Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author’s right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage. [http://researchspace.auckland.ac.nz/feedback](http://researchspace.auckland.ac.nz/feedback)

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form.
The Development and Application of a Modelling System for Supporting Managerial Decision Making.

Grant Crawford Cowie

A thesis submitted in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY in the Department of Management Studies University of Auckland

To I.C. and F.M.
This thesis studies the potential role of formal models in providing support to the strategic-level decision processes within an organisation. Strategic activities are defined as being essentially outward looking and primarily concerned with the positioning of the organisation in its environment in the long term. The types of interaction which are of interest at this level are defined and a general mathematical framework is developed to describe these processes. Optimisation techniques which are potentially useful in the screening and evaluation of decision alternatives are also discussed.

A collection of mathematical techniques will, however, remain only of theoretical interest. To be of practical assistance in the decision-making process they must be accessible to the decision maker. This can only be accomplished by means of a properly constructed computer system. A brief review of decision support systems, the field of study which is concerned with the application of computers to the decision making process, is followed by an investigation into the requirements that a computer system should fulfil if it is to be successful as a strategic-level decision aid. An overview is then given of "Stratagem", a computer system designed and built to integrate the mathematical modelling framework and the decision support concepts. Finally, two different applications of Stratagem are described to illustrate its potential role in strategic-level decision processes.
ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors, Professor Brian Henshall and Associate-Professor Alistair MacCormick, for their help and guidance throughout this project. A special thanks is also due to Professor Tony Gear, now Head of Department of Business Studies at Trent Polytechnic, England, for first motivating me to pursue postgraduate studies and for his advice throughout the preparation of this thesis. My sincerest thanks also go to Dr Alan Creak of the University Computer Centre for his encouragement and support.

I would also like to thank Feltex Ltd for providing financial support for the first part of this project, the Auckland Savings Bank and Burroughs New Zealand Ltd for providing computer facilities, Rockgas Ltd for permission to use them as a case study, Dr Michael Snowden for permission to use his real estate model, Associate-Professor Martin Putterill for his helpful comments on the draft of this thesis, and Jim Luff of the University Computer Centre for his ready assistance in overcoming many operational problems that arose with the use of the university computer. Special mention must also be made of Sue Lynn for her superb typing efforts.

Finally, I would like to acknowledge the contribution of my parents, without whose continual encouragement and support none of this would have been possible.
# TABLE OF CONTENTS

INTRODUCTION 1

CHAPTER ONE: AN OVERVIEW OF SYSTEMS, ORGANISATIONS AND STRATEGY 3

1.1 Systems and Organisations 3
1.2 An Overview of Management Functions 4
1.3 Strategic Concepts 5
1.4 The Environment 7
1.5 Interaction with the Environment 9
1.6 Strategic Modelling of an Organisation 11
1.7 Summary 11

CHAPTER TWO: TYPES OF DECISION MAKING 12

2.1 The Process of Decision Making 12
2.2 The Structure of Decisions 15
2.3 Synoptic and Incremental Decision Making 16
2.4 Decision Paradigms 18
2.5 Types of Response to Change 20
2.6 Managing the Environment 22
2.7 A Synthesis of Strategy and the Environment 22

CHAPTER THREE: FORMAL MODELS AND THE DECISION PROCESS 24

3.1 The Nature of Models 24
3.2 The Use of Models in Decision Recognition 25
3.3 The Use of Models in Decision Diagnosis 31
3.4 The Use of Models for Screening Alternatives 31
3.5 The Use of Models in the Evaluation-Choice Routine 32
3.6 A Conceptual Framework for Modelling Decisions 33
   3.6.1 Basic Model Components 34
   3.6.2 Environmental Scenarios 35
   3.6.3 Modelling Decisions 35
3.7 Summary 36

CHAPTER FOUR: A MATHEMATICAL FRAMEWORK FOR MODELLING 37

4.1 Model Definition 38
   4.1.1 Types of Models 41
4.2 Scenario Definition 43
   4.2.1 Dynamic Scenarios 45
   4.2.2 Scenario Interdependencies 49
4.3 Decision Definition 52
4.4 Scenarios and Decisions 55
4.5 Solution Techniques 55
   4.5.1 Deterministic Simulation 55
   4.5.2 Simulation for Risk Analysis 56
   4.5.3 Optimisation 58
4.6 Summary 59
# CHAPTER FIVE: OPTIMISATION AND MODELLING

## 5.1 Types of Optimisation

## 5.2 Optimisation Models in Corporate Financial Planning

## 5.3 Simulation and Optimisation in Financial Policy Making

## 5.4 Optimisation Framework: The Gear-Lockett Model

### 5.4.1 Formulation of the Gear-Lockett Model

### 5.4.2 Strengths of the Gear-Lockett Model

### 5.4.3 Short-Comings of the Gear-Lockett Model

## 5.5 Modifications to the Gear-Lockett Model

### 5.5.1 The Problem of Model Size

### 5.5.2 The Problem of Contingency Data Collection

### 5.5.3 The Problem of Output Readability

### 5.5.4 The Problem of Non-Independence of Project Trees

### 5.5.5 The Problem of Structural Alterations

## 5.6 Project Interdependence with the Environment

## 5.7 The Advantages of Response Curves

## 5.8 Risk Analysis

## 5.9 Summary

---

# Appendix to Chapter Five: Implementation Techniques for Optimisation Models

## 5A.1 Structure of Financial Linear Programming Models

## 5A.2 LPGen Language Structure

### 5A.2.1 Identifiers

### 5A.2.2 Variables

### 5A.2.3 Expressions

### 5A.2.4 Statements

### 5A.2.5 Time Periods

### 5A.2.6 Initial Data

### 5A.2.7 Time Relationships

### 5A.2.8 Objective Function

### 5A.2.9 Statement Delimiters

### 5A.2.10 Comments

### 5A.2.11 Options

### 5A.2.12 Output

## 5A.3 Review of LPGen

### 5A.3.1 Structure of Financial Linear Programmes

### 5A.3.2 Relationship between Simulation and Optimisation Models

### 5A.3.3 Structural Relationships Acting as Constraints

### 5A.3.4 Expression Structure

### 5A.3.5 Identifier Structure

### 5A.3.6 Output

### 5A.3.7 Balancing Constraints

---

# CHAPTER SIX: ILLUSTRATIVE EXAMPLE OF FRAMEWORK

## 6.1 Model Background

## 6.2 Structure of Model

### 6.2.1 Organisational Variables

### 6.2.2 Organisational Functions

### 6.2.3 Model Parameters

### 6.2.4 Environmental Variables

### 6.2.5 Environmental Scenarios

### 6.2.6 Interface Relationships

### 6.2.7 Decision Alternatives

### 6.2.8 Policy Limitations

### 6.2.9 Summary of Model
6.3 Solution by Simulation
  6.3.1 An Example of Recursive Substitution

6.4 Solution by Optimisation
  6.4.1 An Example of Recursive Factorisation

6.5 Risk Analysis
  6.5.1 An Example of Risk Analysis

6.6 Review

CHAPTER SEVEN: REQUIREMENTS OF A COMPUTER-BASED MODELLING SYSTEM

7.1 Decision Support Systems

7.2 Technological Requirements of a Decision Support System
  7.2.1 Language System Requirements
    7.2.1.1 Simplicity
    7.2.1.2 Conversational
    7.2.1.3 User Control
  7.2.2 Knowledge System Requirements
    7.2.2.1 Data Availability
    7.2.2.2 Confidentiality
  7.2.3 Problem Processing System Requirements
    7.2.3.1 Flexibility
    7.2.3.2 Feedback
  7.2.4 Decision Support System Environmental Requirements
    7.2.4.1 Response Time
    7.2.4.2 Protection
    7.2.4.3 Accessibility
    7.2.4.4 Economy

