

Definition: A m-dimensional non-negative integer vector $Y$ is called a T-invariant if and only if

$$CY = 0$$  \hspace{1cm} (2.28)

The set of places whose corresponding components in $X$ are strictly positive is called the support of $X$ and is denoted by $< X >$.

The set of transitions whose corresponding components in $Y$ are strictly positive is called the support of $Y$ and is denoted by $< Y >$.

The support of an invariant is said to be minimal if and only if it does not contain the support of another invariant, but itself and the empty set.
Let $X$ be a S-invariant of Petri Net $PN$ and let $< X >$ be its support. The S-component associated with $< X >$ is the unique subnet whose set of places is $< X >$ and whose set of transitions consists of all the transitions connected to the places of $< X >$ [Sifakis (1978)]. In other words, the S-component corresponding to $< X >$ is the subnet $[X] = (P_s, T_s, I_s, O_s)$ such that:

$$P_s = < X >$$

$$T_s = \{p^*: p \in P_s\} \cup \{^*p: p \in P_s\}$$

(2.29) \hspace{1cm} (2.30)

$I_s$ is the restriction of $I$ to $P_s \times T_s$

$O_s$ is the restriction of $O$ to $P_s \times T_s$

T components are defined similarly. Let $Y$ be a T-invariant of $PN$ and let $< Y >$ be its support. The T-component associated with $< Y >$, denoted $[Y]$, is the unique subnet whose set of transitions is $< Y >$ and whose places are the input and output places of the transitions of $< Y >$.

An S-component is said to be minimal if it corresponds to an S-invariant whose support is minimal, i.e. it does not contain any other S-component but itself and the empty set.

The S invariants of the Petri Net of Figure 2.1 are found as follows:

Let $X^T = [abcde]$ ; then

$$\begin{bmatrix}
-1 & 0 & 0 & 1 \\
1 & -1 & 0 & 0 \\
0 & 1 & 0 & -1 \\
1 & 0 & -1 & 0 \\
0 & 0 & 1 & -1 \\
\end{bmatrix} [abcde] = 0$$

(2.31)

Hence,

$$a - b - d = 0$$

(2.32)

$$b - c = 0$$

(2.33)
\[
d - e = 0 \quad \quad \quad \quad (2.34)
c + e - a = 0 \quad \quad \quad \quad (2.35)
\]

Thus,
\[
a = b + d \quad \quad \quad \quad (2.36)
c = b \quad \quad \quad \quad \quad (2.37)
e = d \quad \quad \quad \quad \quad (2.38)
\]

and the S invariant is
\[
X = [b + d \quad b \quad b \quad d \quad d] \quad \quad \quad \quad (2.39)
\]

To obtain the minimal support S invariants let \( b = 1 \), \( d = 0 \), and \( b = 0 \), \( d = 1 \). Then
\[
X_1 = [1 \quad 1 \quad 1 \quad 0 \quad 0] \quad \quad \quad \quad (2.40)
\]

and
\[
X_2 = [1 \quad 0 \quad 0 \quad 1 \quad 1] \quad \quad \quad \quad (2.41)
\]

The corresponding supports are:
\[
< X_1 > = [p_1 \quad p_2 \quad p_3] \quad \quad \quad \quad (2.42)
\]
\[
< X_2 > = [p_1 \quad p_4 \quad p_5] \quad \quad \quad \quad (2.43)
\]

Note that these supports are minimal. Finally, other S invariants (not minimal support) are obtained by linear combination of the minimal support invariants \( X_1 \) and \( X_2 \), i.e.
\[
X = \lambda_1 X_1 + \lambda_2 X_2 \quad \quad \quad \quad (2.44)
\]

where \( \lambda_1 \) and \( \lambda_2 \) are integers.

The S component \([X_1]\) corresponding to \(< X_1 >\) is shown in Figure 2.3 and the S component \([X_2]\) corresponding to \(< X_2 >\) is shown in Figure 2.4.

An important result obtained by Hillion (1986) states that the directed circuits of an Event graph are exactly the minimal S components of the Event graph. This result is used to determine all the circuits of an Event graph using an algorithm developed by Martinez and Silva (1980), and improved by Alaiwan and Toudic (1985), that determines the minimal support S invariants of a net.
Figure 2.3: S component $[X_1]$

Figure 2.4: S component $[X_2]$
2.10 The Switch

A variant of the transition, the switch, has been introduced by Tabak and Levis (1985), to enable the modeling of the selection of one transition from a group of transitions. The switch is enabled if there is at least one token in each of its input places. When the switch fires, one token is consumed from each of its input places, and one token is deposited to only one of its output places, according to a decision rule that is associated with the switch. The switch is represented by a bar with rounded corners. A switch and the transitions on its branches are depicted in Figure 2.5. In modeling decision-making organizations, a decision rule selects one of the branches following the switch. The decision rule may be probabilistic. In this case, a probability $P_i$ is associated with branch $i$ such that

$$\sum_{i=1}^{n} P_i = 1 \quad (2.45)$$

for a switch with $n$ branches. Another rule may condition the probabilities of branch selection on the value of a variable.
2.11 Macro-nodes

A macro-node is a node that represents a subnet. A macro-transition is denoted by two rectangles one inside the other, while a macro-place is denoted by a two concentric circles, (Figure 2.6). The use of macro-nodes enables the aggregation of nodes of a Petri Net. For example, when alternate algorithms are used to process the information, the switch that selects which transition is enabled and the associated transitions that represent the alternate algorithms can be aggregated into a macro-transition. This aggregation makes possible the analysis of Petri Nets with decision switches, using results that are valid for event graphs.

2.12 Information Flow Paths

A Petri Net representing a decision-making organization receives data from the environment (source), which is represented by a transition, and produces responses which appear at its output nodes. A transition representing a sink can be connected to each of the output places of the Net (Figure 2.7).

Information flow paths are the paths of the Petri Net that emanate from the input (the source transition) and terminate at one of the outputs (sink transitions) of the net. When the switches and their corresponding branches are aggregated into macro-transitions, the Petri Net becomes an Event graph. A feedback loop may be used to connect the sink and the source (Figure 2.8). The addition of the feedback loop makes the Event graph strongly connected. Then the information flow paths are obtained from the directed elementary circuits of the Event graph, by removing the feedback loop.

In order to identify the directed elementary circuits of the Event graph, the algorithm developed by Alaiwan and Toudic (1985) may be used. Alternatively, another procedure, developed by Jin (1985, 1986), may be used to identify the information flow paths of the event graph. It should be noted that both algorithms cannot operate on nets which contain switches. Therefore, they should be applied to the Event Graphs
Figure 2.6: Macro Nodes
Figure 2.7: Source and sink added to Petri Net

Figure 2.8: Feedback Loop added to Petri Net
with the macro-transition aggregation of the decision switches and their branches. These information flow paths, are also called simple paths. The simple paths of the Petri Net of Figure 2.8 are depicted in Figure 2.9.

A set of concurrently active simple paths comprises a complete path of the Petri Net. When the net does not contain switches, there is only one complete path. If the net contains switches, then there exist more than one complete paths. When a switch selects one branch, one transition is enabled, while the rest are disabled. Thus some simple paths are active, i.e. conduct information, while others are inactive.

The number $N$ of complete paths for a net with $n$ switches is

$$N = \prod_{i=1}^{n} k_i$$

(2.46)

where $k_i$ is the number of branches of switch $i$.

When there are switches present in the Petri Net, the complete paths are obtained as follows: First the net is transformed into an event graph by representing the switches and their branches by macro-transitions. Then the complete path of the event graph is modified by replacing each macro-transition by the corresponding switch and one of its branches. The probability of each complete path is computed by multiplying the probabilities corresponding to the selected branches. As the switches implement their
Figure 2.10: Petri Net with switches

decision rules, different complete paths are generated, corresponding to the combinations of branch selections.

A Petri Net with switches is depicted in Figure 2.10 and its complete paths in Figures 2.11 through 2.16.

2.13 Timed Petri Nets

The Petri Net formalism, as presented so far, does not associate time with the operation of the net. Timed Petri Nets assign processing times to either the transitions or the places of the net. The processing times may be either deterministic or stochastic. It can be proved that assigning processing times to places is equivalent to assigning processing times to transitions [Sifakis (1980)].

In this respect, a timed Petri Net transition can be represented by two transitions
Figure 2.11: Complete path # 1

Figure 2.12: Complete path # 2
Figure 2.13: Complete path # 3

Figure 2.14: Complete path # 4
Figure 2.15: Complete path # 5

Figure 2.16: Complete path # 6
and a place connected to them, as depicted in Figure 2.17. When the transition is initiated, \( t_b \) fires and a token is removed from \( p_1 \) and deposited to \( p_2 \). The token remains in \( p_2 \) during the processing time. At the end of this time interval, transition \( t_e \) fires; the token is removed from \( p_2 \) and is deposited in \( p_3 \).

A Timed Petri Net is a pair \((PN, \mu)\), where \(PN\) is a Petri Net and \(\mu\) is a function that assigns a positive real number to each transition of the net [Ramachandani (1974)].

The rules of operation of Timed Petri Nets are the same as for ordinary Petri Nets, with the addition of the following phases of transition firing:

1. The firing initiation, which occurs when the transition is enabled. When the firing is initiated, a token is removed from each of the input places of the transition.

2. The firing execution which takes time equal to the processing time.

3. The firing termination, which occurs at the end of the execution. At the termination, one token is deposited to each of the output places of the transition.

### 2.14 Resource Availability Place

When a procedure or function, represented by a transition, completes its processing, it is ready to accept a new set of input data. A place, called the resource availability
place (Figure 2.18), is used to depict the fact that a transition has completed its execution and it is ready to process the next token. Thus, when the resource availability place is empty, it disables a new firing of the transition which is currently processing. The introduction of the resource availability place creates a self loop, which cannot be represented in the incidence matrix. Consequently there exist two choices. Either remove the self loops while understanding that they exist, or replace the resource availability place by two places and a transition in between (Figure 2.19).

Similarly, when a set of functions is performed by a decisionmaker, a place which is the output of the last transition and the input to the first transition allocated to the decisionmaker, is used to represent that the decisionmaker has completed the processing and is ready to accept a new set of input data (Figure 2.20).
Figure 2.20: Resource availability place for a decisionmaker

In the case of a decisionmaker with multiple transitions (i.e. performing multiple functions), the number of tokens in the initial marking of the resource availability place represents the number of functions (processes) that the decisionmaker may perform at the same time. In this thesis, we assume that the decisionmakers can perform only one task at a time, and hence, the maximum number of tokens in the resource availability place is one.

In the case of a single transition, the number of tokens in the resource availability place of the self loop represents the maximum number of data that can be simultaneously processed by the transition.

2.15 Synchronization Loop

When two or more decisionmakers receive information from the source, it may be desirable to synchronize the instant at which they make their observations, so that when they exchange information, they refer to the same environment state. To accomplish this synchronization, a synchronizing transition is introduced that is enabled when all decisionmakers have completed their processing, and a feedback loop is added for each decisionmaker that enables his first transition. Figure 2.21 depicts a Petri Net without the synchronization loops, while Figure 2.22 depicts the same net with the loops present.
Figure 2.21: Petri Net without synchronization of observations

Figure 2.22: Petri Net with synchronization of observations
2.16 Lines and Slices of Occurrence Nets

An occurrence net is a Petri Net that is acyclic (has no loops), and each place of the net has at most one input and at most one output transition. Thus an occurrence net is an event graph that contains no loops [Best (1979)].

Let $\prec$ denote the partial order, defined as follows: Let $a$ and $b$ be any two nodes (places or transitions) of the net, then:

$a \prec b$ iff there exists a directed path from $a$ to $b$

and

$a \not\prec b$ iff there does not exist any directed path from $a$ to $b$

A line $L$ is a set of nodes (places or transitions) such that for any two nodes $a$ and $b$ in $L$, $a \prec b$ or $b \prec a$, and there does not exist any node $c$ in the net, which does not belong to the line $L$, such that it satisfies $c \prec d$ or $c \prec d$ with any node $d$ in $L$.

A slice $S$ is a set of nodes (places or transitions) such that for any two nodes $a$ and $b$ of the slice, $a \not\prec b$ and $b \not\prec a$, and there does not exist a node $c$ in the net, which does belong to the slice $S$, such that it satisfies $c \not\prec d$ and $d \not\prec c$, with any node $d$ in $S$.

The elements of a line are strictly ordered and characterize the serial (sequential) operations of the net, while the elements of a slice are unordered and characterize the concurrent activities of the net, i.e. activities that may occur simultaneously.

A Petri Net and its slices are depicted in Figure 2.23, while its lines are shown in Figure 2.24.
Figure 2.23: Slices of Petri Net

Figure 2.24: Lines of Petri Net
2.17 Chapter Summary

In this chapter, the fundamental concepts and results of Petri Net theory have been presented. Switches, macro-nodes, subnets, information flow paths and resource availability places are of special interest for the modeling of decision-making organizations.

Switches enable us to model the selection of one procedure from a set of alternative procedures, while macro-nodes and subnets are used to aggregate nodes and to use results valid for event graphs for the analysis of nets which contain switches.

Information flow paths are used to identify the sequentially executed procedures of the organization, and to compute the corresponding processing time along each simple path, while complete paths are used to compute the total response time of the organization. The resource availability places allow us to depict the function allocation to decisionmakers, and to create the directed elementary circuits that are used for the throughput rate computation of the organization (Chapter 4).
Chapter 3

INFORMATION THEORY FUNDAMENTALS

3.1 Introduction

Information theory [Raisbeck (1963), Shannon and Weaver (1963)] is used to compute the workload (total activity) of the information processing and decision-making functions, performed by the human decisionmakers. In this chapter the fundamental definitions of Information theory are reviewed, and the Partition Law of Information is presented. The human decisionmaker is assumed to be well trained. Each information processing function is assumed to be precisely defined and described by an algorithm. Some information processing functions may be implemented by more than one algorithm. In this case the decisionmaker is free to select which of the available algorithms to use. The computation of the workload (total activity) of a decision-making function implemented by a set of alternate algorithms, is based on the result obtained by Boettcher (1981). A detailed example is presented in Appendix B to illustrate the application of Information theory to the computation of the total activity of decision-making algorithms.

3.2 Entropy and Transmission

The entropy \( H(w) \) of the discrete random variable \( w \) is defined as:

\[
H(w) = - \sum_{i=1}^{n} \left[ p(w = w_i) \log p(w = w_i) \right]
\]  

(3.1)

If the logarithmic base is 2, then the entropy is measured in bits. The entropy quantifies the uncertainty about the value taken by the random variable \( w \).
The total activity $G$ of a system of variables $S = \{w_1, w_2, \ldots, w_n\}$ is defined by:

$$G = \sum_{i=1}^{n} H(w_i)$$  \hspace{1cm} (3.2)

The conditional entropy of variable $x$, when variable $y$ is known, $H_y(x)$, is defined by:

$$H_y(x) = - \sum_{i=1}^{n} p(y = y_i) \sum_{j=1}^{k} p(x = x_j | y_i) \log p(x = x_j | y_i)$$  \hspace{1cm} (3.3)

The conditional entropy quantifies the uncertainty about the value of $x$, when the value of $y$ is known.

The joint entropy of variables $x$ and $y$ is given by

$$H(x, y) = - \sum_{i=1}^{n} \sum_{j=1}^{k} p(x = x_i, y = y_j) \log p(x = x_i, y = y_j)$$  \hspace{1cm} (3.4)

The joint entropy quantifies the uncertainty of the value taken by the pair of random variables $x$ and $y$.

A useful identity is:

$$H(x, y) = H(x) + H_z(y)$$  \hspace{1cm} (3.5)

This identity is obtained as follows:

$$H(x, y) = - \sum_{x} \sum_{y} p(x, y) \log p(x, y)$$
$$= - \sum_{x} \sum_{y} p(x, y) \log [p(y|x)p(x)]$$
$$= - \sum_{x} \sum_{y} p(x, y) \log p(y|x) - \sum_{x} \sum_{y} p(x, y) \log p(x)$$
$$= - \sum_{x} \sum_{y} [p(y|x)p(x)] \log p(y|x) - \sum_{x} \log p(x) \sum_{y} p(x, y)$$
$$= - \sum_{x} p(x) \sum_{y} p(y|x) \log p(y|x) - \sum_{x} [\log p(x)] p(x)$$
$$= H_z(y) + H(x)$$
The mutual information or throughput $T(x : y)$ of two variables $x$ and $y$ is defined as:

\begin{align}
T(x : y) &= H(x) + H(y) - H(x,y) = T(y : x) \\
T(x : y) &= H(x) - H_y(x) = H(y) - H_x(y) = T(y : x)
\end{align}

Mutual information measures the relatedness of variables $x$ and $y$; it quantifies how much information is obtained about $x$, when $y$ is known and vice versa. If the variables are independent, then the mutual information is zero.

The mutual information for $n$ variables [McGill (1954)] is defined by:

$$T(x_1 : x_2 : \ldots : x_n) = \sum_{i=1}^{n} H(x_i) - H(x_1, x_2, \ldots, x_n)$$

### 3.3 Partition Law of Information

Consider a system with $n$ internal variables $w_i$, $i = 1, 2, \ldots, n$. Let $x$ represent the input, and $w_n$ represent the output $y$ of the system (Figure 3.1).

The Partition Law of Information [Conant (1976)] states that: the total activity $G$ of a system of variables is the sum of throughput $G_t$, blockage $G_b$, coordination $G_c$ and noise $G_n$.

\begin{align}
\sum_{i=1}^{n} H(w_i) &= T(x : y) + T_y(x : w_1, w_2, \ldots, w_{n-1}) + T(w_1 : w_2 : \ldots : y) \\
&\quad + H_x(w_1, w_2, \ldots, w_{n-1}, y) = G_t + G_b + G_c + G_n
\end{align}
where

\[ G_t = T(x : y) \] is the transmission or throughput between \( x \) and \( y \), and is a measure of the extent to which the uncertainty about the output is reduced by knowledge of the input, or equivalently, the uncertainty about the input is reduced by knowledge of the output.

\[ G_b = T_y(x : w_1, w_2, ..., w_{n-1}) \] is the transmission between the input and the internal variables of the algorithm, conditioned on knowledge of the output \( y \). This is a measure of the extent that the uncertainty about the input may be reduced by knowledge of the internal variables of the system, when the output is already known. In this context, \( G_b \) is a measure of the information about the input that has been blocked by the system and is not present in the output. \( G_b \) is called blockage.

\[ G_c = T(w_1 : w_2 : .. : y) = \sum_{i=1}^{n} H(w_i) - H(w_1, w_2, ..., w_{n-1}, y) \] is the \( N \)-dimensional mutual information of the system variables. \( G_c \) is a measure of the relatedness of the system variables and quantifies the coordination required among the system variables in order to process the input and produce an output. \( G_c \) is called coordination.

\[ G_n = H_z(w_1, w_2, ..., w_{n-1}, y) \] is the uncertainty in the system variables, when the input is known, and quantifies the extent to which the system is not a deterministic function of the input. This term is called noise. It does not necessarily represent the effect of undesirable noise, and in the decision-making context it is interpreted as the entropy of decision variables that are not deterministic functions of the input.