7.3 User Requirements

7.4 Requirements of a Specific Decision Support System
  7.4.1 Basic Requirements
  7.4.2 Task Requirements
    7.4.2.1 Decision Recognition Routine
    7.4.2.2 Diagnosis Routine
    7.4.2.3 Evaluation-Choice Routine

7.5 Summary

Appendix to Chapter Seven: A Review of Existing Systems

CHAPTER EIGHT: THE IMPLEMENTATION OF A COMPUTER-BASED MODELLING SYSTEM

8.1 System Overview
  8.1.1 Implementation Philosophy
  8.1.2 System Structure
  8.1.3 Composition of System
  8.1.4 Basic System Components
  8.1.5 Element Declaration
  8.1.6 Element Definitions
  8.1.7 Attribute Interrogation
  8.1.8 Input-Output Monitor

8.2 Decision Support System Implementation
  8.2.1 Language System Requirements
    8.2.1.1 Simplicity
    8.2.1.2 Conversational
    8.2.1.3 User Control
  8.2.2 Knowledge System Requirements
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.2.3</td>
<td>Problem Processing System</td>
<td>153</td>
</tr>
<tr>
<td>8.2.4</td>
<td>Specific DSS Requirements</td>
<td>154</td>
</tr>
<tr>
<td>8.2.4.1</td>
<td>Sets of Scenarios</td>
<td>154</td>
</tr>
<tr>
<td>8.2.4.2</td>
<td>Display Results in Familiar Form</td>
<td>155</td>
</tr>
<tr>
<td>8.2.4.3</td>
<td>Comparison of Results</td>
<td>157</td>
</tr>
<tr>
<td>8.2.4.4</td>
<td>Updating and Saving Models</td>
<td>158</td>
</tr>
<tr>
<td>8.2.4.5</td>
<td>Decision Alternatives</td>
<td>158</td>
</tr>
<tr>
<td>8.2.4.6</td>
<td>Analyse the Effects of Options Simultaneously</td>
<td>158</td>
</tr>
<tr>
<td>8.2.4.7</td>
<td>Sensitivity over Environmental Scenarios</td>
<td>159</td>
</tr>
<tr>
<td>8.2.5</td>
<td>Solution Techniques</td>
<td>160</td>
</tr>
<tr>
<td>8.2.5.1</td>
<td>Deterministic Simulation</td>
<td>160</td>
</tr>
<tr>
<td>8.2.5.2</td>
<td>Optimisation</td>
<td>161</td>
</tr>
<tr>
<td>8.2.5.3</td>
<td>Risk Analysis</td>
<td>162</td>
</tr>
<tr>
<td>8.2.5.4</td>
<td>Simultaneous Equations</td>
<td>163</td>
</tr>
<tr>
<td>8.3</td>
<td>Summary</td>
<td>163</td>
</tr>
</tbody>
</table>

CHAPTER NINE: TWO APPLICATIONS

9.1 Application One: Rockgas Limited
9.1.1 Organisational Structure | 166  |
9.1.2 Proposed Investment | 166  |
9.1.3 Stratagem Model | 168  |
9.1.4 Model Results | 175  |
9.1.5 Ongoing Modelling | 180  |
9.1.6 Evaluation | 182  |

9.2 Application Two: A Real Estate Model
9.2.1 Model Background | 184  |
9.2.2 Original Implementation | 185  |
9.2.3 Stratagem Implementation | 187  |
9.2.4 Summary of Stratagem Runs
9.2.4.1 Base Case | 189  |
9.2.4.2 Low Inflation | 191  |
9.2.4.3 Post Horizon Effects (1) | 191  |
9.2.4.4 Post Horizon Effects (2) | 192  |
9.2.4.5 Lower Real Rate of Return | 193  |

9.2.5 Review of Stratagem Implementation
9.2.5.1 Model Specification | 193  |
9.2.5.2 Model Outputs | 194  |
9.2.5.3 Manual Errors | 194  |
9.2.5.4 Model Analysis | 195  |
9.2.5.5 Interactive | 195  |
9.2.5.6 Disadvantages | 195  |

9.3 Summary and Conclusions | 196  |

CHAPTER TEN: SUMMARY AND CONCLUSIONS

10.1 Strategic Management | 198  |
10.2 Management Science | 198  |
10.3 Decision Support Systems | 199  |
10.4 Computer Science | 200  |
10.5 Stratagem | 200  |
APPENDIX A: COMPUTER IMPLEMENTATION ASPECTS OF STRATAGEM

A1  System Environment  202
A2  System Structure  203
A2.1  Model Spread Sheet  206
A2.2  Symbol Table  207
A2.3  Code Storage  208
A2.4  Symbol Storage  210
A2.5  Variable Environment  210
A2.6  Structure Storage  212
A2.7  Report Storage  214
A3  Structure of the Statement Virtual Machine  214
A3.1  Virtual Machine Instruction Set  215
A3.2  Loading the Value  219
A3.3  Comment on Virtual Machine Instruction Set  220
A4  Factorisation Virtual Machine  221
A4.1  Structure of Factorised Matrix  222
A4.2  Multiple Pass Recursive Factorisation  223
A4.3  Discussion of Multiple Pass Recursive Factorisation  225
A4.4  Single Pass Factorisation  227
A4.4.1  Stack Structure  228
A4.4.2  Revised Operations  228
A4.4.3  Discussion of Single Pass Factorisation  233

APPENDIX B: STRATAGEM REFERENCE MANUAL  234

B1  Concepts  249
B2  Syntax Diagrams  255
B3  System Input  257
B4  Declarations  275
B5  Definitions  285
B6  Function Primary  307
B7  Interrogation  314
B8  Reports  316
B9  Decisions  333
B10  Scenarios  340
B11  Augmented Definitions  348
B12  Statements  355
B13  Reserved Words  377