The Partition Law of Information is developed as follows:

\[
\sum_{i=1}^{n} H(w_i) = T(x : y) + T_y(x : w_1, w_2, ..., w_{n-1}) + T(w_1 : w_2 : .. : y) + H_z(w_1, w_2, ..., w_{n-1}, y)
\]

By definition

\[ T(w_1 : w_2 : .. : w_n) = \sum_{i=1}^{n} H(w_i) - H(w_1, w_2, ..., w_n) \quad (3.10) \]
Also
\[ T(x : w_1, w_2, .., w_n) = H(w_1, w_2, .., w_n) - H_x(w_1, w_2, .., w_n) \] (3.11)

Since \( w_n = y \)
\[ T(x : w_1, w_2, .., w_n) = T(x : w_1, w_2, .., y) = T(x : y) + T_y(x : w_1, w_2, .., w_{n-1}) \] (3.12)

Combine the last two equations to obtain
\[ H(w_1, w_2, .., w_n) = H_x(w_1, w_2, .., w_n) + T(x : y) + T_y(x : w_1, w_2, .., w_{n-1}) \] (3.13)

Thus,
\[
\sum_{i=1}^{n} H(w_i) - T(w_1 : w_2 : .. : w_n) = H_x(w_1, w_2, .., w_n) \\
+ T(x : y) + T_y(x : w_1, w_2, .., w_{n-1})
\] (3.14)

An important decomposition of the input is obtained as follows:
\[ T(x : w_1, w_2, .., w_{n-1}) = H(x) - H_{w_1, w_2, .., w_{n-1}}(x) \] (3.15)

Apply the rule of uniform subscripts [Conant (1976)], which states that by conditioning all the quantities of an information theoretic identity on the same variable, another identity is obtained.
\[ T_y(x : w_1, w_2, .., w_{n-1}) = H_y(x) - H_{y, w_1, w_2, .., w_{n-1}}(x) \] (3.16)
\[ H_y(x) = T_y(x : w_1, w_2, .., w_{n-1}) + H_{y, w_1, w_2, .., w_{n-1}}(x) \] (3.17)

also
\[ T(x : y) = H(x) - H_y(x) \]
\[ = H(x) - [T_y(x : w_1, w_2, .., w_{n-1}) + H_{y, w_1, w_2, .., w_{n-1}}(x)] \] (3.18)
\[ H(x) = T(x : y) + T_y(x : w_1, w_2, .., w_{n-1}) + H_{y, w_1, w_2, .., w_{n-1}}(x) \] (3.19)

The first and second terms are recognized as the throughput and the blockage. The third term quantifies the uncertainty about the input when the output and the internal variables of the algorithm are known. Thus, it is the part of the input that is rejected by the system (algorithm) and is called the rejection \( G_r \).
\[ H(x) = G_t + G_b + G_r \] (3.20)
Consider an algorithm which copies its input in the first internal variable \( w_1 \), i.e. \( w_1 = x \). Then \( G_r = 0 \), and

\[ H(x) = G_t + G_b \]

(3.21)

For a deterministic system, the variables \( w_i, i = 2,3,\ldots n \) are completely determined, when the copy of input \( w_1 = x \) is known. Thus

\[ H_{w_1}(w_2, w_3, \ldots, w_n) = 0 \]

(3.22)

and hence

\[
G_c = \sum_{i=1}^{n} H(w_i) - H(w_1, w_2, \ldots, w_n)
\]
\[
= \sum_{i=1}^{n} H(w_i) - [H(w_1) - H_{w_1}(w_2, \ldots, w_n)]
\]
\[
= \sum_{i=1}^{n} H(w_i) - H(w_1) = \sum_{i=2}^{n} H(w_i)
\]

(3.23)

Finally, since there is no remaining uncertainty about the input when the output and the internal variables are known, the noise term is zero: \( G_n = 0 \) and the Partition Law of Information for deterministic systems becomes

\[ G = G_t + G_b + G_c \]

(3.24)

### 3.4 Computation of Total Activity

The total activity, \( G \), of a function implemented by one deterministic algorithm is defined as the sum of the entropies of the algorithm’s internal variables:

\[
G = \sum_{i=1}^{n} H(w_i) = H(w_1) + \sum_{i=2}^{n} H(w_i)
\]
\[
= H(x) + \sum_{i=2}^{n} H(w_i) = H(x) + G_c
\]

(3.25)

where \( G_c = \sum_{i=2}^{n} (w_i) \) is the coordination of the algorithm.

A decisionmaker may have several alternate algorithms available to process the information. In this case, a decision variable \( u \) is used to represent the algorithm selection
Figure 3.2: Function implemented by alternative algorithms

(Figure 3.2). Assume that the selection of the algorithms is independent of the value of the input $x$, i.e. that there is a probability $p_i = p(u = i)$ associated with algorithm $i$, representing the relative frequency of use of algorithm $i$. For $k$ deterministic algorithms, used alternatively to implement the function, the total activity $G$ of the function is:

$$G = H(x) + \sum_{i=1}^{k} [p_i G_{ci} + a_i \lambda(p_i)] + H(z) + H(u)$$  \hspace{1cm} (3.26)

where $G_c = \sum_{i=1}^{k} [p_i G_{ci} + a_i \lambda(p_i)] + H(z)$ is the coordination of the algorithm variables, and $w_1 = x$ is the copy of the input.

- $G_n = H(u)$ is the noise
- $p_i$ is the probability of use of algorithm $i$
- $G_{ci}$ is the coordination of algorithm $i$
- $a_i$ is the number of internal variables of algorithm $i$
- $z$ is the output of the function, i.e. is a variable that takes the value $y_i$ when algorithm $i$ is active
\( H \) is the entropy of a binary variable defined by:

\[
H(p) = -p \log_2 p - (1 - p) \log_2 (1 - p)
\]  

(3.27)

The derivation of equation (3.26) is presented in Appendix A.

If the selection of algorithms depends on the value of the input \( x \), i.e. algorithm \( i \) is selected with probability \( p(u = i|x) \) conditioned on the value of \( x \), then the total activity is similarly computed. It should be noted, however, that the relative frequency of algorithm use, and the probability mass functions of the input to each algorithm are affected by the conditioning of the probability on the value of the input. Thus, Bayes' rule should be used to evaluate the probability of use of algorithm \( i \), \( p_i \), and rescale the pmf of the input to each algorithm.

\[
p_i = p(u = i) = \sum_{j=1}^{m} [p(u = i|x = x_j)p(x = x_j)]
\]  

(3.28)

and

\[
p(x = x_j|u = i) = \frac{p(x = x_j, u = i)}{p(u = i)} = \frac{p(u = i|x = x_j)p(x = x_j)}{p(u = i)}
\]  

(3.29)

Finally, the noise term \( G_n \), in this case, is

\[
G_n = H_x(u)
\]  

(3.30)

3.5 Modeling of Input Tasks

In order to use Information theory to quantify the activity of decision-making procedures, the source that generates the input is assumed to be memoryless, stationary and ergodic. When modeling the input tasks, all combinations of the input element values must be considered, so that all the decision modes of the decision-making process are excited. It should be noted that not all values obtainable by an input variable
(attribute) need be included, but only one value from each representative range. These representative ranges are identified by considering the thresholds that are used in the decision-making and information processing functions. Finally, the probability of occurrence of each input value must be selected to reflect the relative frequency with which the input takes on the corresponding values.

In order to compute the entropies involved in the activity computation, the probability mass functions of all the variables must be obtained. This computation is performed by software, which exercises the decision-making process for each input value and keeps track of the values of the algorithms' variables and their respective frequency.

3.6 Chapter Summary

In this Chapter, the fundamental Information theoretic quantities and identities have been reviewed. The Partition Law of Information has been discussed, and the Throughput, Blockage, Rejection, Coordination and Noise terms have been interpreted in the context of decision-making. Finally, the computation of the total activity of a function implemented by a set of alternative algorithms has been presented.
Chapter 4

ANALYSIS OF $C^2$ ORGANIZATIONS

4.1 Introduction

The properties that characterize decision-making organizations are quantified by the corresponding Measures of Performance (MOPs). MOPs for Command and Control organizations include accuracy, response time, and throughput rate. In this chapter, the model of the Command and Control process is described. Then the measures of performance are defined, and the procedures for their computation are presented. Finally, the effectiveness of $C^2$ organizations is discussed and a Measure of Effectiveness (MOE) is defined.

4.2 Model of the Command and Control Process

The Command and Control process may be decomposed into functions (Figure 4.1). These functions are: environment surveillance, situation assessment, development of courses of action, response selection, planning and direction [Lawson (1981)].

Information about the environment is collected by sensors; subsequently it is processed in order to assess the state of the environment. In the situation assessment stage, the data or signals received from the sensors are used to develop hypotheses about the state of the environment. Signals received passively connote the existence of the corresponding transmitters, while signals received actively connote the existence of objects. Hypotheses about the state of the environment are formed and the data received is used to evaluate the likelihood of each hypothesis. A threshold is used to accept or reject the hypothesis. When the likelihood of one hypothesis exceeds the
Figure 4.1: Model of the Command and Control process
threshold, the hypothesis is accepted as true, and the processing continues with the development of courses of action. It is evident that the situation assessment stage is a judgement process as opposed to a pure decision-making process.

In the next function, courses of action are developed based on the perceived state of the environment. The objective is to bring the state of the environment to the desired state, from the organization’s point of view, in order to accomplish the mission.

The response selection stage evaluates the courses of action and selects the one that is feasible and has maximum utility (payoff). Finally, in the planning and direction stage, the logistics of the selected response are evaluated and directions for its implementation are given to the appropriate personnel.

If the Command and Control process is distributed, i.e., performed by more than one processing channels, there may exist interaction among the channels. In this case, the exchange of information may exist at several levels of the process: raw data exchange, hypotheses exchange, most likely hypothesis exchange, courses of action exchange, or selected course of action exchange (Figure 4.2).

It should be noted that the functions of a processing channel may be carried out by several decisionmakers belonging to the decision-making organization and possibly by a set of system components such as decision aids and databases. The connectivity of these functions is therefore an important feature of the process implementation.

The functions of the Command and Control process, or in general of any information processing and decision-making process, can be modeled by algorithms. It is plausible that there exist several procedures for the implementation of a function. These alternate procedures may employ different processing algorithms or may involve different actions such as the use of decision aid, or the access of a database. The difference in the processing algorithms may be in their complexity and accuracy. One algorithm may be more complex, computationally more demanding, and presumably more accurate, while another may be less complex and possibly less accurate.

The deviations of the results of the information processing from the optimal or desired results, may be due to a variety of reasons. Errors may be introduced when
Figure 4.2: Interaction of processing channels
the decisionmakers utilize approximations (approximate algorithms). In the case of algorithms that use parameters such as thresholds, the use of different parameter values may lead to deviations from the correct assessment and optimal response. Finally not using a decision aid or not accessing a database may result in inferior assessment and response selection.

4.3 Model of Command and Control Organizations

Block diagrams may be used to depict the connectivity of the functions of the $C^2$ process of one or more processing channels. The drawback in using block diagrams is their inability to denote the presence of data and information at the various stages of processing. Consequently, the coordination of the processing, as well as the parallelism of the function execution cannot be demonstrated.

Petri Nets can be used to depict the process as well as the presence of data and the concurrency of operations. In modeling $C^2$ organizations, the transitions represent processes that carry out the required Command and Control functions. The flow of tokens in the Petri Net represents the flow of information through the decisionmaking organization.

Switches are used to represent the selection of one algorithm from a set of algorithms that perform the same function (Figure 4.3). A decisionmaker may or may not access a database or use a decision aid. The Petri Net model for decision aid or database access is shown in Figure 4.4, while in Figure 4.5. a switch is added to depict the choice of processing the data with or without the decision aid - database access.

The selection of one procedure versus another is the result of implementing a decision strategy. Thus, a decision strategy may forego the use of a decision aid as opposed to using the decision aid, or may select one processing algorithm versus another. The selection of a specific procedure from a set of procedures can be modeled either as a stochastic event, or can be conditioned upon the value of the data and the type of the information that are input to the function. A combination of switch settings determines
Figure 4.3: Petri Net of alternate procedure use

decision aid or database

Figure 4.4: Petri Net of decision aid - database access
Figure 4.5: Petri Net with decision switch selecting decision aid - database access

which procedures are active, i.e., process data.

4.4 Decision Strategies

Consider a function which can be implemented by \( k \) algorithms \( f_i, i = 1, 2, \ldots, k \). Let \( u \) be the decision variable which selects the algorithm to be used. A decision strategy is defined by the vector \( d \):

\[
d^T = [p(u = 1) \ p(u = 2) \ \cdots \ p(u = k)]
\]

(4.1)

such that

\[
\sum_{j=1}^{k} p(u = j) = 1
\]

(4.2)

A pure strategy, \( D_j \), is a strategy that selects always the same algorithm, i.e., \( p(u = j) = 1 \) for a specific \( j \) and \( p(u = i) = 0 \) for \( i \neq j \).

\[
D_j^T = [p(u = 1) \ \cdots \ p(u = j) \ \cdots \ p(u = k)]
\]

\[
= [0 \ \cdots \ 1 \ \cdots \ 0]
\]

(4.3)
Note that there exist exactly $k$ pure strategies. Any strategy $d$ can be obtained by linear combination of the pure strategies

$$d = \sum_{j=1}^{k} \delta_j D_j$$

$$\sum_{j=1}^{k} \delta_j = 1$$

The strategies thus obtained are called mixed strategies [Owen [1968]], if

$$\delta_j \neq 1 \quad \text{for all } j$$

Note that $\delta_i$ corresponds to the probability (relative frequency) of use of pure strategy $D_i$.

Consider a function which can be implemented by $n$ algorithms and let $v$ to be the decision variable which selects the algorithm $f_i$, conditioned on the value of the input $x$ to the function. Let the input $x$ obtain $m$ discrete values. Then a decision strategy is defined by a matrix $d$

$$d = \begin{bmatrix}
    p(u = 1|x = x_1) & p(u = 1|x = x_2) & \cdots & p(u = 1|x = x_m) \\
    p(u = 2|x = x_1) & p(u = 2|x = x_2) & \cdots & p(u = 2|x = x_m) \\
    \vdots & \vdots & \ddots & \vdots \\
    p(u = n|x = x_1) & p(u = n|x = x_2) & \cdots & p(u = n|x = x_m)
\end{bmatrix}$$

such that

$$\sum_{i=1}^{n} p(u = i|x = x_j) = 1$$

A pure decision strategy is defined when

$$p(u = i|x = x_j) = 1$$

for a specific $i$ given that $x = x_j$, and

$$p(u \neq i|x = x_j) = 0$$
Thus, for each column of the matrix, there exist \( n \) choices to assign the value 1 to one element of the column. This assignment to the elements of any column is independent of the selection for the other columns; hence, there exist \( n^m \) combinations, and consequently, there exist \( n^m \) pure strategies.

Let \( D_j \) be a pure strategy. Then, mixed strategies can be obtained by linear combination of pure strategies.

\[
d = \sum_{j=1}^{n^m} \delta_j D_j
\]

\[
\sum_{j=1}^{n^m} \delta_j = 1
\]  

(4.11)  

(4.12)

For example if the input obtains two discrete values \( (m = 2) \), and there are two processing algorithms \( (n = 2) \), the matrix \( d \) is

\[
d = \begin{bmatrix}
    p(u = 1|x = x_1) & p(u = 1|x = x_2) \\
    p(u = 2|x = x_1) & p(u = 2|x = x_2)
\end{bmatrix}
\]

(4.13)

and the number of pure strategies is \( n^m = 2^2 = 4 \). The four pure strategies are:

\[
D_1 = \begin{bmatrix}
    1 & 1 \\
    0 & 0
\end{bmatrix}
\]

(4.14)

\[
D_2 = \begin{bmatrix}
    1 & 0 \\
    0 & 1
\end{bmatrix}
\]

(4.15)

\[
D_3 = \begin{bmatrix}
    0 & 1 \\
    1 & 0
\end{bmatrix}
\]

(4.16)

\[
D_4 = \begin{bmatrix}
    0 & 0 \\
    1 & 1
\end{bmatrix}
\]

(4.17)
The mixed strategy for $\delta_1 = 0.4, \delta_2 = 0.3, \delta_3 = 0.2, \delta_4 = 0.1$, is

$$
\mathbf{d} = \begin{bmatrix}
0.7 & 0.6 \\
0.3 & 0.4
\end{bmatrix}
$$

(4.18)

Consider a decision-making organization with two functions that may be implemented by alternate algorithms. Let the first function's decision variable select the algorithms independently from the value of the input $x$. Let the input $z$ to the second function be the output of the first function (Figure 4.6). Assume that the decision variable of the second function selects the algorithms with probability conditioned on the value of the input $z$ to the function.

Assume that the alphabet of $z$ is $[z_1, z_2, \ldots, z_m]$ i.e., $z$ obtains $m$ discrete values, whereas each algorithm $f_i$ of the first stage has output $y_i$ with alphabet a subset of $[z_1, z_2, \ldots, z_m]$. Let $m_i$ be the number of discrete values of $y_i$. When algorithm $f_i$ is active, the second stage has $\xi_i$ pure strategies, where

$$
\xi_i = n^{m_i}
$$

(4.19)
and the total number $N$ of pure strategies is

$$N = \sum_{i=1}^{k} \xi_i = \sum_{i=1}^{k} n^{m_i}$$  \hspace{1cm} (4.20)

where $k$ is the number of alternate algorithms of the first function.

Note that if all $m_i$ are equal to $m$ then

$$N = \sum_{i=1}^{k} n^{m_i} = kn^m$$  \hspace{1cm} (4.21)

A *behavioral* strategy of the organization is defined when the individual decision-makers select their mixed strategies. When the weighting coefficients $\delta_i$ are discretized, and are therefore restricted to a finite number, the number of behavioral strategies obtained is finite.

### 4.5 Workload Computation

Workload of the individual organization members is an attribute of interest in the analysis and design of $C^2$ organizations. Workload represents the amount of mental effort expended by the decisionmakers in order to perform their assigned tasks. Since the stimuli (inputs) of the $C^2$ organization are not deterministic, there exists uncertainty of their nature and attributes. This uncertainty requires that the decisionmakers have available appropriate procedures to assess the situation and select a response. The model developed by Boettcher and Levis postulates that the decisionmaker is well trained and can select among several procedures in order to process the available information.

The analytical framework for workload computation is N-dimensional Information Theory. The information processing workload is quantified by the total activity of the functions performed by the decisionmaker, and the computation follows the procedure described in Chapter 3. In this respect, the activity of each function is computed (in bits/symbol), and then the workload of each decisionmaker is calculated (in
bits/symbol) by summing the activities of the functions performed by the decision-maker.

In order to compute the workload of the members of the decision-making organization, the procedures and the algorithms that implement the functions must be developed in detail, and the tasks to be processed by the organization must be modeled. This entails the selection of their attributes and the probability of their occurrence. The decision-making process is then simulated, i.e., all decision strategies are exercised for all inputs, and the probability mass functions of the algorithm's variables are computed. Then the entropies of all the variables and the total activity of all the functions are computed.

4.6 Computation of Processing Time

The decisionmakers are limited in their capacity to process information and make decisions [Miller (1969), Sheridan and Ferrel (1974)]. This limitation, called bounded rationality, is expressed quantitatively by the rationality threshold $F_o$. $F_o$ is the maximum processing rate of the human decisionmaker, and is expressed in bits/sec. The processing time $t$ required by a process whose total activity is $G$ bits/symbol is computed by dividing the workload by the processing rate, $F$, of the human decisionmaker.

$$t = \frac{G}{F}$$

(4.22)

and has units of sec/symbol. The minimum processing time, $t_o$, corresponds to the maximum processing rate $F_o$.

It should be noted that $F_o$ in general varies among decisionmakers and may also vary for a specific decisionmaker, as a function of stress, fatigue, environmental conditions and even time of the day.

The probability density function of the rationality threshold may be available from experimental data, or can be assumed to be known. Let the probability density function (pdf) be $h(F_o)$. Then the pdf $q(t_o)$ for the minimum processing time may be computed as follows: Let $H(F_o)$ and $Q(t_o)$ be the corresponding cumulative distribution functions.
Then

\[ Q(t_o^*) = P(t_o \leq t_o^*) = P\left( \frac{G}{F_o} \leq \frac{G}{F_o^*} \right) \]
\[ = P(F_o \geq F_o^*) = 1 - H(F_o^*) = 1 - H\left( \frac{G}{t_o^*} \right) \]  \hspace{1cm} (4.23)

Thus for any \( t_o \)

\[ Q(t_o) = 1 - H\left( \frac{G}{t_o} \right) \]  \hspace{1cm} (4.24)

\[ Q'(t) = -H'(\frac{G}{t}) \]  \hspace{1cm} (4.25)

\[ q(t) = \left( \frac{G}{t^2} \right) h(G) \]  \hspace{1cm} (4.26)

The pdf \( q(t) \) will be used in all subsequent calculations of the Measures of Performance of the \( C^2 \) organization, namely its response time and throughput rate. This implies that the decisionmakers are always allowed enough time to process the information and they are not overloaded, i.e., the tasks that they are assigned to perform do not require them to process information at a rate higher than the rationality threshold.

### 4.7 Timeliness

Timeliness expresses the ability of \( C^2 \) organizations to respond to an incoming stimulus or task within an allotted time. The allotted time (or window of opportunity) is a time interval \( (T_{\text{min}}, T_{\text{max}}) \) defined by the mission (Figure 4.7). \( T_{\text{max}} \) is a threshold such that, if the \( C^2 \) organization issues commands in response to the input after the threshold, there will not be enough time to execute the response.

The time elapsed between the instant an input is received and an output is produced by the \( C^2 \) organization is the response time or time delay \( T_r \). The expected response time (time delay) is a measure of performance that can be used to assess the timeliness of \( C^2 \) organizations. If the expected response time is within the interval \( (T_{\text{min}}, T_{\text{max}}) \), the \( C^2 \) organization’s response is timely.

72
Figure 4.7: Window of opportunity

Alternatively, we may be interested in computing the maximum processing times, corresponding to the minimum processing rate threshold \((F_o)_{\min}\), and use these processing times to compute the maximum response time \((T_r)_{\max}\) of the organization. Again, this computation implies that the decisionmakers will not be overloaded when performing their tasks. In this case, if \((T_r)_{\max} < T_{\max}\), the response is timely.

These measures of performance do not take into account the variance of the response time. Another measure of performance is the probability that the response time, \(T_r\), lies inside the interval \((T_{\min}, T_{\max})\), i.e., \(P(T_{\min} \leq T_r \leq T_{\max})\) [Andreadakis and Levis (1987)]. This measure of performance is better than the expected time delay because it incorporates the effect of the time delay's variance on Timeliness. The probability density function (pdf) of the response time \(T_r\) must be computed in order to evaluate the measure \(P(T_{\min} \leq T_r \leq T_{\max})\).
If a decision-making organization has more than one outputs, then to each output corresponds a different set of complete paths; the corresponding response time and measure of timeliness are computed for each output.

In the case of deterministic delays, the total response time is computed as follows: For each simple information flow path add the processing times of the transitions that belong to the path; the total delay of the complete path is the maximum delay of its simple paths. Then weigh the total delays of the complete paths by the corresponding probabilities.

In the case of stochastic delays, the pdf of the response time is computed as follows: All the information flow paths and the groups of concurrently active information flow paths are identified using one of the algorithms referenced in Chapter 2. Pdfs are then computed for the processing times of the functions performed by the decisionmakers, according to the procedure described in the previous section. Pdfs of the time delays are assigned to the communication processes of the organization, as well as to the decision aid and database access. In the sequel, it is understood that each algorithm, communication process, and database or decision aid access is represented by a transition in the Petri Net of the organization.

For two cascaded procedures with delay pdfs \( f(t) \) and \( g(t) \), the total delay is the sum of the two delays corresponding to the two procedures. Therefore, the pdf of the total delay, \( h(t) \), is given by the convolution of \( f(t) \) and \( g(t) \)

\[
h(t) = \int_0^\infty f(\tau)g(t - \tau)d\tau
\]

For two procedures with delay pdfs \( f(t) \) and \( g(t) \) on parallel concurrently active paths (Figure 4.8), the total delay is the maximum of the delays of the two procedures. Let \( t_1 \) and \( t_2 \) be the delays of the two procedures; then

\[
t = \max(t_1, t_2)
\]

is the total delay (Figure 4.8). If the two delays are independent random variables
Figure 4.8: Computation of pdf of maximum of two stochastic delays

\[
P(t_0 \leq t \leq t_o + dt) = P(t_o \leq t_1 \leq t_o + dt)P(t_2 < t_o) + P(t_o \leq t_1 \leq t_o + dt)P(t_o \leq t_2 \leq t_o + dt) + P(t_o \leq t_2 \leq t_o + dt)P(t_1 < t_o) = f(t_o)dt \ G(t_o) + f(t_o)dt \ g(t_o)dt + g(t_o)dt \ F(t_o) \quad (4.29)
\]

where \( F(t) \) and \( G(t) \) are the cumulative probability functions of \( f(t) \) and \( g(t) \), respectively. Hence, the pdf of the total delay, \( h(t) \), is given by:

\[
h(t) = f(t)G(t) + F(t)g(t) \quad (4.30)
\]

Some simple information flow paths have common transitions, and hence, the corresponding time delays are correlated. In this case the pdf of the maximum of the two delays is computed as follows: Let \( r \) be the processing time of the common transitions, with pdf \( f(t) \), and \( \tau_1 \) and \( \tau_2 \) be the time delays of the non-common transitions with corresponding pdfs \( g_1(t) \) and \( g_2(t) \). The maximum time delay \( t \) is

\[
t = max(\tau + \tau_1, \tau + \tau_2) = \tau + max(\tau_1, \tau_2) \quad (4.31)
\]

75
and its pdf $h(t)$ is obtained by convolving $f(t)$ and $g(t) = g_1(t)G_2(t) + g_2(t)G_1(t)$.

$$h(t) = f(t) * g(t)$$  \hspace{1cm} (4.32)

For two complete paths with delay pdfs $f(t)$ and $g(t)$, that are active with probability $p_1$ and $p_2 = 1 - p_1$, the total delay is the probability weighted sum of the delays of the two complete paths. Consequently, the pdf of the total delay, $h(t)$, is given by:

$$h(t) = p_1 f(t) + p_2 g(t)$$  \hspace{1cm} (4.33)

To compute the pdf of the response time of an organization, first compute the pdf of the total delay of each complete path, applying equation (4.30) or (4.32) to the pdfs of the total delay across the simple information flow paths of the complete path. Then compute the probability weighted sum of the pdfs of the complete paths.

The measure of timeliness, $T$, is defined by:

$$T = P(T_{min} \leq t \leq T_{max}) = H(T_{max}) - H(T_{min})$$  \hspace{1cm} (4.34)

where $H$ is the cumulative distribution function of the response time.

### 4.8 Computation of Throughput Rate

The throughput rate of the $C^2$ Organization is defined as the processing rate that can be maintained without queueing of the input tasks, and without queueing of information at any stage of processing.

The throughput rate of the organization is equal to the minimum of the processing rates of the sets of functions performed by individual decisionmakers and the processing rates of sets of functions of several decisionmakers processing information in an interleaved pattern. These sets of functions correspond to the transitions of the directed elementary circuits of the net. Figure 4.9 depicts an Event graph with two resource availability places delimiting the function allocation to two decisionmakers. The three directed elementary circuits are shown in Figure 4.10.
Figure 4.9: Event Graph

Figure 4.10: Directed Elementary Circuits
The processing rate of a directed elementary circuit is the inverse of the total processing time of the circuit. The total processing time, \( t \), of a directed elementary circuit is equal to the sum of the processing times of its transitions, divided by the token content, \( C \), of the circuit [Ramchandani (1974)]. The token content is equal to the sum of the tokens in the initial marking, or equivalently, to the sum of tokens initially placed in the resource availability places.

\[
    t = \frac{\sum_i t_i}{C} \quad (4.35)
\]

Recall that in this thesis, the decision-makers are assumed to perform one task at a time. In this case, the token content of each circuit is equal to the number of resource availability places that belong to the circuit.

To compute the throughput rate, in the case of deterministic delays, first remove the synchronization loops (if present), and then obtain the directed elementary circuits. Compute the total processing time of each directed elementary circuit, and obtain the inverse of the maximum of the circuit processing times.

In the case of stochastic processing times, the pdf \( g(r) \) of the processing rate \( r \) of a set of functions, when the pdf of the corresponding total processing time \( t \) is known, is computed as follows: Let \( G(r) \) be the cumulative distribution function of \( r \). Then

\[
    G(r^*) = P(r \leq r^*) = P(\frac{1}{t^*} \leq \frac{1}{t}) = P(t^* \leq t) = 1 - P(t \leq t^*) = 1 - F(t^*) = 1 - F(\frac{1}{r^*}) \quad (4.36)
\]

Thus, for any \( r \)

\[
    G(r) = 1 - F(\frac{1}{r}) \quad (4.37)
\]

\[
    G'(r) = -F'(\frac{1}{r}) \quad (4.38)
\]

\[
    g(r) = (\frac{1}{r^2})f(\frac{1}{r}) \quad (4.39)
\]

78
Let \( r_1 \) and \( r_2 \) be the processing rates of two directed elementary circuits and \( f(r) \), and \( g(r) \) be the corresponding pdfs. The processing rate \( r \) is:

\[
 r = \min(r_1, r_2)
\]  
(4.40)

If the processing rates \( r_1 \) and \( r_2 \) are independent random variables, as in the case of directed elementary circuits with no transition in common, then

\[
P(r_o - dr \leq r \leq r_o) = P(r_o - dr \leq r_1 \leq r_o)P(r_2 > r_o) + P(r_o - dr \leq r_1 \leq r_o)P(r_o - dr \leq r_2 \leq r_o) + P(r_o - dr \leq r_2 \leq r_o)P(r_1 > r_o) \\
= f(r_o)dr \ [1 - G(r_o)] + f(r_o)dr \ g(r_o)dr \\
+ g(r_o)dr \ [1 - F(r_o)]
\]  
(4.41)

Hence, the pdf of the processing rate, \( h(r) \), is given by:

\[
h(r) = f(r) \ [1 - G(r)] + [1 - F(r)] \ g(r)
\]  
(4.42)

If the elementary circuits have transitions in common, then their processing rates are correlated. Let two circuits have one transition in common (Figure 4.11) with processing time \( \tau \), with pdf \( g(t) \), and one unique transition in each circuit with corresponding processing times \( \tau_1 \) and \( \tau_2 \), with pdfs \( q_1(t) \) and \( q_2(t) \).

Then the pdf \( h(r) \) of the throughput rate is computed as follows: First compute the pdf \( q(t) \) of the maximum of the processing times, \( \tau^* = \max(\tau_1, \tau_2) \)

\[
q(t) = q_1(t)q_2(t) + q_1(t)q_2(t)
\]  
(4.43)

Then convolve the pdf \( q(t) \) with the pdf \( g(t) \) of the processing time \( \tau \), to obtain the pdf \( f(t) \) of the maximum processing time of the two directed elementary circuits.

\[
f(t) = q(t) * g(t)
\]  
(4.44)

Finally, obtain the pdf \( h(r) \)

\[
h(r) = \left( \frac{1}{\tau^2} \right) f\left( \frac{1}{r} \right)
\]  
(4.45)
The measure $S$ of the processing capacity is defined as the probability that the throughput rate $R$ is greater than the task arrival rate $R_o$

$$S = P(R > R_o) = 1 - H(R_o)$$  \hspace{1cm} (4.46) 

where $H$ is the cumulative distribution function of the throughput rate.

### 4.9 Computation of Accuracy

Accuracy measures the degree to which the actual organization response matches the desired or ideal response. For each input task, a mapping, $L(x)$, known to the organization designer defines the desired response. A cost is assigned to the discrepancy between the actual and desired response. This cost is defined by the organization designer and reflects the cost incurred by the implementation of the suboptimal decision. The cost is computed for each input task and each decision strategy. One accuracy measure is the expected value of the cost which is computed using the probability distributions of the input tasks (Figure 4.12). Let $Y_{d_i}$ be the ideal or desired response, and $Y_i$ be the actual response for each input $x_i$. A cost $C(Y_i, Y_{d_i})$ is computed and the
expected cost $J$ is defined by:

$$J = \sum_i C(Y_i, Y_{d_i}) P(X = x_i) \quad (4.47)$$

The expected cost is affected by the probability assigned to each input task. Thus, the accuracy is sensitive to the modeling of input tasks.

When the organization has multiple outputs, a cost may be associated with each output, and the corresponding accuracy measure, $J_i$, can be computed, so that the designer can evaluate the accuracy of each output separately.

4.10 Effectiveness of $C^2$ Organizations

4.10.1 Organization Locus and Requirements Locus

The Measures of Performance (MOPs) are functions of the organization parameters. In the case of $C^2$ organizations, these parameters include the decisions of individual decisionmakers. These decisions are represented (parameterized) by the weighting coef-
ficients $\delta_j$ in equations (4.5) and (4.12). Obviously, the coefficients $\delta_j$ are real numbers and vary between 0 and 1. For computational simplicity, the values obtained by the coefficients $\delta_j$ are discretized, and thus a finite number of mixed strategies is defined.

The vector, whose elements are the decision strategies of all the decisionmakers of the organization, is the behavioral strategy of the $C^2$ Organization. The set of all possible values of the decision strategies define the decision space. To each behavioral strategy corresponds a value of the vector of MOPs; the set of behavioral strategies determines a set of values in the MOP space (Figure 4.13).

Similarly, if the organization parameters $c$ may take $k$ discrete values $c_1, \ldots, c_k$, with probability $p_1, \ldots, p_k$ respectively, a parameter vector is defined. Parameters of interest may be threshold levels of processing algorithms, or attributes of decision aids and databases.

The strategy vector and the parameter vector may be combined into one vector. As the organization parameters vary, the MOP vector sweeps a locus in the MOP space. This locus is called the organization locus.

The requirements locus is defined in the MOP space as the set of points that satisfy the design requirements. The intersection of the organization locus and the requirements locus is the set of operating points of the organization that satisfy the requirements (Figure 4.14).
4.10.2 Measure of Effectiveness

In order to assess the effectiveness of an organization, the organization's MOPs are compared to the organization's requirements for all behavioral strategies, and all parameter values, i.e., all operating points of the organization. Measures of Effectiveness (MOEs) are quantities that result from this comparison [Levis (1986), Bouthonnier (1982), Karam (1985)].

Each operating point of the organization occurs with a certain probability (relative frequency). The measure of effectiveness, $Q$, can be defined as the probability that the organization satisfies its requirements $R_1, R_2, \ldots, R_k$.

$$Q(R_1, R_2, \ldots, R_k) = P(MOP_i \leq R_i; i = 1, \ldots, k)$$

(4.48)

Thus the MOE is equal to:

$$Q = \sum_{j=1}^{N} \delta_j \times P(\text{organization operates at point } j)$$

(4.49)

where $\delta_j$ is defined by:

$$\delta_j = \begin{cases} 
1 & \text{if } MOP_i \leq R_i \text{ for every } i \text{ when the} \\
& \text{organization operates at point } j \\
0 & \text{otherwise} 
\end{cases}$$

(4.50)

and $N$ is the total number of operating points. Since these probabilities are not known, (they can presumably be evaluated by experiments), we can assume that all operating
points are equally likely. Then

\[ Q = \frac{1}{N} \sum_{j=1}^{N} \delta_j \tag{4.51} \]

For fixed values of the parameters, the Measure of Effectiveness is defined as the ratio of behavioral strategies that satisfy the requirements to the total number of behavioral strategies. It should be noted that the organization may operate at any point of the decision space, as different decisionmakers implement different mixed strategies. In this context, the Measure of Effectiveness is a measure of robustness of the organization with respect to the decision strategies implemented by different decisionmakers.

4.11 Chapter Summary

In this chapter, the measures of performance of Command and Control organizations have been defined, and the respective procedures for their computation have been developed. The key point for the computation of response time and throughput rate is the estimation of the processing time of individual decision-making functions by the ratio of the total activity of each function to the rationality threshold. This approach allows enough time to the decisionmakers to perform their assigned tasks, without being overloaded. Finally, Measures of Effectiveness have been examined, and a Measure of Effectiveness has been defined.
Chapter 5

SYNTHESIS OF $C^2$ ORGANIZATIONS

5.1 Introduction

In this chapter, the synthesis problem of $C^2$ organizations is formulated. The concepts of data flow structure (DFS), decision-making organization (DMO), and $C^2$ organization ($C^2$O) are contrasted, and are employed in order to develop a structured process for the synthesis of $C^2$Os. The DFS is a representation of the connectivity of the functions of the organization and illustrates the flow of information from function to function. The DMO is a DFS whose functions have been allocated to decisionmakers. Finally, a $C^2$O is a DMO which is supported by hardware and software (the $C^3$ system) in the execution of its tasks. A classification of data flow structures, suitable for decision-making organizations, is introduced, and a procedure for the generation of data flow structures is developed. The transformation of data flow structures to decision-making organizations and subsequently to $C^2$ organizations is described. Finally, the rules for the modification of $C^2$ organizations, in order to improve their measure of effectiveness, are presented.

5.2 Synthesis Problem Formulation

This thesis explores the synthesis of Command and Control organizations using the following formulation of the design problem: Given a mission and a set of tasks to be performed, design a $C^2$ organization which is accurate, timely, exhibits a task throughput rate that is higher than the task arrival rate, and whose decisionmakers are not overloaded [Andreadakis and Levis (1987)].
These qualitative design requirements can be stated quantitatively:

Accuracy greater than or equal to a given threshold, or equivalently, expected cost $J$ less than or equal to some threshold $J_o$:

$$J \leq J_o$$  \hspace{1cm} (5.1)

Timeliness measure $T$ greater than or equal to some threshold $T_o$:

$$T \geq T_o$$ \hspace{1cm} (5.2)

Processing capacity measure $S$ greater than or equal to some threshold $S_o$:

$$S \geq S_o$$ \hspace{1cm} (5.3)

The constraints that must be observed are that decisionmakers not be overloaded, i.e., the decisionmakers' information processing rate $F_i$ should be less than or equal to the rationality threshold $F_o$:

$$F_i < F_o \quad \text{for decisionmaker } i$$ \hspace{1cm} (5.4)

The second and third design requirements may also be stated as:

Maximum response time $(T_r)_{max}$ less than or equal to some threshold $T_{ro}$:

$$(T_r)_{max} \leq T_{ro}$$ \hspace{1cm} (5.5)

and task throughput rate greater than the task arrival rate $R_o$:

$$R > R_o$$ \hspace{1cm} (5.6)

The design of $C^2$ organizations includes the following steps:

1. Identification of the functions that must be performed by the organization to execute its task.
2. Design of the data flow structure of the $C^2$ organization.
3. Function allocation to decisionmakers.
4. Selection of the resources (software and hardware) and integration of these resources into the organization structure.
5.3 Synthesis Approach

Given a complex information processing task there exists a multitude of ways to partition the decision-making process into sub-processes (functions), to define the schema of information exchange among the functions, to allocate the functions to decisionmakers, and to specify the supporting system (software and hardware).