Stratagem Reference Manual Index  379

APPENDIX C: A STRATAGEM EXAMPLE  383

BIBLIOGRAPHY  412
## TABLE OF FIGURES

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A Model of the Decision Process</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Types of Response</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Use of a Model in Decision Recognition</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>The Solution Matrix</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>The Partitioned Solution Matrix</td>
<td>41</td>
</tr>
<tr>
<td>4.3</td>
<td>Types of Interaction</td>
<td>44</td>
</tr>
<tr>
<td>4.4</td>
<td>Composite Scenario for Market Share and Inflation</td>
<td>46</td>
</tr>
<tr>
<td>4.5</td>
<td>Two Dynamic Scenarios for Market Share and Inflation</td>
<td>46</td>
</tr>
<tr>
<td>4.6</td>
<td>Possible Combinations of Scenarios</td>
<td>48</td>
</tr>
<tr>
<td>4.7</td>
<td>Inflation and Interest Rate Scenarios</td>
<td>50</td>
</tr>
<tr>
<td>4.8</td>
<td>Combined Scenarios</td>
<td>50</td>
</tr>
<tr>
<td>4.9</td>
<td>Conditional Scenarios</td>
<td>51</td>
</tr>
<tr>
<td>4.10</td>
<td>Sample Decision to Build Aluminium Smelter</td>
<td>54</td>
</tr>
<tr>
<td>4.11</td>
<td>Deterministic Simulation by Recursive Substitution</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>An Example of the Gear-Lockett Model</td>
<td>64</td>
</tr>
<tr>
<td>5.2</td>
<td>Formulation of the Example in Figure 5.1 into a Zero-One Programme</td>
<td>66</td>
</tr>
<tr>
<td>5.3</td>
<td>Linear Response Curve as Used in Gear-Cowie Model</td>
<td>73</td>
</tr>
<tr>
<td>5.4</td>
<td>Non-Linear, Negatively Correlated Response Curve</td>
<td>73</td>
</tr>
<tr>
<td>5.5</td>
<td>Estimated Point Impacts of Indicator Changes</td>
<td>73</td>
</tr>
<tr>
<td>5.6</td>
<td>Interpolation of Intermediate Impacts</td>
<td>73</td>
</tr>
<tr>
<td>5.7</td>
<td>Linear Interpolation of Intermediate Points</td>
<td>75</td>
</tr>
<tr>
<td>5A1</td>
<td>LPGen Example</td>
<td>86</td>
</tr>
<tr>
<td>6.1</td>
<td>Environmental Scenarios</td>
<td>96</td>
</tr>
<tr>
<td>6.2</td>
<td>Response Curves</td>
<td>98</td>
</tr>
<tr>
<td>6.3</td>
<td>Possible Strategies and their Impact</td>
<td>100</td>
</tr>
<tr>
<td>6.4</td>
<td>Recursive Substitution Sequence for Interest Calculation</td>
<td>104</td>
</tr>
<tr>
<td>6.5</td>
<td>Linear Programme Tableau Generated by Recursive Factorisation</td>
<td>107</td>
</tr>
<tr>
<td>6.6</td>
<td>Summary of Risk Analysis Results</td>
<td>109</td>
</tr>
<tr>
<td>7.1</td>
<td>Taxonomy of Decision Support Systems</td>
<td>112</td>
</tr>
<tr>
<td>7.2</td>
<td>Components and Users of DSS</td>
<td>115</td>
</tr>
<tr>
<td>7.3</td>
<td>DSS Types, Technologies and Users</td>
<td>117</td>
</tr>
<tr>
<td>7.4</td>
<td>Technological Requirements of a Decision Support System</td>
<td>118</td>
</tr>
<tr>
<td>7.5</td>
<td>Summary of Requirements for DSS</td>
<td>130</td>
</tr>
<tr>
<td>8.1</td>
<td>Structure of Stratagem</td>
<td>140</td>
</tr>
<tr>
<td>8.2</td>
<td>Basic System Structure</td>
<td>142</td>
</tr>
<tr>
<td>8.3</td>
<td>Range of, and Mechanism for Determining, the Attribute &quot;Value&quot;</td>
<td>145</td>
</tr>
<tr>
<td>8.4</td>
<td>The Structure and Function of Definitions</td>
<td>148</td>
</tr>
<tr>
<td>8.5</td>
<td>Example of a Stratagem Scenario</td>
<td>156</td>
</tr>
<tr>
<td>8.6</td>
<td>Comparison of Simplan and Stratagem Approaches to Solving a Simultaneous Model</td>
<td>164</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>9.1</td>
<td>Rockgas Implementation Options</td>
<td>169</td>
</tr>
<tr>
<td>9.2</td>
<td>Summary of Model Construction</td>
<td>171</td>
</tr>
<tr>
<td>9.3</td>
<td>Structure of Rockgas Model</td>
<td>172</td>
</tr>
<tr>
<td>9.4</td>
<td>Outline of Rockgas Reports</td>
<td>176</td>
</tr>
<tr>
<td>9.5</td>
<td>Alterations Required to Switch Futures in Real Estate LPGen Model</td>
<td>188</td>
</tr>
<tr>
<td>9.6</td>
<td>Summary of Applications</td>
<td>197</td>
</tr>
<tr>
<td>A1</td>
<td>Stratagem Data Structures</td>
<td>205</td>
</tr>
<tr>
<td>A2</td>
<td>Structure of Variable Environment</td>
<td>211</td>
</tr>
<tr>
<td>A3</td>
<td>Structure of Structure Storage</td>
<td>213</td>
</tr>
<tr>
<td>A4</td>
<td>Multiple Pass Arithmetic Operators</td>
<td>224</td>
</tr>
<tr>
<td>A5</td>
<td>Multiple Pass Factorisation Sequence</td>
<td>226</td>
</tr>
<tr>
<td>A6</td>
<td>Single Pass Stack Structure</td>
<td>229</td>
</tr>
<tr>
<td>A7</td>
<td>Single Pass Factorisation Sequence</td>
<td>232</td>
</tr>
<tr>
<td>Section</td>
<td>Table Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>7.1</td>
<td>Requirements of an On-Line, Time Shared</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>Computer System</td>
<td></td>
</tr>
<tr>
<td>7.2</td>
<td>&quot;Stanley House Criteria&quot;</td>
<td>119</td>
</tr>
<tr>
<td>7A1</td>
<td>Systems and Suppliers</td>
<td>135</td>
</tr>
<tr>
<td>7A2</td>
<td>Model Statements</td>
<td>136</td>
</tr>
</tbody>
</table>
Introduction.

This research is aimed at providing a modelling aid for strategic decision making. The motivation for the research was the observation that there is a "technological gap" between modelling techniques used in practice and those advocated in the academic literature. The latter tend to be complex models based on optimisation or stochastic techniques, whereas models in practice (when used at all) tend to rely on simple deterministic simulation. The primary objective of this research was to narrow this gap.

The two main reasons for the existence of the gap are:

1) there is a lag between academic theory and management practice. Only now is the impact of management science education being felt at the senior executive level. Previously top management was aware only vaguely of the possible benefits of modelling techniques. However now that the first wave of business-school trained managers is reaching this level, it is possible that more advanced techniques will make a direct impact;

2) the techniques themselves, although known, are perceived to be irrelevant because they are largely inaccessible to the non-specialist. Even if top management wanted to use an optimisation model of a strategic-type decision, it would not be possible as the facilities for doing this in a quick and easy manner just do not exist.

It is to this second problem of "technique availability" that this research is directed.
The research methodology is as follows:

1) identify the characteristics of the strategic decision making problem;

2) identify possible types of decision, and the decision making process itself;

3) study the possible applications of formal modelling to the components of the decision making process, with particular emphasis on strategic decision making;

4) synthesise a modelling framework which incorporates the modelling applications into the decision making process;

5) make the modelling framework available to the non-specialist user by implementing it as an interactive computer system.

The research draws heavily from the areas of strategic management, management science, decision support systems and computer science, and for this reason cannot be said to fit neatly into any one of these categories. Instead, it attempts to integrate the relevant aspects of the disciplines in such a manner as to narrow the "technological gap".

The first four chapters of this thesis correspond to the first four steps of the methodology, as described above. Chapter five investigates the structure of optimisation models and presents an integrated framework for optimisation and simulation models. Chapter six gives a small hypothetical example which illustrates the framework. The requirements for a computer system which can implement the framework are given in chapter seven, and a brief overview of Stratagem, the computer-based modelling system constructed around the framework, is given in chapter eight. Two applications of Stratagem are discussed in chapter nine. The first application illustrates its use in an actual strategic-decision situation. In the second application, Stratagem is used to analyse a (hypothetical) optimisation model of corporate real estate ownership.

Summary and conclusions are given in chapter ten.
Chapter One:
An Overview of Systems, Organisations and Strategy.

An organisation can be viewed as an open system, exchanging inputs and outputs with its environment. This chapter studies the managerial activities within the organisational system, particularly those that are concerned with the environment and the posturing of the organisation within that environment. These activities are termed strategic.