The approach taken in this thesis decouples the decomposition of the decision-making process and the exchange of data among the functions from the function allocation to decisionmakers and the selection of the supporting system.

The data flow structure design focuses on information processing schemata, while the organization architecture design focuses on function allocation to decisionmakers and on the development of the decision support systems.

5.4 Description of the Synthesis Methodology

The synthesis methodology has four phases (Figure 5.1).

In Phase 1, the procedure for generating data flow structures produces a set of candidate designs. In Phase 2, each data flow structure is augmented and transformed to one or more $C^2$ organizations, in which the functions have been allocated to decisionmakers, and the supporting hardware and software have been incorporated. In Phase 3, the activity of the individual functions or processes, the accuracy, the processing time, the throughput rate and the measure of effectiveness of each $C^2$ organization are computed. The designs obtained in this manner are revised in Phase 4, to increase their measure of effectiveness by changing the function allocation, introducing decision aids, modifying the databases design, and improving the communication links. Finally, a Command and Control organization is selected from the candidate designs on the basis of the greatest MOE value.
Figure 5.1: Flowchart of the Design Methodology
5.5 Classification of Data Flow Structures

Data flow structures represent the functional decomposition of the information processing, as well as the function connectivity. The flow of data within the organization, and the processing functions are described in general terms in the sequel.

The environment under surveillance is partitioned into sectors. Sensors observe each sector, and collect data. Processing of the data produces assessments. These assessments are in effect estimates of the environment state. The assessments are combined (fused) in order to obtain a global situation assessment, and filter out erroneous assessments. This fusion is called data fusion.

Further processing produces results which are the alternative courses of action that are required to bring the environment to the desired state. The results are fused in order to avoid conflicting courses of action by individual processing channels, and to maximize the coordination of the actions. Finally, a response is selected from the available courses of action.

Depending on the degree of centralization of decision-making at the situation assessment stage, and the magnitude of the geographical area for which global assessment is desired, the data (situation assessment) fusion stage may be more or less complex.

Similarly, depending on the degree of centralization of decision-making at the response selection stage, and the magnitude of the geographical area where the response needs to be coordinated, the results fusion stage may be more or less complex.

The need for redundancy of information within the structure (organization) arises from survivability considerations and topological factors.

The data fusion stage and results fusion stage are pivotal for the classification of data flow structures. The complexity of the information processing is manifested at the fusion stages: the more data or results that are fused, the more complex the processing. The redundancy of information is also manifested at the fusion stages: the more fusion functions that receive the same information, the more redundant the processing.
In order to classify and generate data flow structures, the definition of processing stages (nodes) is required. The processing stages definition capitalizes on the concepts of data fusion and results fusion.

In order to generate data flow structures in a consistent, methodical way, it is advisable to parameterize the designs by the complexity and the redundancy of the information processing.

The Petri Net formalism is used to represent the data flow structures. The processing stages are represented by transitions, whereas the data or information that are input or output of the processing stages are represented by places. The availability of data or information at specific places of the Petri Net is represented by the existence of tokens in the respective places.

In order to describe the information processing, the following processing stages are introduced:

**Initial Processing [IP]:** this stage receives data from the sensors and develops hypotheses (situation assessment).

**Data Fusion [DF]:** this stage receives and combines (fuses) the results of IP.

**Middle Processing [MP]:** this stage follows the DF stage; it evaluates the most likely hypothesis, and develops courses of action.

**Results Fusion [RF]:** this stage combines the results of several MP stages.

**Final Processing [FP]:** this stage operates on the output of the RF stage and selects a response - produces an output.

**Interactions between stages.**

In order to design a data flow structure, the permissible interactions among processing stages must be established. These are:
Table 5.1: Classification of Data Flow Structures

<table>
<thead>
<tr>
<th>flow type</th>
<th>class</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>all information flow paths of type 1, Figure 5.3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>all information flow paths of type 2, Figure 5.4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>all information flow paths of type 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>indistinguishable from class 2, Figure 5.4</td>
</tr>
<tr>
<td>1,2</td>
<td>12</td>
<td>information flow paths of type 1 and 2, Figure 5.5</td>
</tr>
<tr>
<td>1,3</td>
<td>13</td>
<td>information flow paths of type 1 and 3, Figure 5.6</td>
</tr>
<tr>
<td>1,2,3</td>
<td>123</td>
<td>information flow paths of type 1,2 and 3, Figure 5.7</td>
</tr>
</tbody>
</table>

\[
IP \rightarrow DF \text{ or } IP \rightarrow RF \\
DF \rightarrow MP \text{ or } DF \rightarrow FP \\
MP \rightarrow RF \\
RF \rightarrow FP
\]

It should be noted that more than one IP nodes can be connected to one DF node (RF node) and more than one MP nodes can be connected to one RF node, whereas exactly one MP node can follow each DF node and exactly one FP node can follow each RF node (DF node).

Thus, the information flow types are (Figure 5.2):

- flow type 1 \( IP \rightarrow DF \rightarrow MP \rightarrow RF \rightarrow FP \)
- flow type 2 \( IP \rightarrow DF \rightarrow FP \)
- flow type 3 \( IP \rightarrow RF \rightarrow FP \)

The classification is performed on the basis of the data flow types that are combined in the data flow structure, i.e., the interactions that are present in the data flow structure. The feasible combinations and the corresponding classes thus defined are given in Table 5.1, and shown in Figures 5.3 to 5.7.

Note that information flow paths of flow type 2 and 3 cannot exchange data; hence such structures are infeasible.
Figure 5.2: Information Flow Types

Figure 5.3: Class 1 Data Flow Structure
Figure 5.4: Class 2 or Class 3 Data Flow Structure

Figure 5.5: Class 12 Data Flow Structure
Figure 5.6: Class 13 Data Flow Structure

Figure 5.7: Class 123 Data Flow Structure
Given a class and the number of inputs, the data flow structures of the class are characterized by two parameters: the degree of complexity and the degree of redundancy.

Degree of complexity of a data fusion [DF] node (or results fusion [RF] node) is the number of initial processing [IP] nodes (middle processing [MP] nodes) that are connected to the fusion node. The term complexity is justified by the observation that the more data that are fed to a data fusion [DF] node, the more complex the middle processing [MP] is. Similar considerations apply to the results fusion [RF] and final processing [FP].

Degree of complexity of the DF stage (or RF stage) is the maximum of the degrees of complexity of the individual DF (RF) nodes.

Degree of redundancy of an initial processing [IP] node (or middle processing [MP] node) is the number of data fusion [DF] nodes (result fusion [RF] nodes) that receive data (results) from the same initial processing IP (middle processing MP) node. The term redundancy is justified by the observation that the same information is communicated to more than one processing paths of the data flow structure.

Degree of redundancy of the DF stage (or RF stage) is the maximum of the degrees of redundancy of the individual IP (MP) nodes corresponding to the DF (RF) stage.

If the structure has both data fusion and results fusion stages, two degrees of complexity and two degrees of redundancy are required to characterize the structure.

Figures 5.8 and 5.9 depict two class 2 structures, with seven inputs each. In Figure 5.8 the degree of complexity $c$ is 2 and the degree of redundancy $r$ is 2, whereas in Figure 5.9 the degree of complexity $c$ is 3 and the degree of redundancy $r$ is 3. In both cases, all fusion nodes have the same degree of complexity and the same degree of redundancy. This need not be the case, in general.
Figure 5.8: Class 2 DFS with $r=2$, $c=2$
Figure 5.9: Class 2 DFS with $r=3$, $c=3$
Table 5.2: Transition Sets

<table>
<thead>
<tr>
<th>set</th>
<th>transition type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>initial processing</td>
</tr>
<tr>
<td>$T_2$</td>
<td>data fusion</td>
</tr>
<tr>
<td>$T_3$</td>
<td>middle processing</td>
</tr>
<tr>
<td>$T_4$</td>
<td>results fusion</td>
</tr>
<tr>
<td>$T_5$</td>
<td>final processing</td>
</tr>
</tbody>
</table>

5.6 Data Flow Structures Design

The first phase of the synthesis methodology be expressed as follows: Given $n$ inputs to the $C^2$ organization, design the data flow structures that are parameterized by the degrees of complexity and redundancy $c_1, r_1$ for the data fusion stage and $c_2, r_2$ for the results fusion stage.

The procedure for the generation of data flow structures produces the incidence matrix of the corresponding Petri Net, and consists of four steps:

1. selection of IP nodes that lead to DF
2. computation of the number of MP nodes
3. selection of the MP nodes that lead to RF
4. computation of the number of RF nodes

The incidence matrix of the data flow structure is composed of several blocks. The transitions form five sets defined in Table 5.2, and the places form seven sets defined in Table 5.3.

Thus the five transition sets and the seven place sets form 35 blocks in the incidence matrix. Each such block will be denoted as $P_iT_j$, corresponding to place set $P_i$ and transition set $T_j$.

Each IP transition has exactly one input place. Thus, the number of places in $P_1$ is $n$ and the number of transitions in $T_1$ is $n$. 

98
Let $n_1$ be the number of IP transitions which provide data for fusion (DF stage) and $n_2$ the number of IP transitions which provide results for fusion (RF stage). The number $n$ of IP transitions is:

$$n = n_1 + n_2$$  \hspace{1cm} (5.7)

The number $p$ of output places of the $n_1$ IP transitions is equal to the degree of redundancy of the DF stage, $r_1$, times $n_1$.

$$p = r_1 \times n_1$$  \hspace{1cm} (5.8)

The number $k$ of DF transitions is equal to the number of places $p$, divided by the degree of complexity of the DF stage, $c_1$.

$$k = \frac{p}{c_1} = n_1 \frac{r_1}{c_1}$$  \hspace{1cm} (5.9)

where $k$ must be integer. Thus, the number of places in $P_2$ is $p$ and the number of transitions in $T_2$ is $k$. If $k$ is not integer, then the pair $(r_1, c_1)$ is not feasible, i.e., it is not possible for all DF transitions to have the same degree of complexity and for all IP transitions to have the same degree of redundancy.
Each DF transition has one output place and each MP transition has exactly one input place. Consequently, the number of places in $P_3$ is $k$ and the number of transitions in $T_3$ is $k$.

Let $k_1$ be the number of MP transitions which produce outputs of the data flow structure, and $k_2$ be the number of MP transitions which produce results for fusion. Then

$$k = k_1 + k_2 \quad (5.10)$$

Let $r_2$ and $c_2$ be the redundancy and complexity of the RF stage. The number $q$ of input places to RF transitions is equal to:

$$q = (n_2 + k_2)r_2 \quad (5.11)$$

The number $m$ of RF transitions is equal to:

$$m = \frac{q}{c_2} = \frac{(n_2 + k_2)r_2}{c_2} \quad (5.12)$$

where $m$ must be integer. If $m$ is not integer, then the pair $(r_2, c_2)$, is not feasible, i.e., it is not possible for all RF transitions to have the same degree of complexity and for all MP transitions to have the same degree of redundancy. Thus, the number of places in $P_4$ is $q$ and the number of transitions in $T_4$ is $m$.

Each RF transition has one output place and each FP transition has exactly one input place. Hence, the number of places in $P_5$ is $m$ and the number of transitions in $T_5$ is $m$.

Each of the $k_1$ MP transitions that produce outputs has exactly one output place, and finally, each FP transition has exactly one output place. Thus, the number of places in $P_6$ is $k_1$ and the number of transitions in $P_7$ is $m$. The block form of the incidence matrix is depicted in Figure 5.10.

The assignment of values to the elements of each block is described next.

Block $P_1T_1$ is of dimension $n \times n$, and has diagonal elements equal to -1, and non-diagonal elements equal to 0. The blocks $P_1T_2$, $P_1T_3$, $P_1T_4$ and $P_1T_5$ have all elements equal to zero.
\[
\begin{align*}
\begin{bmatrix}
P_1 T_1 \\
n \times n
\end{bmatrix} & \begin{bmatrix}
P_1 T_2 \\
n \times k
\end{bmatrix} & \begin{bmatrix}
P_1 T_3 \\
n \times k
\end{bmatrix} & \begin{bmatrix}
P_1 T_4 \\
n \times m
\end{bmatrix} & \begin{bmatrix}
P_1 T_5 \\
n \times m
\end{bmatrix} \\
\begin{bmatrix}
P_2 T_1 \\
p \times n
\end{bmatrix} & \begin{bmatrix}
P_2 T_2 \\
p \times k
\end{bmatrix} & \begin{bmatrix}
P_2 T_3 \\
p \times k
\end{bmatrix} & \begin{bmatrix}
P_2 T_4 \\
p \times m
\end{bmatrix} & \begin{bmatrix}
P_2 T_5 \\
p \times m
\end{bmatrix} \\
\begin{bmatrix}
P_3 T_1 \\
k \times n
\end{bmatrix} & \begin{bmatrix}
P_3 T_2 \\
k \times k
\end{bmatrix} & \begin{bmatrix}
P_3 T_3 \\
k \times k
\end{bmatrix} & \begin{bmatrix}
P_3 T_4 \\
k \times m
\end{bmatrix} & \begin{bmatrix}
P_3 T_5 \\
k \times m
\end{bmatrix} \\
\begin{bmatrix}
P_4 T_1 \\
q \times n
\end{bmatrix} & \begin{bmatrix}
P_4 T_2 \\
q \times k
\end{bmatrix} & \begin{bmatrix}
P_4 T_3 \\
q \times k
\end{bmatrix} & \begin{bmatrix}
P_4 T_4 \\
q \times m
\end{bmatrix} & \begin{bmatrix}
P_4 T_5 \\
q \times m
\end{bmatrix} \\
\begin{bmatrix}
P_5 T_1 \\
m \times n
\end{bmatrix} & \begin{bmatrix}
P_5 T_2 \\
m \times k
\end{bmatrix} & \begin{bmatrix}
P_5 T_3 \\
m \times k
\end{bmatrix} & \begin{bmatrix}
P_5 T_4 \\
m \times m
\end{bmatrix} & \begin{bmatrix}
P_5 T_5 \\
m \times m
\end{bmatrix} \\
\begin{bmatrix}
P_6 T_1 \\
k_1 \times n
\end{bmatrix} & \begin{bmatrix}
P_6 T_2 \\
k_1 \times k
\end{bmatrix} & \begin{bmatrix}
P_6 T_3 \\
k_1 \times k
\end{bmatrix} & \begin{bmatrix}
P_6 T_4 \\
k_1 \times m
\end{bmatrix} & \begin{bmatrix}
P_6 T_5 \\
k_1 \times m
\end{bmatrix} \\
\begin{bmatrix}
P_7 T_1 \\
m \times n
\end{bmatrix} & \begin{bmatrix}
P_7 T_2 \\
m \times k
\end{bmatrix} & \begin{bmatrix}
P_7 T_3 \\
m \times k
\end{bmatrix} & \begin{bmatrix}
P_7 T_4 \\
m \times m
\end{bmatrix} & \begin{bmatrix}
P_7 T_5 \\
m \times m
\end{bmatrix}
\end{align*}
\]

Figure 5.10: Block form of Incidence Matrix
Block $P_2 T_1$ has dimension $p \times n$. Its elements $B_{ij}$ obtain their values according to:

$$B_{ij} = \begin{cases} 
1 & \text{for } j = 1, 2, \ldots, n_1, \\
0 & \text{otherwise}
\end{cases} 
$$  

(5.13)

Block $P_2 T_2$ has dimension $p \times k$. Its elements $B_{ij}$ obtain their values according to:

$$B_{ij} = \begin{cases} 
-1 & \text{if place } i \text{ is connected to transition } j \\
0 & \text{otherwise}
\end{cases}  
$$  

(5.14)

The blocks $P_2 T_3$, $P_2 T_4$, and $P_2 T_5$ have all elements equal to zero.

Block $P_3 T_2$ has dimension $k \times k$. The diagonal elements $B_{ij}$ are equal to 1 and the non-diagonal elements are equal to 0.

The dimension of block $P_3 T_3$ is $k \times k$. The diagonal elements $B_{ij}$ are equal to -1 and the non-diagonal elements are equal to 0. The blocks $P_3 T_1$, $P_3 T_4$, and $P_3 T_5$ have all elements equal to zero.

The dimension of block $P_4 T_1$ is $q \times n$, while the dimension of block $P_4 T_3$ is $q \times k$, and the dimension of block $P_4 T_4$ is $q \times m$.

The elements of block $P_4 T_1$ obtain values according to:

$$B_{ij} = 0$$  

(5.15)

for all $i$ and all $j$, if $n_2 = 0$. If $n_2 \neq 0$ then

$$B_{ij} = \begin{cases} 
1 & \text{for } j = n_1 + 1, n_1 + 2, \ldots, n \\
0 & \text{otherwise}
\end{cases}  
$$  

(5.16)

The elements of block $P_4 T_3$ obtain values according to:
\[ B_{ij} = \begin{cases} 1 & \text{for } j = k_1 + 1, \ldots, k, \\
 & i = [n_2 + (j - k_1) - 1] r_1 + l, l = 1, 2, \ldots, r_1 \\
0 & \text{otherwise} \end{cases} \] (5.17)

The elements of block \( P_4 T_4 \) obtain values according to:

\[ B_{ij} = \begin{cases} -1 & \text{if place } i \text{ is connected to transition } j \\
0 & \text{otherwise} \end{cases} \] (5.18)

The blocks \( P_4 T_3 \) and \( P_4 T_5 \) have all elements equal to zero.

Block \( P_6 T_4 \) has dimension \( m \times m \). The diagonal elements \( B_{ij} \) are equal to 1 and the non-diagonal elements are equal to 0. The dimension of block \( P_6 T_5 \), is \( m \times m \). Its diagonal elements \( B_{ij} \) are equal to -1 and the non-diagonal elements are equal to 0. The blocks \( P_6 T_1, P_6 T_2, \) and \( P_6 T_3 \) have all elements equal to zero.

The dimension of block \( P_6 T_3 \) is \( k_1 \times k \). The elements \( B_{ij} \) with \( i = j \), are equal to 1 and the other elements are equal to 0. The blocks \( P_6 T_1, P_6 T_2, P_6 T_4, \) and \( P_6 T_5 \) have all elements equal to zero.

The dimension of block \( P_7 T_5 \) is \( m \times m \). The diagonal elements \( B_{ij} \) are equal to 1 and the non-diagonal elements are equal to 0. The blocks \( P_7 T_1, P_7 T_2, P_7 T_3, \) and \( P_7 T_4 \) have all elements equal to zero.

5.7 Design Parameters

The design parameters are: \( n_1, k_1, r_1, r_2, c_1 \) and \( c_2 \). The number of IP transitions that lead to data fusion is selected by \( n_1 \). The number of MP transitions which produce outputs of the organization is selected by \( k_1 \). The redundancy of the DF and RF stages is selected by \( r_1 \) and \( r_2 \). The complexity of the DF and RF stages is selected by \( c_1 \) and \( c_2 \).

Note that it has been assumed that all nodes in the DF stage have the same degree of complexity and redundancy. The same applies for the RF stage. Data structures
with degrees of complexity and redundancy of individual transitions less than those of the data and results fusion stages, i.e.

$$r < r_1, \ r < r_2$$  \ (5.19)

and

$$c < c_1, \ c < c_2$$  \ (5.20)

can be obtained by removing links from the structure obtained under this assumption.

In order to generate candidate data flow structures from each class, the range of the degree of complexity and the range of the degree of redundancy for the DF and RF stages must be specified. These are selected by considering the adaptability of the processing schema represented by the data flow structure to the data processing functions required by the task. The degree of redundancy should be less or equal to the number of available assets, while the degree of complexity should be less or equal to the number of transitions that have data or results available for fusion.

Once these ranges \((c_{11}, c_{12}), (r_{11}, r_{12}), (c_{21}, c_{22})\) and \((r_{21}, r_{22})\), have been selected, all structures with

$$r_{11} \leq r_1 \leq r_{12}$$
$$c_{11} \leq c_1 \leq c_{12}$$
$$r_{21} \leq r_2 \leq r_{22}$$
$$c_{21} \leq c_2 \leq c_{22}$$  \ (5.21)

are generated.

5.8 Examples

Three examples will be used to illustrate the generation of data flow structures. First, a class 1 data flow structure with three inputs is constructed. Since there are 3 inputs, \(n = 3\), and for a class 1 structure, \(n_2 = 0\). Thus \(n_1 = 3\). For degree of redundancy \(r_1 = 5\) and degree of complexity \(c_1 = 3\), the number of output places, \(p\), of
the IP transitions is
\[ p = n_1 r_1 = 3 \times 5 = 15 \]  \hspace{1cm} (5.22)

and the number \( k \) of DF transitions is
\[ k = \frac{p}{c_1} = \frac{15}{3} = 5 \]  \hspace{1cm} (5.23)

For a class 1 structure, \( k_1 = 0 \). Hence, \( k_2 = k - k_1 = 5 \). For degree of redundancy \( r_2 = 3 \) and degree of complexity \( c_2 = 5 \), the number of output places, \( q \), of the MP transitions is
\[ q = (n_2 + k_2) r_2 = 5 \times 3 = 15 \]  \hspace{1cm} (5.24)

and the number \( m \) of RF transitions is
\[ m = \frac{q}{c_2} = \frac{15}{5} = 3 \]  \hspace{1cm} (5.25)

The incidence matrix is shown in Figure 5.11 and the corresponding Petri Net in Figure 5.12.

The second example is a class 13 data flow structure with three inputs. Since there are three inputs, \( n = 3 \). Let \( n_2 = 1 \). Thus \( n_1 = 2 \). For degree of redundancy \( r_1 = 2 \) and degree of complexity \( c_1 = 2 \):
\[ p = n_1 r_1 = 2 \times 2 = 4 \]
\[ k = \frac{p}{c_1} = \frac{4}{2} = 2 \]

For a class 2 structure, \( k_1 = 0 \). Hence for degree of redundancy \( r_2 = 5 \) and degree of complexity \( c_2 = 3 \):
\[ k_2 = k - k_1 = 2 - 0 = 2 \]
\[ q = (n_2 + k_2) r_2 = 3 \times 5 = 15 \]
\[ m = \frac{q}{c_2} = \frac{15}{3} = 5 \]

The incidence matrix is shown in Figure 5.13 and the corresponding Petri Net in Figure 5.14.
Figure 5.11: Class 1 DFS: Incidence Matrix
Figure 5.12: Class 1 DFS: Petri Net
Figure 5.13: Class 13 DFS: Incidence Matrix
Figure 5.14: Class 13 DFS: Petri Net
Finally, a class 123 data flow structure with three inputs is constructed. Since there are three inputs, \( n = 3 \). Let \( n_2 = 1 \). Thus \( n_1 = 2 \). For degree of redundancy \( r_1 = 3 \) and degree of complexity \( c_1 = 2 \):

\[
\begin{align*}
p &= n_1 r_1 = 2 \times 3 = 6 \\
k &= \frac{p}{c_1} = \frac{6}{2} = 3
\end{align*}
\]

Let \( k_1 = 2 \). Hence, for degree of redundancy \( r_2 = 3 \) and degree of complexity \( c_2 = 2 \):

\[
\begin{align*}
k_2 &= k - k_1 = 3 - 2 = 1 \\
q &= (n_2 + k_2)r_2 = 2 \times 3 = 6 \\
m &= \frac{q}{c_2} = \frac{6}{2} = 3
\end{align*}
\]

The incidence matrix is shown in Figure 5.15 and the corresponding Petri Net in Figure 5.16.

### 5.9 Evaluation of Data Flow Structures

In order to evaluate the generated data flow structures, we need to determine the adaptability of each data flow structure to the information processing task that it is designed to implement. Thus, we need to develop the processing algorithms of the functions in general form, and then associate functions to the transitions of each data flow structure.

In general, the functions and the corresponding algorithms may be classified as local and global with respect to the geographical area whose data they process, and as situation assessment, courses of action development, and response selection, with respect to the process that they implement.

In this context, initial processing transitions correspond to local situation assessment, while data fusion transitions correspond to global situation assessment. Middle processing transitions develop local courses of action, result fusion transitions develop global courses of action and final processing transitions implement response selection.
Figure 5.15: Class 123 DFS: Incidence Matrix
Figure 5.16: Class 123 DFS: Petri Net
If middle processing or final processing transitions are not present in some information flow paths of the data flow structure, then the corresponding functions are either missing, or they are implemented at the appropriate transitions of the other information flow paths.

Obviously, some modification of the algorithms may be required, in order to adapt them to the processing schema. If the algorithms cannot be adapted to the processing schema imposed by a data flow structure, then that data flow structure is rejected. During the modification, some communication links may be removed if the corresponding communications are not implemented. Similarly, some transitions may not perform the corresponding processing, i.e. they correspond to null processes, and may be removed.

5.10 Organization Architecture Design

The candidate data flow structures obtained by the procedure described in the preceding sections are augmented and transformed into decision-making organizations. During this phase, functions are allocated to the decisionmakers.

Functions allocated to a decisionmaker must observe three requirements:

1. Must be related through an input-output relationship, i.e., the output of one function must be the input to the next function performed by the decisionmaker so that each decisionmaker processes information relevant to the same subtask;

2. Must belong to different slices on the Petri Net so that they observe concurrency; and

3. Must conform to the specialization of the respective decisionmaker.

Requirements 1 and 2 are satisfied by functions that are on the same information flow path; thus only functions that belong to the same information flow path are considered for allocation to any decisionmaker. When such a set of functions is allocated to a decisionmaker, a resource place is introduced that is connected so that it is an
Figure 5.17: Synchronous Protocol

output place of the last and an input place to the first transition allocated to the
decisionmaker.

Then the DMO is transformed into a $C^2O$ by introducing the supporting $C^3$ system,
i.e., software and hardware such as decision aids, databases, and communication links.

Transitions are introduced at the appropriate locations on the Petri Net to depict
the communication links. It should be noted that the protocols for information exchange among
decisionmakers were selected by the data flow structure. The only choice left is that of synchronous vs asynchronous communication protocols among
the decisionmakers (not of the communication hardware). In the case of voice links,
the protocol is synchronous (Figure 5.17), whereas for digital links, the protocol is
asynchronous (Figure 5.18).

Transitions are also introduced to depict the use of decision aids and databases.
In general the decisionmaker may or may not access a database or use a decision aid.
Therefore switches are introduced to depict the choices available.
5.11 Modification of Candidate Designs and Selection

Next, the computation of the measures of performance of the candidate decision-making organization designs is performed. Specifically the Accuracy $J$, measure of Timeliness $T$, and measure of processing capacity $S$ are computed. Then the Measure of Effectiveness $Q$ of each design is evaluated. If the MOE is not satisfactory, iterations are performed to modify the design so that the MOE value is increased. The modifications may include alternate function allocation, introduction of multiple processing channels, modification or introduction of decision aids and databases and improvement of the communication links.

The procedure for the modification depends on the location of the organization locus with respect to the requirements locus. The existing cases are described in Table 5.4.

Case 1: In this case the accuracy must be improved, i.e. the discrepancy of actual and desired response be reduced. One way of attaining this, is the introduction of decision aids. Decision aids may implement more exact algorithms, resulting in more accurate situation assessment, for the same quality (accuracy) of input data, and better
### Table 5.4: Design Modification Cases

<table>
<thead>
<tr>
<th>case</th>
<th>$J &lt; J_o$</th>
<th>$T &gt; T_o$</th>
<th>$S &gt; S_o$</th>
<th>must improve</th>
<th>modification required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>false</td>
<td>true</td>
<td></td>
<td>accuracy</td>
<td>introduce decision aid</td>
</tr>
<tr>
<td>2</td>
<td>true</td>
<td>false</td>
<td></td>
<td>response time</td>
<td>better communications</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>improve database access</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>improve decision-aids</td>
</tr>
<tr>
<td>3</td>
<td>false</td>
<td>false</td>
<td></td>
<td>accuracy and</td>
<td>introduce decision aid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>response time</td>
<td>better communications</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>improve database access</td>
</tr>
<tr>
<td>4</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>throughput rate</td>
<td>modify function allocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>more processing channels</td>
</tr>
</tbody>
</table>

Sources of action development.

**Case 2:** In this case the response time must be improved. There exist two areas for improvement: Introduction of better communication hardware, and modification of existing decision-aids and databases to reduce processing and access time.

**Case 3:** In this case both the accuracy and the response time must be improved. First improve the accuracy by introducing decision aids. A secondary effect may be an improvement of response time. If response time needs further improvement, introduce better communication hardware, and modify databases to reduce access time.

**Case 4:** In this case the throughput rate must be increased. Two areas for improvement exist. (1) Identify which set of functions (directed elementary circuit) has the minimum processing rate that slows the organization throughput rate. Re-allocate the functions of that circuit to more decision-makers so that the throughput rate increases. If the minimum processing rate of the circuits is low due to large delays of decision aid or database access, improve the response time of the hardware. (2) Introduce more processing channels, i.e., copies of the simple information flow path that contains the set of functions with the lowest processing rate, and allocate the arriving data to these channels alternatively. For example, if a second processing channel is introduced, the
processing rate of the corresponding directed elementary circuit will be doubled.

After the modifications are completed for all candidate designs, compute the MOE of each design and select the design having the highest MOE value.

5.12 Chapter Summary

In this Chapter, the issues pertinent to the synthesis of organizations have been reviewed, and the synthesis problem has been formulated. A classification of data flow structures suitable for decision-making has been introduced, and a procedure for the generation of data flow structures has been developed. It should be noted that the considered data flow structures have been limited to only two fusion stages. The extension to three fusion stages is straightforward, both in the classification and in the procedure for their generation. The procedure for the transformation of the data flow structures into decision-making organizations, and finally into Command and Control organizations has been described. Finally, the modification of the developed designs, to improve their Measure of Effectiveness has been examined.
Chapter 6

APPLICATION OF THE SYNTHESIS METHODOLOGY

6.1 Introduction

In this chapter, a naval air defense example is employed to illustrate the synthesis methodology. Specifically, the objective is to design a Command and Control organization for the outer air battle.

6.2 Problem Description

The objective of air defense for a naval battle-group is to deploy aircraft and set up a screen to locate incoming enemy aircraft, and engage them before they reach a distance from which they can fire their missiles and endanger the carrier, or its escorts. Assume that the enemy missiles have a range $R_m$. The circumference of the circle of radius $R_m$ with center the battle-group, is the weapons release line. The mission is to keep the enemy aircraft as far from the weapons release line as possible. The engagement at distance greater than the weapons release line is called the outer air battle. If enemy aircraft penetrate the screen, they are not followed by the interceptors participating in the outer battle, but are engaged by aircraft under the carrier command, surface to air missiles or, finally, point defense systems. The latter engagement is called inner air battle.

The assumptions and abstractions used in this chapter to model the decision-making functions pertinent to the outer air battle may not correspond to actual naval air tactics, and do not necessarily reflect real naval air operations.

It is assumed that an aircraft carrier has two squadrons of interceptor aircraft (F14
Tomcat or F4 Phantom), two squadrons of light attack aircraft (A7 Corsair), one squadron of all weather attack aircraft (A6 Intruder), and four airborne warning radar aircraft (E2C Hawkeye).

Two E2C aircraft patrol two sectors and are positioned at a distance $R_p$ from the carrier. Each E2C commands one squadron of interceptors, which are directed to engage the threats in the corresponding sector. The two squadrons of A7 may be deployed in either of the two sectors or in the inner air battle, whereas the squadron of A6 is used along with tanker aircraft to refuel the fighters before the beginning of the engagement.

The E2C is equipped with passive and active radar. Active radar receives the reflection of its beam by objects (such as other aircraft) while passive radar receives the radar transmission of other aircraft. The active radar has a range $R_a$, whereas the passive radar has a range $R_o$, assumed to be twice the range $R_a$. Assume that enemy active radar has also the same range $R_a$.

Each E2C initially operates only the passive radar so as not to reveal its position to enemy units. When enemy aircraft approach the E2C at a range $R_a$ (called the burn-through range) and therefore are about to acquire the E2C on their radar, the E2C turns its active radar on.

Enemy aircraft may be identified on the basis of their emmitter signature. Emmitter signatures may be classified according to the signal characteristics. An emmitter signature connotes the existence of aircraft equipped with the corresponding emmitter. Correlation of transmitter signature and aircraft velocity may be used to classify the aircraft type with a level of certainty (likelihood).

Based on the assessment of incoming threats in each sector, the decisionmakers develop courses of action (plans about their response), and finally select a response from the available alternatives. In this context, when contacts are initially made on passive radar, the squadrons of F14 and A7 are re-fueled and a third E2C may be launched to patrol in another sector. The squadrons of A7 are assigned to the sectors where the incoming raid is stronger, or remain in the inner battle region.

In this example, the vectoring of aircraft to threats is not modeled. Similarly, the
details of inner air battle are not included.

6.3 Outline of Synthesis

The synthesis of the candidate $C_2$ organizations for the outer air battle involves the following steps:

1. Develop suitable data flow structures
2. Develop a decision-making model for the outer air battle
3. Identify the decision-making functions and the corresponding algorithms
4. Evaluate the data flow structures
5. Develop $C^2$ organizations from each structure.
6. Model the input tasks and define the cost function $C(Y_i,Y_d)$
7. Develop software for the computation of workload
8. Compute measures of performance (accuracy, response time, processing rate)
9. Compute the measure of effectiveness for each candidate organization
10. Compare the organizations on the basis of the MOE value.

6.4 Generation of Data Flow Structures

Let us start with the design of class 2 DFS. The number of inputs is 2 (two E2C aircraft are airborne); consequently the number of IP transitions is 2 ($n_1 = 2$). The degree of complexity of the DF stage must be less or equal to the number of inputs $c_1 \leq 2$. The number of assets is 3; hence the degree of redundancy of the DF stage must be $r_1 \leq 3$. The number $k$ of DF transitions is

$$k = n_1 \frac{r_1}{c_1} \quad (6.1)$$
Table 6.1: Computation of the number of DF transitions, class 2 DFS

<table>
<thead>
<tr>
<th>$r_1$</th>
<th>$c_1$</th>
<th>$k$</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>not enough assets</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>6</td>
<td>not enough assets</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>shown in Figure 6.1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>shown in Figure 6.2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>shown in Figure 6.3</td>
</tr>
</tbody>
</table>

The combinations of $r_1$ and $c_1$, and the number $k$ of DF transitions are given in Table 6.1. The corresponding data flow structures are shown in Figures 6.1 to 6.3.

Note that the class 3 data flow structures have the same Petri Net representation as class 2 structures; they differ only in the nature of the processing performed at the individual transitions.

Now let us develop class 1 data flow structures. The design of the DF stage is similar to that of class 2 structures. From Table 6.1 we observe that we may have 1, 2, or 3 MP transitions. The number of assets is 3 (3 command centers: two airborne radar aircraft and one carrier); hence the degree of redundancy must be $r_2 \leq 3$. The degree of complexity must be less or equal to the number of MP transitions. The number $m$ of RF transitions is:

$$m = k \frac{r_2}{c_2}$$  \hspace{1cm} (6.2)

The combinations of $k$, $r_2$, $c_2$, and the computed number $m$ of RF transitions, are given in Table 6.2, while the corresponding data flow structures are shown in Figures 6.4 to 6.12.

Now let us consider class 12 data flow structures. There are 2 or 3 MP transitions; some of the MP transitions lead to results fusion and some produce outputs. In the case of 2 transitions, both are inputs for fusion, and hence a class 12 structure is not feasible. In the case of 3 MP transitions, 2 may lead to fusion and 1 may produce an output. The degree of complexity must be less than or equal to the number of transitions that lead to RF; thus $c_2 \leq 2$. The degree of redundancy must be less than
Table 6.2: Computation of the number of RF transitions, class 1 DFS

<table>
<thead>
<tr>
<th>$k$</th>
<th>$r_2$</th>
<th>$c_2$</th>
<th>$m$</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>shown in Figure 6.4</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>shown in Figure 6.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>not enough assets</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>not enough assets</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>shown in Figure 6.6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>shown in Figure 6.7</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>shown in Figure 6.8</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>not enough assets</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>not enough assets</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1.5</td>
<td>not integer</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>shown in Figure 6.9</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4.5</td>
<td>not integer</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>shown in Figure 6.10</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>shown in Figure 6.11</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>shown in Figure 6.12</td>
</tr>
</tbody>
</table>

or equal to the number of assets available for fusion, in this case two; hence $r_2 \leq 2$. The combinations of $r_2$, $c_2$ and $m$ are given in Table 6.3, and the data flow structures are depicted in Figures 6.13 and 6.14.

Note that class 13 and class 123 Data flow structures cannot be constructed for this example, because there are only two IP transitions; if they lead to data fusion (flow type 1), then there are no other IP transitions for results fusion (flow type 3).

Table 6.3: Computation of the number of RF transitions, class 12 DFS

<table>
<thead>
<tr>
<th>$r_2$</th>
<th>$c_2$</th>
<th>$m$</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>not enough assets</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>shown in Figure 6.13</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>shown in Figure 6.14</td>
</tr>
</tbody>
</table>

122
Figure 6.1: DFS # 1

Figure 6.2: DFS # 2
Figure 6.3: DFS # 3

Figure 6.4: DFS # 4
Figure 6.5: DFS # 5

Figure 6.6: DFS # 6
Figure 6.7: DFS # 7

Figure 6.8: DFS # 8
Figure 6.9: DFS # 9

Figure 6.10: DFS # 10
Figure 6.11: DFS # 11

Figure 6.12: DFS # 12

128
Figure 6.13: DFS # 13

Figure 6.14: DFS # 14
6.5 Function and Algorithm Description

The model of the decision-making pertinent to the outer air battle, while simple, preserves the following important features:

1. Threat identification (classification) on the basis of emitter signature and threat velocity

2. Development of courses of action considering the strength of the incoming raid and the type of enemy aircraft. Due to limited information about naval air tactics, the courses of action are only identified by labels without specification of their particulars.

3. Response selection from the available courses of action based on the availability of resources.

The following functions are defined: local and global situation assessment, local courses of action development, and local and global response selection.

Note that the functions and algorithms are not decisionmaker oriented but content (objective) oriented. They are not related to any data flow structure and they can be modified to adapt to each candidate data flow structure.

Local (sector) situation assessment

The inputs to the algorithm are the range and number of threats, their signature and their speed. Based on the number of threats, and the range, the raid strength is estimated as unknown, weak, strong or very strong. Assume that there exist two distinct emitter signatures. Based on the combination of emitter signature and threat speed, the detected threats are classified as aircraft type a, b, c or unknown. The flowchart is shown in Figure 6.15.

Global situation assessment

The objective of this function is to determine where the most powerful raid is about to develop. The inputs are the strength of the raid and the type of aircraft in each
Figure 6.15: Local Situation Assessment Algorithm
sector. The output may be

1. strong raid in both sectors
2. weak raid in both sectors
3. strong raid of aircraft type c in sector 1
4. strong raid of aircraft type c in sector 2
5. strong raid of aircraft type a, b, or unknown in sector 1
6. strong raid of aircraft type a, b, or unknown in sector 2

The flowchart is shown in Figures 6.16 through 6.18.