Various types of different environment and environmental interaction are studied. It is shown that environmental events which in themselves would have little or no effect on an organisation can react together to produce other events which do have a significant impact upon the organisation. Because of such complexities, it is necessary to make simplifications when attempting to understand the organisation and its environment.

1.1 SYSTEMS AND ORGANISATIONS

A system can be defined as a group of often diverse parts arranged to fulfil some common purpose. All items which do not form part of the system constitute its environment. Each component of the system is itself a system of components (Klir and Valach (1967), Bertalanffy (1968)).

A system is said to be open if it exchanges matter with its environment, i.e. has inputs and outputs (von Bertalanffy). Apart from in the idealised world of physics and chemistry, there is no such thing as a closed system. Neither does the environment exist independently of the system, as it will be altered or influenced to some extent by the inputs and outputs of the system (Boulding (1978); p.31).
Parsons (1961) defines an organisation to be:-

"... a system which, as the attainment of its goal, 'produces' an identifiable something which can be utilised in some way by another system; that is the output of the organisation is, for some other system, an input. In the case of an organisation with economic primacy, this output may be a class of goods or services which are either consumable or serve as instruments for a further phase of the production process by other organisations."

Beer (1959; p.16) has categorised commercial organisations as "exceedingly complex probabilistic systems", i.e. it does not seem reasonable to imagine that such a system will ever be fully described.

1.2 AN OVERVIEW OF MANAGEMENT FUNCTIONS

The activities that management undertake within an organisation may be grouped into three broad categories (Ansoff (1965), Anthony (1965)).

Strategic: Those activities which occur at the highest level of the organisation and involve important decisions which affect the way in which the organisation operates with respect to its environment.

Administration: The process by which the organisation is structured internally, and the process by which resources are obtained and used to best achieve the organisation's goals.

Operational: The process by which specific tasks (such as production, marketing, credit control) are carried out.

These categories are by no means distinct and separate, but rather they tend to overlap. Some situations do not fit neatly into one particular category, and others may change categories depending upon time and circumstances. Nonetheless the framework, if not applied rigidly, is useful. This study will be primarily concerned with some aspects of the strategic function.
1.3 STRATEGIC CONCEPTS

In the context of business management two principal dimensions have been identified with the concept of strategy. Anthony (1965; p.16) for example, states that:-

"It [the word "strategy"] connotes big plans, important plans, plans with major consequences."

The second dimension is that the concept of "strategic" is involved with looking outwards away from the organisation towards its environment (as distinct from operational considerations which are inwards looking). Ansoff (1965; p.18) for example, states that:-

"Here we use the term strategic to mean 'pertaining to the relation between the firm and its environment'."

This is not to imply that strategy is concerned only with the external environment at the expense of the internal processes within an organisation. Strategic considerations will affect the internal culture and structure of the organisation, and vice versa (Ansoff (1979), Chandler (1962)).

This study will concentrate on the outwards looking dimension of strategy, which is one aspect of the field of activities called Strategic Management (SM). SM is the continual monitoring and control of the organisation's strategic perspective. It is the process by which management attempts to ensure that the culture and configuration of the organisation matches the environment (Ansoff (1979)). It is an ongoing activity that scans outwardly for signs of shifts or discontinuities in the environment, and inwardly for possible mismatches in the structure, services and paradigm of the organisation.

One aspect of SM is strategic planning. Planning is a fundamental function of all goal seekers, whether they be individuals, governments, commercial enterprises or whatever. It is a process that attempts to produce one or more desired future states according to some value system when the future states are not likely to occur unless some action is taken (Ackoff (1970), Mason (1969)).

Planning is done before specific actions are made. It is characteristic
of entities that plan to make predictions about the future state of the world and to believe that they have a choice which will secure a desired future state (Mason). Ackoff (1970; p.1) states that:-

"The need for corporate planning is so obvious and so great that it is hard for anyone to be against it."

He suggests that strategic planning is differentiated from other (tactical) planning by:-

1. being long-term and hard to reverse.
2. wider in scope.
3. an emphasis on the formulation of goals, and
   a means by which they are to be achieved.

This view is shared by McKinney (1969; p.6) who defines strategic planning as:-

"...the process of generating a composite corporate strategy, where a composite strategy includes not only a comprehensive statement of the long-term goals of the enterprise, but also an explicit statement of the actions required to attain those goals and the timing of these actions."

Both Ackoff's and McKinney's definitions follow that of Anthony, focussing upon the "importance" dimension of strategy and apparently ignoring the other dimension of "outward looking". An awareness of this has grown with the evolution of SM (see Ansoff (1973)). Radford (1980; p.ix) for example, states that:-

"... the aim of strategic planning is to ensure that the present and future activities of an organisation are appropriately matched to conditions in the environment in which it must operate."

This definition, implying as it does that control is an integral part of the planning function, is perhaps too broad in scope, and could better be applied to what is generally regarded as SM. It does however reflect the awareness of the importance of the environment to the strategic planning process. Drucker (1980; p.61) succinctly differentiates between strategy and planning when he states:-

"Planning tries to optimise tomorrow the trends of today.
Strategy aims to exploit the new and different opportunities of tomorrow."
1.4 THE ENVIRONMENT

As already stated, the organisational environment and how the organisation relates to it is of critical concern to the strategic activities of the organisation. Ansoff (1979) for example states that the strategic behaviour of an organisation is "the process of interaction with the environment, accompanied by a process of changing internal configurations and dynamics". He also hypothesises that "an organisation will be successful if environment, response, culture, and capability match each other".

Radford (1977; p.21) groups the elements that comprise an environment into three broad categories:—

1. social: those matters concerned with the mutual relations between the participants;
2. artificial: elements (such as economic, engineering, etc.) that have a bearing on the technological aspects;
3. natural events: acts of God or events caused by agents not directly involved.

Mintzberg (1979) has identified four dimensions of organisational environments:—

1. stability: an environment may range from stable to dynamic. For example, instability can be caused by an unstable government or the unexpected actions of a competitor.
2. complexity: if an environment requires that the organisation have a great deal of sophisticated knowledge about it, then the environment is complex. If the information can be rationalised then the environment becomes simple.
3. market diversity: the environment is diversified to the extent that there is a range of clients, products, geographical locations etc. The environment is integrated to the extent that these are homologous.
4. hostility: the extent to which an environment is hostile (or munificent) is determined by factors such as competition, labour and government relations, resource availability etc.

Emergy and Trist (1965) have identified four different types of environment:

- placid, randomised - "in which goals and noxiant ('goods' and 'bads') are relatively unchanging in themselves, and randomly distributed" (i.e. there are no connections between the components of the environment). "A critical property from the organisation's viewpoint is that there is no difference between tactics and strategy and organisations can exist adaptively as single, and indeed quite small, units." This is the type of environment assumed under the economic theory of perfect competition.

- placid, clustered - goals and noxiant are loosely connected, and not randomly distributed. Some positions in the environment are potentially richer than others, and as such strategy (as opposed to tactics) must be employed if the organisation is to survive. This is the type of environment assumed under the economic theory of "imperfect competition".

- disturbed, reactive - similar to the placid, clustered environment except with the dominant characteristic of similar organisations existing in the environment. Thus each organisation must be aware of the likely activities of the others, such as in an oligopolistic market.

- turbulent fields - dynamic processes which affect the component organisations arise, not only from the interactions between organisations (as in
disturbed-reactive) but also from the environment itself.

These types of environment do not represent individual, discrete categories, but are placements on a continuum.

There is a trend forcing mature organisations to operate towards the turbulent end of this continuum. The nature of this environment is such that (cf. Post (1978; p.17));

1. managers tend to emphasise the importance of keeping the organisation on course (according to perceived mission and strategy);
2. attention is given to bringing more stability into the environment (e.g. forming trade associations);
3. a premium is placed on the organisation's ability to respond to change.

1.5 INTERACTION WITH THE ENVIRONMENT

Mintzberg (1979; p.296) states:--

"... it is not the environment per se that counts but the organisation's ability to cope with it - to predict it, comprehend it, deal with its diversity and respond quickly to it."

A commercial organisation has been previously defined to be a complex open system operating in a turbulent environment.

Emery and Trist (1965) propose the following:--

"That a comprehensive understanding of organisational behaviour requires some knowledge of each member of the following set, where L indicates some potentially lawful connection, and the suffix 1 refers to the organisation and the suffix 2 to the environment:--

\[
\begin{align*}
L_{11} & \quad L_{12} \\
L_{21} & \quad L_{22}
\end{align*}
\]

\(L_{11}\) here refers to processes within the organisation - the area of internal interdependencies; \(L_{12}\) and \(L_{21}\)
to exchanges between the organisation and its environment - the area of transactional interdependencies, from either direction; and \( L_{22} \) to processes through which parts of the environment become related to each other - i.e. its causal texture - the area of interdependencies within the organisation itself."

Terreberry (1968) uses these symbols to denote the relationship between organisation and environment as follows:-

\[
\begin{align*}
L_{11} & \quad \text{- intra-system interdependencies} \\
L_{21} & \quad \text{- input} \\
L_{12} & \quad \text{- output} \\
L_{22} & \quad \text{- extra-system interdependencies}
\end{align*}
\]

These relate generally to Ansoff's (1979) categorisation of company behaviour as:-

\[
\begin{align*}
L_{21} \quad & \text{strategic behaviour: "the process of} \\
L_{12} \quad & \text{interaction with the environment ...} \\
L_{11} \quad & \text{... accompanied by a process of changing} \\
\quad & \text{internal configurations and dynamics".} \\
L_{11} \quad & \text{operating behaviour: "... concerned with} \\
\quad & \text{the internal resource conversion process} \\
\quad & \text{in an organisation."}
\end{align*}
\]

(A similar parallel can be drawn with Anthony's framework).

The concept of the causal texture of the environment (\( L_{22} \)) is not included in either of the categorisations, even though it is of crucial importance in understanding the effects of a turbulent environment. The example of the oil price rise of 1973-1974 is particularly salient. Simplified, there were two events:-

- Event 1: Arab-Israeli war
- Event 2: U.S. supports Israel

Most organisations would not be directly affected by either of these two events (i.e. the \( L_{12} \) and \( L_{21} \) between most organisations and this particular segment of the environment exist only weakly, if at all). Yet the result of the interaction of these two events (\( L_{22} \)) was an oil embargo which directly affected the \( L_{21} \) of most organisations.
1.6 STRATEGIC MODELLING OF AN ORGANISATION

The concept of an organisation as a complex, open system has certain ramifications to both strategic management and modelling. The first and most obvious of these is that, as the organisation is too complex to be fully described (Beer (1959)), it will be necessary to make simplifications and generalisations when attempting to understand any organisation. In translating this understanding into a formal model, it will be necessary to make further simplifying assumptions.

The second point is that as "strategy", in part, connotes "outward looking towards the environment", a model at the strategic level must include:

1. important environmental aspects and (if possible) the interactions between them (\(L_{22}\) effects);
2. critical components of the environmental interface with the organisation, comprising the inputs (\(L_{21}\)) and the outputs (\(L_{12}\));
3. relevant aspects of the resource conversion process within the organisation (\(L_{11}\)).

1.7 SUMMARY

This chapter has discussed the concept of an organisation operating as a complex open system in a turbulent environment. The strategic or outward looking component of the organisation must be aware of the important characteristics of, and interactions within, this environment. A further strategic function is an understanding of the organisation's environmental interface, which is the mechanism by which these effects are transmitted to the organisation. The complexities of the processes involved however mean that any understanding can only be based on a simplified view of the organisational system and its environment.

The next chapter looks at types of decision processes within an organisation, and how these attempt to cope with environmental uncertainty and change.
Chapter Two: Types of Decision Making.

The previous chapter has discussed the organisation as a complex system in a turbulent environment. It is the function of the strategic component of the organisation to "manage" this environment. This chapter is concerned with the various types of decision-making that are used in this process.

2.1 THE PROCESS OF DECISION-MAKING

Simon (1960) has recognised three phases of the decision-making process. He describes these as:

1. intelligence activity - searching the environment for conditions calling for a decision;
2. design activity - inventing, developing and analysing possible courses of action;
3. choice activity - selecting a particular course of action from those available.

Mintzberg et al (1976) have broken these phases down even further (see figure 2.1). Their identification phase (Simon's intelligence activity) is subdivided into decision recognition and diagnosis. A decision is recognised as being required when there is found to be a difference between an actual and an expected (more desired) situation. Diagnosis takes place when information is sought that will better define the problem.

The development phase (Simon's design activity) has been divided by Mintzberg et al into two separate activities, search and design. The former of these is involved with seeking ready-made solutions to the problem, whilst design is used to develop custom-made solutions or to
Figure 2.1: A Model of the Decision Process.

(From Mintzberg et al., 1976).
modify ready-made ones. Four types of search behaviour have been identified:-

1. Memory search, in which the organisation's existing memory (human or recorded) is scanned;
2. Passive search, in which the organisation waits for unsolicited alternatives to appear;
3. Trap search, in which 'search generators' are activated (such as, for example, letting suppliers know that the firm is looking for certain equipment);
4. Active search, where direct alternatives are sought.

The selection phase (Simon's choice activity) is a multistage, iterative process which Mintzberg et al have decomposed into three routines, screening, evaluation-choice, and authorisation. The purpose of the screen routine is to eliminate the infeasible, whereas that of the evaluation-choice routine is to make a selection. Three modes of evaluation have been identified by Mintzberg:-

1. judgment, in which one individual makes the choice in his own mind;
2. bargaining, in which a group of conflicting decision-makers, each exercising his own judgment, makes the choice by bargaining with each other;
3. analysis, in which a factual evaluation is made, usually by technocrats, followed by a choice, either by judgment or bargaining.

According to Mintzberg the judgment mode is used most, and the analytical least. Authorisation is needed when the person making the choice does not have the authority to commit the organisation to that course.

Mintzberg makes the point that the decision process does not follow a simple sequence through the identification, development and selection phases. Instead, at any given point it may return to an earlier phase, and thus is cyclic and iterative, and not necessarily sequential.
A further stage in the decision process has been identified in cybernetics (see, for example, Felsen (1973)). This is the learning phase, which involves recording the decision process, alternatives, and outcomes into the organisation's memory.

2.2 THE STRUCTURE OF DECISIONS

Simon (1960) has differentiated between two polar types of decision, programmed and non-programmed. He states (pp.5-6):-

"Decisions are programmed to the extent that they are repetitive and routine, to the extent that a definite procedure has been worked out for handling them so that they don't have to be treated de novo each time they occur ... Decisions are non-programmed to the extent that they are novel, unstructured and consequential. There is no cut-and-dried method for handling the problem because it hasn't arisen before, or because its precise nature and structure are elusive or complex, or because it is so important that it deserves a custom-tailored treatment".

These are not distinct categories of decision, but are the extremes of a continuum. At one extreme are the decisions that can be made entirely by machines (for example, a thermostat or a computer), whereas at the other are decisions that exhibit little or no apparent structure (such as the timing of Eisenhower's D-Day decision). The degree to which a problem is well-structured (programmed) can be gauged by (amongst other criteria) the existence of "a definite criterion for testing any proposed solution, and a mechanisable process for applying the criterion" (Simon (1973)). Clearly the scope, given our present understanding, for applying a mechanised procedure, or even formulating a definite criterion, is limited at the strategic level of organisations.