Local (sector) courses of action development

The inputs to this function are the raid strength and type of aircraft in the corresponding sector. The output is the courses of action that are appropriate for each case. The courses of action are

1. $\phi_1, \phi_2$ in the case of strong or very strong raid of type a or b
2. $\omega_1, \omega_2, \omega_3$ in the case of strong or very strong raid of type c or unknown
3. $\psi_1, \psi_2, \psi_3$ in the case of weak raid

In some situations only one course of action is developed, while in others two courses of action are developed. In the second case the reasoning is that one of the two courses of action is appropriate, depending on the number of assets (interceptors) in the sector. The final selection of one of the two courses of action, will be done by the response selection algorithm. Two algorithms for local courses of action development have been formulated. The flowcharts are shown in Figures 6.19 and 6.20.
Figure 6.16: Global Situation Assessment Algorithm, part 1
Figure 6.17: Global Situation Assessment Algorithm, part 2
Figure 6.18: Global Situation Assessment Algorithm, part 3
Global response selection

The input to this function is the global situation assessment. The objective is to allocate the free resources (the two squadrons of A7 Corsair) to the sector where the strongest or potentially more damaging raid occurs, or in the case of equally strong raid in both sectors, to allocate one squadron to each sector. The output is how many of the free resources will be allocated to each sector. The flowchart of the algorithm is shown in Figure 6.21.

Local (sector) response selection

The inputs to this function are the local courses of action and the global response selection, i.e., the decision to allocate more assets to the sector or not. The output is one of the developed courses of action. The flowchart is shown in Figure 6.22.

6.6 Data Flow Structure Evaluation and Selection

In the DFS evaluation stage, the decision-making algorithms are adapted to all data flow structures. During this stage, some links and some transitions may be removed. If the adaptation of the algorithms is not feasible for some structures, those structures are rejected. The computation of measures of performance requires substantial code development for each organization. In this example, we will apply the second and third phase of the synthesis methodology to only two data flow structures, in order to keep the software development effort within reasonable bounds.

Specifically, the data flow structures of Figures 6.7 and 6.11 have been selected. These structures have been modified as shown in Figures 6.23 and 6.24 respectively.

The operating scenario for each structure is as follows: In Figure 6.23 there exist two processing channels, corresponding to the decision-making of the two E2C aircraft. Each E2C commands two squadrons of interceptors. The E2C personnel performs local and global situation assessment, global response selection, local courses of action development, and local response selection. The global response selection may allocate
Figure 6.19: Local Courses of Action Development; Algorithm 1
z1: raid strength
z2: aircraft type

Yes
z1=null?

No
w=null

No
z1=unknown?

Yes
z1=very strong?

No
w=null

Yes
z1=strong?

No
w=ψ_2

Yes
z2=c?

No
w=ω_1/ω_2

Yes
w=φ_1

w=ψ_1

No
w=ψ_1

Figure 6.20: Local Courses of Action Development; Algorithm 2
Figure 6.21: Global Response Selection Algorithm
Figure 6.22: Local Response selection Algorithm
Figure 6.23: Data Flow Structure A

Figure 6.24: Data Flow Structure B
one of the two squadrons to the sector patrolled by the other E2C.

In Figure 6.24 there exist three processing channels, the top and bottom corresponding to each E2C, and the middle to the carrier. Each E2C commands one squadron of interceptors, while the carrier commands two squadrons. The E2C personnel performs local situation assessment, local courses of action development, and local response selection. The carrier personnel performs global situation assessment and global response selection. The global response selection may allocate one or two squadrons to the sectors depending on the strength of the raid in each sector.

6.7 Organization Design

Two organizations have been developed from each data flow structure, through different function allocation to decisionmakers.

From DFS A, a 2 person and a 4 person organization have been developed. The corresponding Petri Nets are shown in Figures 6.25 and 6.26. In organization 1, one decisionmaker processes the information on each E2C aircraft, while in organization 2 the processing has been allocated to two decisionmakers. The first performs local and global situation assessment, and the second develops courses of action and selects a response.

From DFS B, a 3 person and a 5 person organization have been developed. The corresponding Petri Nets are shown in Figures 6.27 and 6.28. The decisionmaker on the carrier performs global situation assessment and global response selection (resource allocation). In organization 3 the information processing on each E2C aircraft is performed by one decisionmaker, while in organization 4 by two decisionmakers. The one performs local situation assessment, and the second develops courses of action and selects the response.

In Figures 6.29, 6.30, 6.31 and 6.32, the transitions representing the synchronization and the communication processes have been introduced. In this example, we will not incorporate decision aids or databases.
Figure 6.25: First Function Allocation for DFS A

Figure 6.26: Second Function Allocation for DFS A
Figure 6.27: First Function Allocation for DFS B

Figure 6.28: Second Function Allocation for DFS B
Figure 6.29: Organization 1

Figure 6.30: Organization 2

145
Figure 6.31: Organization 3

Figure 6.32: Organization 4
### Table 6.4: Input alphabet

<table>
<thead>
<tr>
<th>range</th>
<th>number</th>
<th>signature</th>
<th>speed</th>
<th>desired response</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>$n$</td>
<td>$S$</td>
<td>$v$</td>
<td></td>
</tr>
<tr>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>$r \geq R_a$</td>
<td>$n &lt; n_o$</td>
<td>$S_1$</td>
<td>$v &lt; v_o$</td>
<td>$\psi_2$</td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n &lt; n_o$</td>
<td>$S_1$</td>
<td>$v_o &lt; v &lt; v^*$</td>
<td>$\psi_3$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n &lt; n_o$</td>
<td>$S_1$</td>
<td>$v^* &lt; v$</td>
<td>$\psi_1$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n &lt; n_o$</td>
<td>$S_2$</td>
<td>$v &lt; v_o$</td>
<td>$\psi_2$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n &lt; n_o$</td>
<td>$S_2$</td>
<td>$v_o &lt; v &lt; v^*$</td>
<td>$\psi_3$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n &lt; n_o$</td>
<td>$S_2$</td>
<td>$v^* &lt; v$</td>
<td>$\psi_1$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n_o &lt; n &lt; n^*$</td>
<td>$S_1$</td>
<td>$v &lt; v_o$</td>
<td>$\omega_1$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n_o &lt; n &lt; n^*$</td>
<td>$S_1$</td>
<td>$v_o &lt; v &lt; v^*$</td>
<td>$\omega_3$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n_o &lt; n &lt; n^*$</td>
<td>$S_2$</td>
<td>$v^* &lt; v$</td>
<td>$\phi_1$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n_o &lt; n &lt; n^*$</td>
<td>$S_2$</td>
<td>$v &lt; v_o$</td>
<td>$\omega_1$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n_o &lt; n &lt; n^*$</td>
<td>$S_2$</td>
<td>$v_o &lt; v &lt; v^*$</td>
<td>$\omega_3$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n_o &lt; n &lt; n^*$</td>
<td>$S_2$</td>
<td>$v^* &lt; v$</td>
<td>$\phi_1$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n^* &lt; n$</td>
<td>$S_1$</td>
<td>$v &lt; v_o$</td>
<td>$\omega_2$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n^* &lt; n$</td>
<td>$S_1$</td>
<td>$v_o &lt; v &lt; v^*$</td>
<td>$\omega_2$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n^* &lt; n$</td>
<td>$S_2$</td>
<td>$v^* &lt; v$</td>
<td>$\phi_2$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n^* &lt; n$</td>
<td>$S_2$</td>
<td>$v &lt; v_o$</td>
<td>$\omega_2$</td>
<td></td>
</tr>
<tr>
<td>$r &lt; R_a$ &amp; $n^* &lt; n$</td>
<td>$S_2$</td>
<td>$v_o &lt; v &lt; v^*$</td>
<td>$\omega_2$</td>
<td></td>
</tr>
</tbody>
</table>

### 6.8 Model of Input Tasks and Cost Function Selection

The input alphabet for each sector is shown in Table 6.1. The alphabet consists of twenty symbols, all assumed equally likely, with probability of occurrence $p = 0.05$. Combination of the inputs for the two sectors, produces the input alphabet of the organization, which consists of 400 symbols, each having probability of occurrence $p = 0.0025$. The cost function $C(Y_t, Y_{di})$ is shown in Table 6.2.
Table 6.5: Cost Function $C(Y_i, Y_{di})$

<table>
<thead>
<tr>
<th>actual</th>
<th>desired response</th>
<th>( \phi_1 )</th>
<th>( \phi_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega_1 )</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( \omega_2 )</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>( \omega_3 )</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>actual</th>
<th>desired response</th>
<th>( \psi_1 )</th>
<th>( \psi_2 )</th>
<th>( \psi_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \psi_1 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( \psi_2 )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( \psi_3 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>actual</th>
<th>desired response</th>
<th>( \phi_1 )</th>
<th>( \phi_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_1 )</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>( \phi_2 )</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>actual</th>
<th>desired response</th>
<th>( \phi_1 )</th>
<th>( \phi_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega_1 )</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( \omega_2 )</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>( \omega_3 )</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>actual</th>
<th>desired response</th>
<th>( \phi_1 )</th>
<th>( \phi_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_1 )</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>( \phi_2 )</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
6.9 Decision Strategies

The course of action development function may by implemented by two algorithms. We have assumed that the probability of algorithm selection is not conditioned on the input to the function. Thus, there exist two pure strategies:

\[ P(u = 1) = 1, \quad P(u = 2) = 0 \]
\[ P(u = 1) = 0, \quad P(u = 2) = 1 \]

Eleven mixed strategies are implemented for each function, corresponding to probability of algorithm selection

\[ P(u = 1) = 0.1 \times j \quad \text{for} \quad j = 0, 1, 2, .., 10 \quad (6.3) \]

Consequently, the total number of behavioral strategies of the organization are 121.

6.10 Design Evaluation

Software has been developed for the computation of the total activity of the functions, and the Accuracy of the four organizations, as well as for the computation of the Response Time and Throughput Rate, which correspond to the minimum rationality threshold value \((F_o)_{min}\). The value used for \((F_o)_{min}\) is 5 bits/sec [Miller (1956)]. The time delay across communication links was set equal to 2 seconds for all links. The ranges of the values of the MOPs for each organization are shown in Table 6.6. Finally, the Measure of Effectiveness, \(Q\), has been computed for each organization, parameterized by the requirements on Accuracy \((J_o)\), Response Time \((T_o)\), and Throughput Rate \((R_o)\).

For each of the four organizations, the MOP locus and its projections on the Accuracy - Response time \((J - T)\), on the Accuracy - Throughput Rate \((J - R)\), and on the Response Time - Throughput Rate \((T - R)\) planes are shown in Figures 6.33 to 6.48.

From these plots we conclude that organizations 1 and 2 have the same Accuracy for every behavioral strategy. The two organizations implement identical algorithms;
Table 6.6: Measures of Performance

<table>
<thead>
<tr>
<th>Organization</th>
<th>$J$ (sec/symbol)</th>
<th>$T$ (sec/symbol)</th>
<th>$R$ (symbols/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.72 - 2.04</td>
<td>12.86 - 16.41</td>
<td>0.061 - 0.078</td>
</tr>
<tr>
<td>2</td>
<td>0.72 - 2.04</td>
<td>14.86 - 18.41</td>
<td>0.114 - 0.131</td>
</tr>
<tr>
<td>3</td>
<td>0.32 - 2.04</td>
<td>13.48 - 13.70</td>
<td>0.073 - 0.074</td>
</tr>
<tr>
<td>4</td>
<td>0.32 - 2.04</td>
<td>13.48 - 13.70</td>
<td>0.146 - 0.217</td>
</tr>
</tbody>
</table>

thus they obtain identical Accuracy for corresponding behavioral strategies. The Response Time of organization 2 is equal to the Response Time of organization 1 plus 2 seconds (because the dominant information flow path contains the transition representing the communication between the two decisionmakers on the same E2C aircraft, which requires 2 seconds). Finally, organization 2 has better Throughput Rate than organization 1, as expected.

Organizations 3 and 4 have also identical Accuracy for corresponding behavioral strategies. For both organizations the same information flow path has the largest processing time; this path does not contain the communication transition between the two decisionmakers on the same E2C aircraft. Consequently, the two organizations have equal Response Time for corresponding behavioral strategies. Finally, organization 4 achieves a higher Throughput rate, as expected, and therefore has a higher Measure of Effectiveness than organization 3 for every set of requirements.

It is also observed, that all four organizations have comparable Accuracy. We can also conclude that organizations 1 and 2 have, in general, greater Response Time than organizations 3 and 4. The reason for this is that while the decisionmaker on the carrier implements the global situation assessment and the global response selection, the personnel on each E2C aircraft develop the local courses of action (increased parallelism of the information processing). Finally we note that organization 4 has greater Throughput Rate that organization 2, and organization 3 has greater Throughput Rate that organization 1.
Figure 6.33: Organization 1, MOP locus

Figure 6.34: Organization 1, MOP locus projection on the $J - T$ plane
Figure 6.35: Organization 1, MOP locus projection on the $J - R$ plane

Figure 6.36: Organization 1, MOP locus projection on the $R - T$ plane
Figure 6.37: Organization 2, MOP locus

Figure 6.38: Organization 2, MOP locus projection on the $J - T$ plane
Figure 6.39: Organization 2, MOP locus projection on the $J - R$ plane

Figure 6.40: Organization 2, MOP locus projection on the $R - T$ plane
Figure 6.41: Organization 3, MOP locus

Figure 6.42: Organization 3, MOP locus projection on the $J - T$ plane
Figure 6.43: Organization 3, MOP locus projection on the $J - R$ plane

Figure 6.44: Organization 3, MOP locus projection on the $R - T$ plane
Figure 6.45: Organization 4, MOP locus

Figure 6.46: Organization 4, MOP locus projection on the $J - T$ plane
Figure 6.47: Organization 4, MOP locus projection on the $J - R$ plane

Figure 6.48: Organization 4, MOP locus projection on the $R - T$ plane
The Measure of Effectiveness, $Q$, has been computed for all four organizations, parameterized by the values of the requirements. Plots of $Q$ have been obtained versus the Accuracy ($J_o$) and the Response Time ($T_o$) requirements for a fixed value of Throughput Rate requirement $R_o$, versus the Accuracy ($J_o$) and the Throughput Rate ($R_o$) requirements for a fixed value of the Response Time requirement ($T_o$), and versus the Response Time ($T_o$) and Throughput Rate ($R_o$) requirements for a fixed value of the Accuracy requirement ($J_o$). The fixed value of the third requirement for each organization is set equal to the value that is satisfied by all behavioral strategies. These plots are shown in Figures 6.49 through 6.60.

For given requirements, the value of the MOE, $Q$, is easily obtained for each organization, and the design with the highest MOE value is selected.

In this example, a qualitative comparison of the four organizations is possible on the $Q$ versus $T_o - R_o$ plots, since all four organizations have comparable Accuracy. From these $Q$ plots we deduce that organization 4 has the highest MOE, for all combinations of Response Time and Throughput Rate requirements. Thus, organization 4 is the best design.

Note that a set of surfaces is obtained for $Q$, when plotted versus two of the requirements, as the third requirement is varied. Figures 6.61 and 6.62 depict the $Q$ surface of organizations 2 and 4, when the requirement for Accuracy is $J_o = 1.2$. As expected, organization 4 has a higher $Q$ value than organization 2, for all combinations of Response Time and Throughput Rate requirements.

Finally, when comparing the $Q$ versus $T_o - R_o$ plots of organizations 2 and 3, we observe that organization 2 has a better MOE for high throughput rate and large response time requirements. In contrast, organization 3 has a better MOE for small response time and low throughput rate requirements. Thus, the two designs trade-off Response Time and Throughput Rate (Timeliness and Processing Capacity). Depending on the values of the requirements, one of the two organizations may be appropriate for the task.
Figure 6.49: Organization 1, MOE versus $J_o - T_o$ requirements

Figure 6.50: Organization 1, MOE versus $J_o - R_o$ requirements
Figure 6.51: Organization 1, MOE versus $T_o - R_o$ requirements

Figure 6.52: Organization 2, MOE versus $J_o - T_o$ requirements
Figure 6.53: Organization 2, MOE versus $J_0 - R_o$ requirements

Figure 6.54: Organization 2, MOE versus $T_0 - R_o$ requirements
Figure 6.55: Organization 3, MOE versus $J_o - T_o$ requirements

Figure 6.56: Organization 3, MOE versus $J_o - R_o$ requirements
Figure 6.57: Organization 3, MOE versus $T_o - R_o$ requirements

Figure 6.58: Organization 4, MOE versus $J_o - T_o$ requirements
Figure 6.59: Organization 4, MOE versus $J_o - R_o$ requirements

Figure 6.60: Organization 4, MOE versus $T_o - R_o$ requirements
Figure 6.61: Organization 2, MOE versus $T_o - R_o$ requirements, $J_o = 1.2$

Figure 6.62: Organization 4, MOE versus $T_o - R_o$ requirements, $J_o = 1.2$
6.11 Chapter Summary

In this chapter, the synthesis methodology has been applied to the design of $C^2$ organizations for the outer air battle. Fourteen Data Flow Structures were generated by phase 1, and two were selected to apply phases 2, and 3. Four organizations were designed (two from each Data Flow Structure). The Measures of Performance and Measure of Effectiveness were computed for each design. Through the qualitative comparison of the MOE of the four organizations, we have selected one organization that has the best MOE among the four designs, for any set of requirements.
Chapter 7

A GLOBAL VIEW

7.1 Introduction

In this chapter, we develop a framework for the analysis and synthesis of $C^2$ organizations which change during their operation in a hostile environment. This framework is the basis for future research directions. Specifically, the concepts of survivability and reconfigurability are investigated. A measure of survivability is defined, and the procedure for its computation is developed. Reconfigurability is examined from the vantage point of synthesis of a set of $C^2$ organizations, which are implemented as the $C^2$ assets of the initial design are physically lost. The reconfigurability problem is formulated and is shown to be an extension of the synthesis problem; consequently it can be tackled using the design methodology developed in Chapter 5. Finally, a summary of the thesis is presented, and directions for future research are described.

7.2 Survivability

A $C^2$ organization is likely to sustain loss of $C^2$ assets during its operation in a hostile environment. $C^2$ assets are the platforms that carry the processing elements (decisionmakers, decision support systems, etc) of the organization. The loss of $C^2$ assets affects the operation and effectiveness of the organization.

Survivability is the quality of $C^2$ organizations to maintain vital functional properties, when operating in a hostile environment.

Functional properties are dependent on the functions implemented by the surviving assets and their connectivity. In this context, if the data flow structure of the surviving
\(C^2\) is capable of accurate and timely response and can handle the rate of the incoming threats, the surviving \(C^2\) organization maintains its functional properties.

The measure of survivability of the \(C^2\) organization can be expressed by the probability that the organization will maintain its functional properties during its operation.

### 7.3 Computation of Survivability Measure

The physical state of the \(C^2\) organization may be defined by the set of surviving \(C^2\) assets. When an asset is physically lost, or temporarily disabled, then the organization undergoes a state transition. Similarly, when a previously disabled asset is enabled, the organization undergoes a transition.

The evolution of the physical state of a \(C^2\) organization may be modeled as a jump process. In the case that the events occurring are only physical loss of assets, the process is a pure death process, whereas in the case of assets being temporarily disabled, the process is a birth and death process.