Mintzberg et al (1976) suggest that the decision-making process for both structured and ill-structured (non-programmed) decision-making is similar in that they both will generally follow the identification-development-selection process. A truly structured decision will, however, go through the process only once. The less structured the decision, the more iterations and cycles will be needed.
2.3 SYNOPTIC AND INCREMENTAL DECISION-MAKING

Lindblom (1965) has identified two extreme approaches to the design-choice aspects of the decision-making process. He calls the first of these synoptic "because of the high degree of synopsis or comprehensiveness of view the decision-maker seeks to achieve in such a method" (Lindblom; p.137). Basically this approach consists of:-

1. identifying and ordering all relevant objectives;
2. identifying all possible means of achieving these, and the probable consequences of each of the means;
3. selecting a means that best fulfils the objectives.

The other approach, termed "incremental", consists of:-

1. analysis of increments only;
2. consideration of a restricted number of policy alternatives, and only the important consequences of any given possible policy alternative;
3. a serial analysis and evaluation.

He suggests that the synoptic approach is impossible because "it assumes intellectual capacities and sources of information that men do not have." (Lindblom (1959)). This is supported by Simon's principle of bounded rationality which states (Simon (1957; p.198)):-

"The capacity of the human mind for formulating and solving complex problems is very small compared with the size of the problems whose solution is required for objectively rational behaviour in the real world - or even for a reasonable approximation to such objective rationality."

Lindblom (1959) also suggests that the synoptic approach is inappropriate because of the time and expense required for a complete analysis.

However, a major disadvantage of the incremental approach is the possibility that the decision-maker will overlook better policies simply because the successive steps of the incremental procedure do not suggest them.
These conclusions are supported by Wheelwright (1970), who conducted experiments into the effects on the planners of synoptic and incremental strategic planning processes. He found that:

1. the synoptic procedure led to the consideration of a much wider range of alternatives;
2. the incremental procedure resulted in better agreement among the planners on the strategy and objectives, and also to more commitment to the strategy. There was a lack of agreement on the objectives derived from the synoptic approach and they tended to be more general;
3. the incremental procedure generally led to better strategies than did the synoptic (as evaluated by both judges and planners).

Etzioni (1968) has suggested a combination of the synoptic and incremental approaches that overcomes the shortcomings of both. The approach, called "mixed scanning" involves differentiating between contextuating (fundamental) decisions and bit (or item) decisions. He states (p.283):

"Contextuating decisions are made through an exploration of the main alternatives seen by the actor in view of his conception of his goals, but unlike what [the synoptic approach] would indicate - details and specifications are omitted so that overviews are feasible. ...But incrementalism overcomes the unrealistic aspects of the synoptic approach (by limiting it to contextuating decisions), and contextuating rationalism helps to right the conservative bias of incrementalism."

Thus, according to Etzioni, certain key decisions (contextuating or strategic) should be made starting with a simplified synoptic approach, but continuing incrementally, with periodic synoptic reviews.

In the context of the decision process described by Mintzberg et al, the mixed scanning approach suggests that the development phase and the screen routine should have more emphasis placed upon them (i.e. a wider search routine and more rigorous screening), but the evaluation phase should be weakened. This isolates the broad options
that are available. Working with these, successive refinements are made (with periodic reviews) in an incremental manner through the development and selection phases.

2.4 DECISION PARADIGMS

Simon (1957; p.198) states that a consequence of the principle of bounded rationality (cf section 2.3) is that:

"... the intended rationality of an actor requires him to construct a simplified model of the real situation in order to deal with it. He behaves rationally with respect to this model, and such behaviour is not even approximately optimal with respect to the real world."

The need to have a simplified model results in perception difficulties by the decision-maker with respect to his environment. The acquisition of information about the external world involves a continual interaction between the incoming raw data and previously assimilated problem solving formulations. Simon (1973) suggests that this interaction is a limitation upon the understanding process...

"... because it tends to mould all new information to the paradigms that are already available. The problem solver never perceives the Ding an Sich, but only the external stimulus filtered through its own perceptions. Hence the acquisition process exercises a strong influence in the direction of conserving existing problem formulations. The world as perceived is better structured than the raw world outside."

This, of course, has major ramifications upon the decision-making and problem-solving processes. The first step in these processes is to recognise that there is a need for a decision, or that a problem exists. The fact that there is such a need can be obscured by the actor's perception of his environment. Post (1978; p.15) identifies an actual situation (which he terms the "objective environment") and a believed situation (the "perceived environment"). He recognises that there can be a considerable difference between these two, and argues that it is the differences in the perceived environment that is one cause for similar organisations reacting differently to the same set of external stimuli (Post; p.233).
The scope of these paradigms is clearly not restricted to extra-organisational environments. The manager is a component of a system (the organisation), which is in turn a component of a larger system (the organisational environment). Thus the manager will not only have a model of the organisational environment, but also one of the organisation itself. Just as it is one of the functions of management accounting systems to ensure that the manager's model of the organisation closely resembles the actual organisation, so it is a function of strategic management to ensure that the organisation's model of the environment resembles the actual environment.

Mason (1969) has suggested several techniques that can be used to attempt to match the subjective with the objective environment:

1. the expert planning approach, in which it is the task of the expert to study the environment. Unfortunately, as Mason points out, "the expert advisor, too, has a particular way of viewing the organisation";

2. the devil's advocate approach, in which management, acting as an adversary, attempts to find fault with any plan. However this process tends to be destructive instead of constructive;

3. the dialectical approach, in which a plan is opposed by a counterplan, which is a credible alternative. By observing the conflict between plan and counter-plan, management learns the assumptions inherent in the planning process, and may synthesise a new and expanded world view.

Just as individuals have objective and perceived environments, so do organisations (Bourgeois (1980), Huxham and Dando (1981), Mitroff and Emshoff (1979)). Quinn (1977) states, for example:

"Consequently, effective executives ... consciously seek multiple contact points with managers, workers, customers, suppliers, technologists, outside professional and government groups, and so on. They purposely short-
circuit all the careful screens an organisation builds to 'tell the top only what it wants to hear' and thus delay important strategic signals."

2.5 TYPES OF RESPONSE TO CHANGE

Ansoff (1968) recognises two types of response to a change in the environment. The first of these is the reactive lag response, where the organisation recognises the need for action when the event that has caused the need has already taken place. The opposite kind of response is the anticipatory response, in which the organisation foresees the requirement for change.

Post (1978) has identified a further type of response, in which the organisation acts to promote desired change in the environment and to eliminate potential threats before they occur. Characteristic of the proactive (and anticipatory) response is a management that forecasts or foresees the future, and makes plans accordingly. Ackoff (1977) terms planners who operate in an anticipatory mode prospective, and those who operate proactively as introspective.

Prospective plans are concerned with modifying the organisation's goals, strategies and procedures in accordance with the changing environment. Introspective plans attempt to modify the environment so as to be more amenable to the organisation's goals. Thus introspective plans and the proactive response is indicative of a synoptic viewpoint whereas the reactive response suggests an incremental approach.

The reactive/proactive continuum closely parallels four general attitudes towards planning identified by Ackoff (1974).

1. inactivists - believe that any intervention in the course of events will only make things worse;
2. reactivists - believe that things are getting worse, and try to return to a previous state;
3. preactivists - believe that the future will be better provided they are ready for it.
4. interactivists - believe that they can shape the future, as well as its affects.
He suggests that each approach is applicable in different situations; for example, if things are moving of their own accord in a desirable manner, then inactivism is probably appropriate.

A further dimension of response type has been identified by Schumpeter (1947):

- adaptive response - reacts within scope of existing practice;
- creative response - reacts outside scope of existing practice.

A creative response is discontinuous with respect to pre-existing facts, so that it cannot be predicted by inference from these facts. Furthermore there are no links with the conditions that it creates and those that would have existed in its absence. Schumpeter equates the creative response with entrepreneurship.

The types of response are summarised in figure 2.2.

![Figure 2.2: Types of Response.](image-url)
2.6 MANAGING THE ENVIRONMENT

A common thread running through many of the definitions of strategy is that strategy has two main purposes (see, for example, Bourgeois (1980)):—

1. the definition of the segment of the environment in which the organisation will operate (domain definition);

2. providing guidance within a particular product-market or task environment for subsequent goal directed activity (domain navigation).

Within this broad hierarchy there are several basic approaches for an organisation to manage its dependencies on the external environment (Kotter (1979)). These can be characterised as:—

1. domain selection
   - search for environments where external dependency is easy to manage (e.g. where market dominance can be achieved)
   - expand the domain through diversification

2. establish external linkages
   - advertising and public relations work to engender favourable attitudes towards the organisation
   - establish personal linkages (e.g. through directorships)
   - establish coalitions with other organisations

3. control who operates in the environment
   - force out competition
   - create associations to control competition
   - influence legislation to limit competition

4. organisational design
   - create subunits within the organisation to deal with specific facets of the environment.

2.7 A SYNTHESIS OF STRATEGY AND THE ENVIRONMENT

Previously in this study (section 1.3) strategic management was
broadly described as the process by which management attempts to ensure that the culture and configuration of the organisation matches the environment. Thus, one part of the strategic management "process" is a continual monitoring of the environment and its interface with the organisation.

The size, complexity and dynamic nature of the external environment is such, however, that it is beyond complete human comprehension. Thus, it is necessary for managers and organisations to construct simplified models of it to aid understanding. Such models are usually implicit and unstated, but are nonetheless important as they constitute the perceived environment and not the actual or objective environment.

Two principle levels of strategy formulation can be identified. The first of these involves scanning the general environment both for trends that may affect the environment and for suitable new task (e.g. product-market) environments (domain selection or primary strategy). The second level deals specifically with the task environment(s) and any significant changes in it (domain navigation or secondary strategy).

The result of these scans will be the realisation at some point in time that a "strategic decision" needs to be made (such as diversification or divestment). These may be reactive to events in the environment, anticipative of events in the environment, proactive on the environment itself, or introspective on the organisation. The processes involved in making such decisions conform to a general structure, starting with the recognition of the problem and ending with the implementation of the decision.
Chapter Three:
Formal Models and
The Decision Process.

In the previous chapters a review was given of the concept of organisational strategy and of the structure of decisions. This chapter develops a conceptual framework for the use of formal models in the organisational decision process. The first section presents a brief review of the nature of models. In subsequent sections the use of models in the various phases of the decision making process is investigated. From this, a modelling framework is developed to support the decision process.

3.1 THE NATURE OF MODELS

There are three categories of model (Howard (1971)):

1. physical, containing those models which look like the subject but do not act like it (iconic), and those that act like the subject but do not look like it (analogue);

2. mental, which are models in the abstract;

3. symbolic, comprising graphical, written and mathematical models.

This study deals primarily with mathematical models, although graphical and written models are used as a means of communication between the mathematical model and its user.

Mathematical models can be further categorised into one or more of the following (Howard):
1. Positive versus normative. A normative model simply represents a statement of fact (e.g. multiplying price by volume will give revenue). A positive model attempts to explain one or more of the variables (e.g. representing volume sales as a function of selling price and advertising expenditure).

2. Deterministic versus probabilistic. A deterministic model has single figure estimates of variables, whereas a probabilistic model incorporates an underlying probability distribution (either discrete or continuous) for at least some of the variables.

3. Static versus dynamic. In static models all variables relate to a single time period, whereas in dynamic models there is more than one time period and these are linked in some manner.

4. Descriptive versus optimising. The variables in a descriptive model are all totally defined, whereas in optimising models some variables are defined only in terms of upper and lower limits, and the solution involves searching for the best values of these with respect to some criterion.

3.2 THE USE OF MODELS IN DECISION RECOGNITION

"The need for a decision is identified as a difference between information on some actual situation and some expected standard".

Mintzberg et al (1976)

A difference between an actual and expected situation is commonly used as an indication that a problem exists (Bonge (1972), Pounds (1969)). It reflects the need to make a decision after an actual problem has arisen (i.e. it is suited for reactive decision-making).
The statement can be extended to include anticipatory decisions:

The need for a decision is identified as a difference between information on some actual situation and some expected standard, or information on some predicted situation and some desired standard.

This restatement acknowledges the importance of problem avoidance by looking for potential as well as actual problems.

According to Pounds, a "difference" is defined by comparing what is actually perceived with the results of a model which predicts the same variable. He identifies four basic types of model which are used by managers in the search for differences:

1. historical models, based on the manager's past experience and consequent expectations for the future;
2. planning models, based on the organisation's formal statement (plan) of what is expected to happen;
3. other peoples' models, based on the expectations of other people (such as customers);
4. extra-organisational models, based on a comparison with external agents (such as similar organisations).

The use of planning models as a standard is the only formalised comparison available. Yet as Pounds and Ackoff (1977) point out, because planning models are used as a relative measure of success, they tend to be very conservatively biased towards some acceptable minimum, and thus act only as a lower bound on performance. Because of this they will not, in general, be in themselves suitable for problem recognition.

Instead of using a planning model for problem recognition, it may be preferable to use a separate, smaller model. This "control model" should be based on the premise that it is better to be able to foresee a potential problem than to be forced to react to an actual problem. However the ability to recognise potential or even actual problems is made difficult because of differences between the perceived
and objective environments. Thus a further function of this control model should be as a monitor of the organisation's perceptions of the environment.

The first use of the control model is prediction: in order to anticipate a problem the likely future(s) must be identified. There are three parts to the model:

1. Environment. As the environment is of such critical concern at the strategic level, it is modelled directly. The model takes the form of expectations and forecasts of future relevant environmental factors over the period of interest (the model horizon). All of these items will be based on assumptions which arise directly from the organisation's perception of the environment.

2. Organisation. The major existing resource conversion processes are modelled, along with those that are planned up to the model horizon. The existing processes will be based on historic performances.

3. Interface. The major inputs and outputs of the organisation, and the critical dependencies between these and the environment. These can be based on subjective estimates and/or econometric analysis.

When the parts are integrated the environmental section becomes the principal exogenous component and as such drives most of the model.

The result of a model run will be a prediction of organisational performance over the planning horizon. If this performance differs significantly from the broad strategic objectives then a problem exists, and the next phase of the decision process (diagnosis) is invoked.

The second use of the model is as a monitor of the organisation's perceptions of itself and the world. Some time after the predictive
run, the predicted results can be compared with the actual results, and the following established:

1. Were the predicted environmental factors in agreement with the actual environment that subsequently occurred? If not then there was a disparity between the organisation's environmental assumptions and the actual environment (i.e. the world was not as perceived). These disparities must be analysed to discover the inconsistencies as well as any critical factors that may have been overlooked.