Let us consider the case of a pure death process. Assume that there exists only one absorbing state, corresponding to the physical loss of all assets. With each non-absorbing state \(x\), there is associated a distribution function \(F_x(t)\), and transition probabilities from state \(x\) to state \(y\), \(Q_{xy}\), which are non-negative and such that \(Q_{xx} = 0\), and

\[
\sum_y Q_{xy} = 1 \quad (7.1)
\]

A process starting at state \(x\) remains at that state for a random length of time \(\tau\), having distribution \(F_x(\tau)\), and then jumps to state \(y\) with probability \(Q_{xy}\). It may be assumed that \(\tau\) and the jump to state \(y\) are independent of each other so that the probability that a transition occurs after a time interval \(\tau\) from state \(x\) to state \(y\), \(P_{xy}(\tau < t)\) is:

\[
P_{xy}(\tau < t) = F_x(t)Q_{xy} \quad (7.2)
\]

It may also be assumed that the parameters of the distribution function \(F_x\) are to a
first approximation functions of the strength of friendly and enemy forces.

Define the satisficing states as those physical states that satisfy the minimal functional requirements, (or alternatively, the states whose MOE exceeds a threshold).

A sequence of physical state transitions is called a transition sequence. To each transition sequence, corresponds a physical state trajectory.

The terminal state of a transition sequence is the first physical state in the trajectory that is not satisficing. Note that the terminal state of a transition sequence is not necessarily the final state of the sequence, and does not correspond to the absorbing state of Markov chains.

Define the degradation time $T_d$ to be the time elapsed from the beginning of the battle until the $C^2$ organization reaches a terminal state.

Define the terminal time $T_f$ as the time at which the battle ends.

Consider the physical state trajectory obtained for a specific transition sequence. At some transition, the terminal state is reached. Compute the pdf of the time elapsed from the start of the process to the arrival at that terminal state by convolving the corresponding pdfs. This is the pdf of the degradation time for the trajectory.

The pdf of degradation time of the $C^2$ organization is the probability weighted sum of the pdfs of degradation times of all state trajectories.

The survivability of the $C^2$ organization is evaluated by the probability that the organization will be in a satisficing state at the terminal time $T_f$.

A measure of survivability is defined by the probability that the degradation time, $T_d$, is greater than the terminal time $T_f$. Thus the measure of survivability $S$ is:

$$S = P(T_d > T_f) = \int_{T_f}^{\infty} g(T_d) dT_d$$

(7.3)

In order to compute the terminal time $T_f$ and the parameters of the pdfs of the transition times, the strength of friendly and enemy forces must be computed as a func-
tion of time, which is achieved using Lanchester type equations of attrition [Woodcock (1986)].

Let $x$ and $y$ be the strength of friendly and enemy forces. Let $a$ and $b$ be the corresponding attrition coefficients.

The simple form of Lanchester equations is:

\[
\begin{align*}
\frac{dx}{dt} &= -by \\
\frac{dy}{dt} &= -ax
\end{align*}
\]  

Let $n(t)$ represent the percentage of friendly forces at time $t$ and $m(t)$ be the percentage of enemy forces at time $t$.

\[
\begin{align*}
  n(t) &= \frac{x_o - x(t)}{x_o} \\
  m(t) &= \frac{y_o - y(t)}{y_o}
\end{align*}
\]  

the terminal conditions are

\[
\begin{align*}
  n(T_f) &= n^* \\
  m(T_f) &= m^*
\end{align*}
\]  

i.e. the battle terminates at $T_f$ when either $n(T_f) = n^*$, in which case the enemy forces win the battle or when $m(T_f) = m^*$, in which case the friendly forces win the battle.

The terminal time $T_f$ is determined by numerical computation. We will assume that the pdfs of transition times are known, and their parameters are functions of the ratio $(y/x)$.

For given sets of values of $x_o$, $y_o$, $n^*$, $m^*$, the strength $x$ and $y$, and the terminal time $T_f$ are computed. Then all the physical state trajectories are identified and the pdf of the degradation time $T_d$ is obtained. Finally, the measure of survivability $S$ is computed.
7.4 Reconfigurability

Reconfigurability is the ability of a $C^2$ organization to undo a series of structural transformations, which may be pre-scheduled (through contingency plans), when $C^2$ assets are lost, and thus implement a processing schema that is satisficing.

The objective of reconfigurability is to re-allocate functions to the remaining assets, so that they operate effectively. When seen in the previously defined context of survivability, the objective is to push the terminal states in time, as far away (as late) from the beginning of the process. In other words, before the $C^2$ organization reaches a terminal state, we seek to reallocate the functions to the remaining $C^2$ assets so that the functional properties of the $C^2$ organization are preserved, even with fewer assets.

The reconfiguration is a design problem: given the remaining assets, we seek to design a $C^2$ organization which has the desired functional properties and has a high MOE value. This is exactly the synthesis problem formulated in Chapter 5, and we have already developed a methodology for its solution.

Reconfigurability requires the development and implementation of a coordinating algorithm that monitors the physical state of the organization and commands the adoption of a suitable data flow structure, based on the status of the surviving processing nodes and existing communications links.

The issues that arise are the following:

Which $C^2$ assets should implement the monitoring algorithm, and what is the required redundancy of the monitoring algorithm, i.e., how many assets should implement the algorithm.

What is the effect of the monitoring algorithm(s) on the measures of performance, specifically response time and throughput rate.
7.5 Summary and Conclusions

This thesis has addressed both the analysis and synthesis of $C^2$ organizations, has investigated the concepts of timeliness and survivability, has defined the corresponding measures of performance and has derived procedures for their computation. The major contribution of the thesis in the theory of Command and Control is the methodology for the synthesis of $C^2$ organizations.

The methodology tackles the design problem at two levels: the data flow structure level and the organization architecture level. The significance of this differentiation is the ability to generate and classify structures parameterized by the complexity and redundancy of the information processing. After the generation of the candidate data flow structures, the methodology addresses the allocation of functions to organization members and the selection of the supporting software and hardware.

In this respect the methodology is a flexible top-down approach to the design problem, that results in the expansion of the set of candidate architectures. A potential benefit from the top-down approach is that the requirements and specifications for decision aids, databases, and communications equipment may be derived through the objective evaluation of the effectiveness of the $C^2$ organization.

Finally the distinction between the data flow structure design and the organization architecture design introduces two opportunities for the fine-tuning of the $C^2$ organization: one at the data flow level and one at the decisionmaker and system level.

7.6 Directions for Future Research

Directions for future research extend in three areas; namely, survivability and reconfigurability, derivation of requirements for the supporting $C^2$ system components, and experimental investigation of the processing capacity of the human decisionmaker.

Survivability is a crucial aspect of the design of $C^2$ organizations. In this thesis a definition of survivability and the corresponding measure have been introduced, and
the framework for the computation of the measure of survivability has been developed. Further investigation of this approach is required to derive efficient algorithms or analytic solutions for the computation of the measure of survivability.

Reconfigurability poses an interesting design problem. Use of the design methodology leads to the design of alternate decision-making organizations. Each organization is appropriate within a range of the problem parameters. The development of the reconfiguration algorithm is an important issue for the synthesis of flexible $C^2$ organizations.

The development of requirements for the supporting $C^3$ system is a problem that may be tackled through the analysis and synthesis of $C^2$ organizations. This problem is formulated as follows: given the decision-making organization, derive the requirements for the supporting software and hardware, so that a desired MOE value is obtained.

Finally, experimental evaluation of the rationality threshold and its probability density function is necessary, so that the calculated processing times are realistic. To this effect, the decision-making procedures of the naval air defense example in Chapter 6 may be used. These procedures involve only logical processing of the data as opposed to numerical computation, and are suitable for the measurement of the processing rate of the human decisionmaker.
REFERENCES


177
Appendix A

ALTERNATE ALGORITHMS’ ACTIVITY COMPUTATION

A.1 Algorithm Selection Independent of Input Value

Consider a function which is implemented by \( k \) deterministic algorithms \( f_1, f_2, ..., f_k \). The selection of the active algorithm is represented by the decision variable \( u \). First consider the probability of algorithm selection \( p(u = i) \) to be independent of the value of the input \( x \) to the function. Let algorithm \( f_i \) contain \( a_i \) internal variables, denoted by \( w^i_j \), where \( j = 1, 2, ..., a_i \). Let \( z \) be the output of the function.

First prove the noise term. By definition

\[ G_n = H_z(u, W, z) \quad (A.1) \]

where

\[ W = w^1_1, w^1_2, ..., w^1_{a_1}, w^2_1, w^2_2, ..., w^2_{a_2}, ..., w^k_1, w^k_2, ..., w^k_{a_k} \quad (A.2) \]

\[ H_z(u, W, z) = H_z(u) + H_{z,u}(W) + H_{z,u,W}(z) \quad (A.3) \]

since the algorithms are deterministic, knowledge of the input \( x \) and the decision variable \( u \), leaves no uncertainty in the internal variables \( W \), and the output \( z \). Consequently

\[ H_{z,u}(W) = 0 \quad (A.4) \]
\[ H_{z,u,W}(z) = 0 \quad (A.5) \]

Finally

\[ H_z(u, W, z) = H_z(u) = H(u) \quad (A.6) \]
since the decision variable $u$ is independent of $z$.

By definition the coordination $G_c$ of the function is the sum of the entropies of the function variables minus their joint entropy. Hence,

$$G_c = \sum_{i=1}^{k} \sum_{j=1}^{a_i} H(w_j^i) + H(u) + H(z) - H(W, u, z) \quad (A.7)$$

$$H(W, u, z) = H(u, W, z) = H(u) + H_u(W) + H_{u,W}(z) \quad (A.8)$$

The uncertainty in $z$ given the values of $u$ and $W$ is zero. Thus

$$H_{u,W}(z) = 0 \quad (A.9)$$

$$G_c = \sum_{i=1}^{k} \sum_{j=1}^{a_i} H(w_j^i) + H(u) + H(z) - [H(u) + H_u(W)] \quad (A.10)$$

$$G_c = \sum_{i=1}^{k} \sum_{j=1}^{a_i} H(w_j^i) + H(z) - H_u(W) \quad (A.11)$$

$$G_c = \sum_{i=1}^{k} \sum_{j=1}^{a_i} H(w_j^i) + H(z) - H_u(W) + \sum_{i=1}^{k} \sum_{j=1}^{a_i} H_u(w_j^i) - \sum_{i=1}^{k} \sum_{j=1}^{a_i} H_u(w_j^i) \quad (A.12)$$

$$G_c = \sum_{i=1}^{k} \sum_{j=1}^{a_i} [H(w_j^i) - H_u(w_j^i)] + H(z) - H_u(W) + \sum_{i=1}^{k} \sum_{j=1}^{a_i} H_u(w_j^i) \quad (A.13)$$

Consider the term

$$[H(w_j^i) - H_u(w_j^i)] = T(u : w_j^i) = H(u) - H_{w_j^i}(u) \quad (A.14)$$

$$H(u) - H_{w_j^i}(u) = -\sum_{u=1}^{k} [p(u) \log_2 p(u)] + \sum_{w_j^i} p(w_j^i) \sum_{u=1}^{k} [p(u|w_j^i) \log_2 p(u|w_j^i)] \quad (A.15)$$
Variable $w_j^i$ is active when algorithm $i$ is active, i.e. $u = i$, otherwise it is inactive. Hence, if variable $w_j^i$ is active, there is no uncertainty in the value of $u$; clearly $u = i$. Thus

\[ p(u = i|w_j^i \text{ active}) = 1 \quad p(u = i|w_j^i \text{ inactive}) = 0 \] (A.16)

and

\[ p(w_j^i \text{ active}) = p(u = i) \quad p(w_j^i \text{ inactive}) = p(u \neq i) \] (A.17)

Consider the term

\[
\sum_{u=1}^{k} \left[ p(u|w_j^i) \log_2 p(u|w_j^i) \right] = p(u = i|w_j^i) \log_2 p(u = i|w_j^i)
\]

\[
+ \sum_{u \neq i} \left[ p(u|w_j^i) \log_2 p(u|w_j^i) \right] =
\]

\[
p(u = i|w_j^i \text{ active}) \log_2 p(u = i|w_j^i \text{ active})
\]

\[
+ p(u = i|w_j^i \text{ inactive}) \log_2 p(u = i|w_j^i \text{ inactive})
\]

\[
+ \sum_{u \neq i} \left[ p(u|w_j^i) \log_2 p(u|w_j^i) \right] =
\]

\[
= 1 \log_2 1 + 0 \log_2 0 + \sum_{u \neq i} \left[ p(u|w_j^i) \log_2 p(u|w_j^i) \right]
\]

\[
= \sum_{u \neq i} \left[ p(u \neq i|w_j^i) \log_2 p(u \neq i|w_j^i) \right] \quad (A.18)
\]

Note that

\[
p(u : u \neq i|w_j^i) = \frac{p(u)}{p(w_j^i \text{ inactive})} = \frac{p(u)}{p(u \neq i)} \quad (A.19)
\]

Thus,

\[
\sum_{u=1}^{k} \left[ p(u|w_j^i) \log_2 p(u|w_j^i) \right] = \sum_{u \neq i} \left[ p(u \neq i|w_j^i) \log_2 p(u \neq i|w_j^i) \right]
\]

\[
= \sum_{u \neq i} \left[ \frac{p(u)}{p(u \neq i)} \right] \log_2 \frac{p(u)}{p(u \neq i)} \quad (A.20)
\]

Substituting obtain

180
\[ H(u) - H_{w_i}(u) = -\sum_{u=1}^{k} [p(u) \log_2 p(u)] + \sum_{\text{inactive } w_j} p(w_j) \sum_{u \neq i} \left[ \frac{p(u)}{p(u \neq i)} \right] \log_2 \left( \frac{p(u)}{p(u \neq i)} \right) \] (A.21)

\[ H(u) - H_{w_i}(u) = -\sum_{u=1}^{k} [p(u) \log_2 p(u)] + p(u \neq 1) \sum_{u \neq i} \left[ \frac{p(u)}{p(u \neq i)} \right] \log_2 \left( \frac{p(u)}{p(u \neq i)} \right) \] (A.22)

\[ H(u) - H_{w_i}(u) = -\sum_{u=1}^{k} [p(u) \log_2 p(u)] + \sum_{u \neq i} \left[ p(u) \log_2 \left( \frac{p(u)}{p(u \neq i)} \right) \right] \] (A.23)

\[ H(u) - H_{w_i}(u) = -\sum_{u=1}^{k} [p(u) \log_2 p(u)] + \sum_{u \neq i} [p(u) \log_2 p(u) - p(u) \log_2 p(u \neq i)] \] (A.24)

\[ H(u) - H_{w_i}(u) = -\sum_{u=1}^{k} [p(u) \log_2 p(u)] + \sum_{u \neq i} [p(u) \log_2 p(u)] - \sum_{u \neq i} [p(u) \log_2 p(u \neq i)] \] (A.25)

\[ H(u) - H_{w_i}(u) = -p(u = i) \log_2 p(u = i) - \log_2 p(u \neq i) \sum_{u \neq i} p(u) \]

\[ = -p(u = i) \log_2 p(u = i) - [\log_2 p(u \neq i)] p(u \neq i) \]

\[ = \mathcal{H}(p(u = i)) \] (A.26)

Consequently (A.8) becomes

\[ G_c = \sum_{i=1}^{k} \sum_{j=1}^{a_i} [H(w_j) - H_u(w_j)] + H(z) - H_u(W) + \sum_{i=1}^{k} \sum_{j=1}^{a_i} H_u(w_j) = \]

\[ = \sum_{i=1}^{k} \sum_{j=1}^{a_i} [\mathcal{H}(p(u = i))] + H(z) - H_u(W) + \sum_{i=1}^{k} \sum_{j=1}^{a_i} H_u(w_j) = \]

\[ = \sum_{i=1}^{k} a_i [\mathcal{H}(p(u = i))] + H(z) + \sum_{i=1}^{k} \sum_{j=1}^{a_i} H_u(w_j) - H_u(W) \] (A.27)
Expand the last term

\[ H_u(W) = H_u(W^1, W^2, \ldots, W^k) = \]
\[ = H_u(W^1) + H_{u,w^1}(W^2) + H_{u,w^1,w^2}(W^3) + \ldots \]
\[ + H_{u,w^1,w^2,\ldots,w^{k-1}}(W^k) \]  
(A.28)

where

\[ W^k = w^k_1, w^k_2, \ldots, w^k_{a_k} \]  
(A.29)

is the set of internal variables of algorithm \( k \).

Because the values that are obtained by the sets of internal variables of the algorithms are independent, knowledge of the values of the other algorithms' variables does not reduce their uncertainty. Thus

\[ H_u(W) = H_u(W^1, W^2, \ldots, W^k) = H_u(W^1) + H_u(W^2) + H_u(W^3) + \ldots + H_u(W^k) \]  
(A.30)

\[ G_c = \sum_{i=1}^{k} a_i [\mathcal{H}(p(u = i))] + H(z) + \sum_{i=1}^{k} \sum_{j=1}^{a_i} H_u(w^i_j) - \sum_{i=1}^{k} H_u(W^i) = \]
\[ = \sum_{i=1}^{k} a_i [\mathcal{H}(p(u = i))] + H(z) + \sum_{i=1}^{k} \left[ \sum_{j=1}^{a_i} H_u(w^i_j) - H_u(W^i) \right] \]  
(A.31)

Consider the terms in brackets

\[ \sum_{j=1}^{a_i} H_u(w^i_j) - H_u(W^i) = \sum_{j=1}^{a_i} \left[ -\sum_u p(u) \sum_{w^i_j} p(w^i_j|u) \log_2 p(w^i_j|u) \right] \]
\[ - \left[ -\sum_u p(u) \sum_{W^i} p(W^i|u) \log_2 p(W^i|u) \right] \]  
(A.32)

For \( u \neq i \) the values of variables \( w^i_j \) are inactive, and have no uncertainty. Hence

\[ \sum_{j=1}^{a_i} H_u(w^i_j) - H_u(W^i) = \]

182
\[
= p(u = i) \sum_{j=1}^{a_i} \left[ -\sum_{w_j} p(w_j | u = i) \log_2 p(w_j | u = i) \right] \\
- p(u = i) \left[ -\sum_{W^i} p(W^i | u = i) \log_2 p(W^i | u = i) \right]
\]

the term
\[
\sum_{j=1}^{a_i} \left[ -\sum_{w_j} p(w_j | u = i) \log_2 p(w_j | u = i) \right] - \left[ -\sum_{W^i} p(W^i | u = i) \log_2 p(W^i | u = i) \right] = \\
\sum_{j=1}^{a_i} H(w_j^i) - H(W^i)
\]
is the coordination of algorithm \(i\), \(G_{ci}\). Thus
\[
\sum_{j=1}^{a_i} H_u(w_j^i) - H_u(W^i) = p(u = i) G_{ci}
\]
Finally,
\[
G_c = \sum_{i=1}^{k} a_i \mathcal{H}(p(u = i)) + H(z) + \sum_{i=1}^{k} \left[ \sum_{j=1}^{a_i} H_u(w_j^i) - H_u(W^i) \right] = \\
\sum_{i=1}^{k} a_i [\mathcal{H}(p(u = i))] + H(z) + \sum_{i=1}^{k} p(u = i) G_{ci}
\]

A.2 Algorithm Selection Dependent on Input Value

In the case of algorithm selection dependent on the value of the input \(z\) to the function, i.e., when the probabilities of algorithm selection are conditioned on the input value, the development of the coordination computation is identical, and leads to the same expression as in the case presented above. Note however, that the probabilities of algorithm use must be computed, using Baye’s rule, as mentioned in Chapter 3.

The noise term \(G_n\) is
\[
G_n = H_z(u, W, z) = H_z(u) + H_{z,u}(W) + H_{z,u,w}(z)
\]
Since the algorithms are deterministic, knowledge of the input $z$ and the decision variable $u$, leaves no uncertainty in the internal variables $W$, and the output $z$. Consequently

\begin{align*}
H_{x,u}(W) &= 0 \\ 
H_{x,u,W}(z) &= 0
\end{align*}

(A.37) 
(A.38)

Finally

\[ H_x(u, W, z) = H_x(u) \] 

(A.39)
Appendix B

ACTIVITY COMPUTATION EXAMPLE

A detailed example will be used to illustrate the application of Information theory to the workload computation of decision-making algorithms. The total activity of two algorithms, and the total activity of a function implemented alternatively by the two algorithms are computed.

B.1 Activity Computation of First Algorithm

The input $X$ to the algorithm has two attributes $X_1$ and $X_2$ and their values and corresponding probability are shown in Table B.1.

The output of the algorithm $Y$ takes on the values $\alpha$, $\beta$ and $\gamma$. The algorithm produces the value of $Y$ according to the flowchart depicted in Figure B.1.

The input, output and the respective probabilities are shown in Table B.2.

Table B.1: Input alphabet and corresponding probability

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>red</td>
<td>0.25</td>
</tr>
<tr>
<td>slow</td>
<td>red</td>
<td>0.25</td>
</tr>
<tr>
<td>fast</td>
<td>blue</td>
<td>0.25</td>
</tr>
<tr>
<td>slow</td>
<td>blue</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Figure B.1: Flowchart of first algorithm

Table B.2: Output $Y$ of first algorithm

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>probability</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>red</td>
<td>0.25</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>slow</td>
<td>red</td>
<td>0.25</td>
<td>$\beta$</td>
</tr>
<tr>
<td>fast</td>
<td>blue</td>
<td>0.25</td>
<td>$\beta$</td>
</tr>
<tr>
<td>slow</td>
<td>blue</td>
<td>0.25</td>
<td>$\gamma$</td>
</tr>
</tbody>
</table>
The algorithm variables \( w_1, w_2, w_3, w_4, \) and \( w_5 \) are defined as follows:

\[
\begin{align*}
    w_1 & = X \\
    w_2 & = (X_1 = fast) \\
    w_3 & = (X_2 = red | X_1 = fast) \\
    w_4 & = (X_2 = red | X_1 = slow) \\
    w_5 & = Y
\end{align*}
\]

Note that \( w_2, w_3, \) and \( w_4 \) are logical variables, obtaining the values true or false.

The entropy of the input is:

\[
\begin{align*}
    H(X) & = H(w_1) = - \sum_{i=1}^{4} [p(X = x_i) \log_2 p(X = x_i)] \\
          & = - \sum_{i=1}^{4} \frac{1}{4} \log_2 \frac{1}{4} = -4 \left( \frac{1}{4} \log_2 \frac{1}{4} \right) = \log_2 4 = 2 \text{ bits/symbol}
\end{align*}
\]

The variables \( w_2, w_3, \) and \( w_4 \) take on the values true and false with probability 0.5 each. Thus their entropy is 1 bit.

\[
H(w_2) = H(w_3) = H(w_4) = 1 \text{ bit/symbol}
\]

Finally, the output \( Y \) takes on the values \( \alpha, \beta \) and \( \gamma \) with probability 0.25, 0.5, and 0.25. Thus the entropy of \( Y \) is:

\[
\begin{align*}
    H(w_5) & = H(Y) = - \sum_{i=1}^{3} [p(Y = y_i) \log_2 p(Y = y_i)] \\
            & = - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} \\
            & = \frac{1}{4} \log_2 4 + \frac{1}{2} \log_2 2 + \frac{1}{4} \log_2 4 = 1.5 \text{ bits/symbol}
\end{align*}
\]

The total activity \( G \) of the algorithm is:

\[
G = \sum_{i=1}^{5} H(w_i) = 2 + 1 + 1 + 1 + 1.5 = 6.5 \text{ bits/symbol}
\]
For a deterministic system the coordination $G_c$ of the variables is:

\[
G_c = \sum_{i=1}^{n} H(w_i) - H(w_1)
\]

Thus, the coordination is:

\[
G_c = 1 + 1 + 1 + 1.5 = 4.5 \text{ bits/symbol}
\]

and the noise $G_n$ is $G_n = 0$.

The mutual information or throughput $T(X : Y)$ is:

\[
T(X : Y) = H(X) - H_Y(X) = H(Y) - H_X(Y)
\]

The conditional entropy of the output $Y$, when the input $X$ is known, $H_Y(X)$, is:

\[
H_Y(X) = - \sum_{i=1}^{l} p(Y = y_i) \sum_{j=1}^{k} p(X = x_j | Y) \log_2 p(X = x_j | Y)
\]

The four cases with non-zero conditional probability are:

\[
\begin{align*}
    p(X= \text{fast, red} | Y = \alpha) &= 1 \\
p(X= \text{fast, blue} | Y = \beta) &= \frac{1}{2} \\
p(X= \text{slow, red} | Y = \beta) &= \frac{1}{2} \\
p(X= \text{slow, blue} | Y = \gamma) &= 1
\end{align*}
\]

and

\[
    P(Y = \alpha) = \frac{1}{4}, \quad P(Y = \beta) = \frac{1}{2}, \quad P(Y = \gamma) = \frac{1}{4}.
\]

Thus,

\[
H_Y(X) = -\frac{1}{4} \left[ \log_2 1 + 0 \log_2 0 \right] - \frac{1}{2} \left[ \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right] - \frac{1}{4} \left[ \log_2 2 \right] = 0.5 \text{ bits/symbol}
\]

The throughput (transmission) is:

\[
G_t = T(X : Y) = H(Y) - H_X(Y) = 1.5 - 0.5 = 1.0 \text{ bit/symbol}
\]
Table B.3: Input alphabet

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>probability</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>red</td>
<td>0.125</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>slow</td>
<td>red</td>
<td>0.250</td>
<td>$\beta$</td>
</tr>
<tr>
<td>fast</td>
<td>blue</td>
<td>0.500</td>
<td>$\beta$</td>
</tr>
<tr>
<td>slow</td>
<td>blue</td>
<td>0.125</td>
<td>$\gamma$</td>
</tr>
</tbody>
</table>

Finally, the blockage $G_b$ is:

$$G_b = H(X) - G_t = 2 - 1.0 = 1.0 \text{ bit/symbol}$$

In order to show how the total activity, throughput, blockage and coordination vary with the probability distribution of the input, the mass function of $X$ is changed as shown in Table B.3.

The entropy of the input is:

$$H(w_1) = H(X) = -\sum_{i=1}^{4} [p(X = x_i) \log_2 p(X = x_i)]$$

$$= -\frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{8} \log_2 \frac{1}{8} = 1.75 \text{ bits/symbol}$$

The entropies of the variables $w_2, w_3, \text{ and } w_4$, are:

$$H(w_2) = -\frac{5}{8} \log_2 \frac{5}{8} - \frac{3}{8} \log_2 \frac{3}{8} = 1.5064 \text{ bits/symbol}$$

$$H(w_3) = -\frac{1}{5} \log_2 \frac{1}{5} - \frac{4}{5} \log_2 \frac{4}{5} = 0.7219 \text{ bits/symbol}$$

$$H(w_4) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0.9118 \text{ bits/symbol}$$

Finally, the output $Y$ takes on the values $\alpha, \beta$ and $\gamma$ with probability 0.125, 0.75, and 0.125. Thus the entropy of $Y$ is:
$$H(w_5) = H(Y) = -\frac{1}{8} \log_2 \frac{1}{8} - \frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{8} \log_2 \frac{1}{8}$$
$$= \frac{1}{8} \log_2 8 + \frac{3}{4} \log_2 3 + \frac{1}{8} \log_2 8 = 1.0612 \text{ bits/symbol}$$

The total activity $G$ of the algorithm is:

$$G = \sum_{i=1}^{5} H(w_i) = 1.75 + 1.5064 + 0.7219 + 0.9118 + 1.0612 = 5.9513 \text{ bits/symbol}$$

The coordination $G_c$ of the algorithm is:

$$G_c = 1.5064 + 0.7219 + 0.9118 + 1.0612 = 4.2013 \text{ bits/symbol}$$

The conditional entropy of the output $Y$, when the input $X$ is known, $H_Y(X)$, is:

$$H_Y(X) = -\sum_{i=1}^{l} p(Y = y_i) \sum_{j=1}^{k} p(X = x_j|Y) \log_2 p(X = x_j|Y)$$

the four cases with non-zero conditional probability are:

- $P(X = \text{fast, red} \mid Y = \alpha) = 1$
- $P(X = \text{fast, blue} \mid Y = \beta) = \frac{2}{3}$
- $P(X = \text{slow, red} \mid Y = \beta) = \frac{1}{3}$
- $P(X = \text{slow, blue} \mid Y = \gamma) = 1$

and

$$P(Y = \alpha) = \frac{1}{8}, \quad P(Y = \beta) = \frac{3}{4}, \quad P(Y = \gamma) = \frac{1}{8}$$

Thus,

$$H_Y(X) = -\frac{1}{8} \left[1 \log_2 1 + 0 \log_2 0\right] - \frac{3}{4} \left[2 \log_2 \frac{2}{3} + 1 \log_2 \frac{1}{3}\right] - \frac{1}{8} \left[1 \log_2 1 + 0 \log_2 0\right]$$
$$= 0.6887 \text{ bits/symbol}$$

The throughput (transmission) is:

$$G_t = T(X : Y) = H(Y) - H_X(Y) = 1.0612 - 0.6887 = 0.3725 \text{ bits/symbol}$$

Finally, the blockage $G_b$ is:

$$G_b = H(X) - G_t = 1.75 - 0.3725 = 1.3775 \text{ bits/symbol}$$

190
Figure B.2: Flowchart of second algorithm

B.2 Activity Computation of Second Algorithm

Another algorithm may be used to process the data, according to the flowchart depicted in Figure B.2. The input, output and the respective probabilities are shown in Table B.4.

Table B.4: Output $Y$ of second algorithm

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>probability</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>red</td>
<td>0.25</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>slow</td>
<td>red</td>
<td>0.25</td>
<td>$\beta$</td>
</tr>
<tr>
<td>fast</td>
<td>blue</td>
<td>0.25</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>slow</td>
<td>blue</td>
<td>0.25</td>
<td>$\gamma$</td>
</tr>
</tbody>
</table>
The algorithm variables are defined as follows:

\[ w_1 = X \]
\[ w_2 = (X_1 = \text{fast}) \]
\[ w_3 = (X_2 = \text{red}|X_1 = \text{slow}) \]
\[ w_4 = Y \]

The entropy of the variables is:

\[ H(w_1) = H(X) = -\sum_{i=1}^{4} \left[ p(X = x_i) \log_2 p(X = x_i) \right] \]
\[ = -\sum_{i=1}^{4} \left[ \frac{1}{4} \log_2 \frac{1}{4} \right] = -4 \left[ \frac{1}{4} \log_2 \frac{1}{4} \right] = \log_2 4 = 2 \text{ bits/symbol} \]

\[ H(w_2) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ bit/symbol} \]
\[ H(w_3) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ bit/symbol} \]

Finally, the output \( Y \) takes on the values \( \alpha, \beta \) and \( \gamma \) with probability 0.50, 0.25, and 0.25. Thus the entropy of \( w_4 \) is:

\[ H(w_4) = H(Y) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 1.5 \text{ bits/symbol} \]

The total activity \( G \) of the algorithm is:

\[ G = \sum_{i=1}^{4} H(w_i) = 2 + 1 + 1 + 1.5 = 5.5 \text{ bits/symbol} \]

The coordination \( G_c \) of the algorithm is:

\[ G_c = \sum_{i=2}^{4} H(w_i) = 1 + 1 + 1.5 = 3.5 \text{ bits/symbol} \]

192
The conditional entropy of the output \( Y \), when the input \( X \) is known, \( H_Y(X) \), is:

\[
H_Y(X) = -\sum_{i=1}^{l} p(Y = y_i) \sum_{j=1}^{k} p(X = x_j|Y) \log_2 p(X = x_j|Y)
\]

the four cases with non-zero conditional probability are:

\[
\begin{align*}
P(X = \text{fast,red} \mid Y = \alpha) &= \frac{1}{2} \\
P(X = \text{fast,blue} \mid Y = \alpha) &= \frac{1}{2} \\
P(X = \text{slow,red} \mid Y = \beta) &= 1 \\
P(X = \text{slow,blue} \mid Y = \gamma) &= 1
\end{align*}
\]

and

\[
P(Y = \alpha) = \frac{1}{2}, \quad P(Y = \beta) = \frac{1}{4}, \quad P(Y = \gamma) = \frac{1}{4}
\]

Thus,

\[
H_Y(X) = -\frac{1}{2} \left[ \frac{1}{2} \log_2 \frac{1}{2} + 0 \log_2 0 \right] - \frac{1}{4} \left[ 1 \log_2 1 + 0 \log_2 0 \right] - \frac{1}{4} \left[ 1 \log_2 1 + 0 \log_2 0 \right]
\]

\[
= \frac{1}{2} \left[ \frac{1}{2} \log_2 2 \right] = 0.5 \text{ bits/symbol}
\]

The throughput (transmission) is:

\[
G_t = T(X : Y) = H(Y) - H_X(Y) = 1.5 - 0.5 = 1.0 \text{ bit/symbol}
\]

and the blockage \( G_b \) is:

\[
G_b = H(X) - G_t = 2 - 1.0 = 1.0 \text{ bit/symbol}
\]

### B.3 Activity Computation for Alternate Algorithm Use

If the two algorithms can be used alternatively to implement the function, then the total activity of the function is:

\[
G = H(z) + \sum_{i=1}^{2} [ p_i G_{e_i} + a_i \lambda (p_i) ] + H(z) + H(u)
\]
Table B.5: Function Output $Z$, and Algorithm Output $Y_1, Y_2$

| $X_1$ | $X_2$ | $Y_1$ | $Y_2$ | $Z|u = 1$ | $Z|u = 2$ | probability |
|------|------|------|------|----------|----------|-------------|
| fast | red  | $\alpha$ | $\alpha$ | $\alpha$ | $\alpha$ | $Q_1$       |
| slow | red  | $\beta$  | $\beta$  | $\beta$  | $\beta$  | $Q_2$       |
| fast | blue | $\beta$  | $\alpha$ | $\beta$  | $\alpha$ | $Q_3$       |
| slow | blue | $\gamma$ | $\gamma$ | $\gamma$ | $\gamma$ | $Q_4$       |

Note that since $p_1 = 1 - p_2$, $\mathcal{H}(p_1) = \mathcal{H}(p_2)$. Also

$$H(u) = -p(u = 1) \log_2 p(u = 1) - p(u = 2) \log_2 p(u = 2)$$

$$= -p_1 \log_2 p_1 - (1 - p_1) \log_2 (1 - p_1) = \mathcal{H}(p_1)$$

Thus,

$$G = H(X) + p_1 G_{e_1} + p_2 G_{e_2} + (a_1 + a_2 + 1) \mathcal{H}(p_1) + H(z)$$

where $a_1 = 5$ and $a_2 = 4$.

In order to compute the entropy of $z$, the pmf of $z$ must first be obtained. Let $Q_1, Q_2, Q_3, Q_4$ be the probabilities of the 4 inputs. The values of $Z$ and the respective probabilities are shown in Table B.5. Then

$$p(z = \alpha) = p(Y_1 = \alpha|u = 1)p(u = 1) + p(Y_2 = \alpha|u = 2)p(u = 2)$$

$$= Q_1 p_1 + (Q_1 + Q_3) p_2$$

$$p(z = \beta) = p(Y_1 = \beta|u = 1)p(u = 1) + p(Y_2 = \beta|u = 2)p(u = 2)$$

$$= (Q_2 + Q_3) p_1 + Q_2 p_2$$

$$p(z = \gamma) = p(Y_1 = \gamma|u = 1)p(u = 1) + p(Y_2 = \gamma|u = 2)p(u = 2)$$

$$= Q_4 p_1 + Q_4 p_2 = Q_4$$

For $Q_1 = Q_2 = Q_3 = Q_4 = 0.25$ obtain

$$p(z = \alpha) = 0.25(1 + p_2)$$

$$p(z = \beta) = 0.25(1 + p_1)$$

$$p(z = \gamma) = 0.25$$
Table B.6: Probability mass function of output $Z$, and entropy $H(Z)$

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$p(z = \alpha)$</th>
<th>$p(z = \beta)$</th>
<th>$p(z = \gamma)$</th>
<th>$H(Z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.5000</td>
<td>0.2500</td>
<td>0.2500</td>
<td>1.5000</td>
</tr>
<tr>
<td>0.25</td>
<td>0.4375</td>
<td>0.3125</td>
<td>0.2500</td>
<td>1.5462</td>
</tr>
<tr>
<td>0.50</td>
<td>0.3750</td>
<td>0.3750</td>
<td>0.2500</td>
<td>1.5612</td>
</tr>
<tr>
<td>0.75</td>
<td>0.3175</td>
<td>0.4375</td>
<td>0.2500</td>
<td>1.5462</td>
</tr>
<tr>
<td>1.00</td>
<td>0.2500</td>
<td>0.5000</td>
<td>0.2500</td>
<td>1.5000</td>
</tr>
</tbody>
</table>

Table B.7: Coordination $G_c$ as a function of $p_1 = p(u = 1)$

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$p_1G_{c_1}$</th>
<th>$p_2G_{c_2}$</th>
<th>$\mathcal{A}(p)$</th>
<th>$(a_1 + a_2)\mathcal{A}(p)$</th>
<th>$H(Z)$</th>
<th>$G_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.000</td>
<td>3.500</td>
<td>0.000</td>
<td>0.000</td>
<td>1.5000</td>
<td>5.000</td>
</tr>
<tr>
<td>0.25</td>
<td>1.125</td>
<td>2.625</td>
<td>0.811</td>
<td>7.299</td>
<td>1.5462</td>
<td>12.595</td>
</tr>
<tr>
<td>0.50</td>
<td>2.250</td>
<td>1.750</td>
<td>1.000</td>
<td>9.000</td>
<td>1.5612</td>
<td>15.561</td>
</tr>
<tr>
<td>0.75</td>
<td>3.375</td>
<td>0.875</td>
<td>0.811</td>
<td>7.299</td>
<td>1.5462</td>
<td>13.095</td>
</tr>
<tr>
<td>1.00</td>
<td>4.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.5000</td>
<td>6.000</td>
</tr>
</tbody>
</table>

For different values of $p_1$, Table B.6 is obtained. The coordination $G_c$ of the function, for several values of the probability $p_1$, is shown in Table B.7, and the total activity $G$ in Table B.8.

Table B.8: Total Activity $G$

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$H(x)$</th>
<th>$G_c$</th>
<th>$G_a$</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>2.000</td>
<td>5.000</td>
<td>0.000</td>
<td>7.000</td>
</tr>
<tr>
<td>0.25</td>
<td>2.000</td>
<td>12.595</td>
<td>0.811</td>
<td>15.406</td>
</tr>
<tr>
<td>0.50</td>
<td>2.000</td>
<td>15.561</td>
<td>1.000</td>
<td>18.561</td>
</tr>
<tr>
<td>0.75</td>
<td>2.000</td>
<td>13.095</td>
<td>0.811</td>
<td>15.906</td>
</tr>
<tr>
<td>1.00</td>
<td>2.000</td>
<td>6.000</td>
<td>0.000</td>
<td>8.000</td>
</tr>
</tbody>
</table>