2. Is the prediction of the organisation in agreement with the situation that actually occurred? This can be established by running the original organisation/interface components \textit{ex post} with the actual environment. Any differences indicate that these components do not agree with reality, so either the underlying assumptions behind the model are incorrect, or the model has been mis-specified with respect to these assumptions.

The processes are illustrated in figure 3.1.

A third possible use of the model is in reactive situations, where an event (or events) occur that might constitute a threat. The event and its perceived effects can be incorporated into the model. The results would serve to indicate the magnitude of the problem.

The main advantage of using a model in this manner, apart from those due to the enhanced ability to recognise problems, is that it has structured an otherwise informal phase of the strategic management process. It forces a statement of (at least some of) the organisation's operating assumptions, and provides a framework for testing these assumptions with respect to the environment.
Figure 3.1: Use of a Model in Decision Recognition.

a) Problem Avoidance.
Figure 3.1: Use of a Model in Decision Recognition.

b) Monitoring of Perceptions.
3.3 THE USE OF MODELS IN DECISION DIAGNOSIS

"... The first step following recognition is the tapping of existing information channels and the opening of new ones to clarify and define the issues..."

Mintzberg et al (1976)

"The diagnostic task that faces a manager is to decide which "differences" discovered in the control activities are merely symptoms and which "differences" are basic causes (problems) ...

Bonge (1972)

A properly constructed control model, as described in the previous section, will be based on stated assumptions. At least part of the diagnosis routine will involve an examination of these. A change in one or more of the assumptions may explain the aberration in behaviour that has caused the problem. This could be tested to some degree by incorporating the revised assumption(s) into the control model and rerunning it to see if the observed deviation is accounted for.

It is important at this stage not only to find out what caused the problem, but also how the problem was caused, i.e. if the assumptions made were correct then the problem should not have arisen, so it is necessary to establish both where the assumptions were wrong, and why they were wrong.

A major change in the environment, such as a change in government policy, may precipitate immediate diagnosis, bypassing the problem recognition phase. The immediate question to be answered is whether or not such a discontinuity poses a threat? It may be possible to see if this does constitute a problem by incorporating the perceived changes into the control model and running it to check that the predicted results are acceptable. Equally important, an attempt should be made to establish how the discontinuity was able to occur without advance warning.

3.4 THE USE OF MODELS FOR SCREENING ALTERNATIVES

"It is a superficial routine, more concerned with what is infeasible than with determining what is appropriate".

Mintzberg et al (1969)
Depending upon the type of situation, there are two types of model which could be of use in the screening process:

1. an optimisation model. The particular feature of optimisation is that it works with a feasible region. A model that incorporates the reasonable alternatives and the corresponding operating constraints could be built, and the result would indicate a "ballpark" area for more detailed analysis.

2. ad hoc models. Simple models of possible alternatives could be constructed. Those alternatives whose models produce unsatisfactory results could be disregarded in subsequent stages of the decision-making process.

All models built for screening purposes would have to be simple, incorporating only the main characteristics of each alternative. Similarly, because of the effort involved in constructing individual ad hoc models, only those alternatives which are marginal should be treated in this manner. Alternatives which can be obviously accepted for further analysis, or rejected outright need not be modelled.

3.5 THE USE OF MODELS IN THE EVALUATION-CHOICE ROUTINE

Evaluation-choice is the traditional area in which formal models have been used. At this stage several different possible courses of action have been identified, and the main task is to evaluate each one and choose the "best" (if none are acceptable the process cycles back to the development, search and design phases). Thus, the problem at this stage takes on some form of structure, and is amenable to computer techniques.

The modelling process here is well defined. A model of the actual situation is built, based upon a stated set of assumptions (if a control model exists the assumptions ought to be the same). Overlaid onto this model is each of the chosen alternatives in turn, so that each possibility is checked against the others according to a common
set of assumptions and criteria. Alternatively one model incorporating all of the alternatives could be built, and the evaluation made by optimising this with respect to some stated criterion.

Ultimately, possibly after several cycles, one alternative will appear more attractive than others. This is then selected for further analysis to test for robustness and risk. If it is not acceptable the cycle repeats and a new alternative is generated.

This approach is what Mintzberg refers to as the "analytic mode" of the evaluation process. He suggests that in fact there is "very little use of such an analytical approach, a surprising finding given the importance of the decision processes studied". He found that "judgment seems to be the favoured mode of selection, perhaps because it is the fastest, most convenient, and least stressful ... it is especially suited to the kinds of data found in strategic decision-making".

Some types of decision involve several levels of decision-maker (R & D project selection, capital budgeting). In some cases the diagnosis, development, research and design, as well as an initial evaluation is done at one level. The alternative selected is then passed up to a higher level of decision-maker for final approval. Typically this person or group will have several such alternatives, but insufficient resources to approve them all. They will initiate their own decision process, perhaps searching for more information, and even revamping some alternatives. They will ultimately, however, have to select a subset of the proposals for implementation. In this situation an optimising model, which evaluates all proposals simultaneously with respect to a consistent (but not necessarily correct) set of assumptions, would be useful.

3.6 A CONCEPTUAL FRAMEWORK FOR MODELLING DECISIONS

This section combines the various usages of formal models in the top-level decision process into one modelling framework. The framework is based on three components (environment, organisation and interface), and its design reflects the need to investigate incremental decisions under multiple environmental scenarios. The framework is constructed in
such a way that it is independent of any solution technique. This permits simulation (both deterministic and probabilistic) and optimisation methods to be used as required.

3.6.1 Basic Model Components

The model is based on a set of time based variables that reflect the critical factors being modelled. The values of these variables, defined in terms of other variables or constants, are specified by a set of functions. The variables and the functions can be grouped into three distinct subsets, each forming a separate and identifiable model.

1. Organisational model. This is composed entirely of variables and functions that represent the critical resource conversion processes within the organisation. It is descriptive in nature (see section 3.1) being based upon historic relationships and planned future relationships. It corresponds to the $L_{11}$ type interactions in the organisational system, as discussed in section 1.5.

2. Environmental model. This is a simple, predictive model of the expected future state of the environment. As it is clearly impossible to model the whole environment, only those factors which are seen as being critical to the success or otherwise of the organisation are included. The expected future state can be estimated either by formal forecasting techniques or subjectively, or both. This model represents the $L_{22}$ effects in the organisational system.

3. Organisation-environment interface model. This represents the effects of changes in the environmental variables upon the organisational variables. In as much as it predicts the levels of inputs and
outputs (i.e. the $L_{12}$ and $L_{21}$ effects) by relating them to the level of environment activity, it is positive in nature.

The three components are linked into a single well-defined model by the relationships between the variables. The principal linkages are achieved by the interface sub-model.

### 3.6.2 Environmental Scenarios

Multiple alternative environment models (each representing a possible future environment scenario) may be used in the framework. The need for this arises through the inability to predict the future with certainty. Different environmental scenarios will be based on a common set of environment variables, but with different relationships between the variables. Treating the environment model as a separate entity enables it to be replaced at any time by an alternative environment model without needing to alter the structure of either the organisational or the interface model. Since the normative effects are transmitted via the interface model, the organisational model will reflect the changes due to the different environment.

If multiple alternative environment scenarios are defined, it should be possible to provide a measure of the relative likelihood of each scenario occurring. If this likelihood is expressed as a probability, it then becomes possible to establish individual probability distributions for the organisational variables. These distributions can be used as an indication of the risk associated with the particular variable due to likely fluctuations in the environment.

### 3.6.3 Modelling Decisions

A model designed primarily for supporting decision analysis should be able to model the incremental effects of different decisions. The framework provides the ability to specify a set of decision alternatives in terms of each of their impacts upon the critical factors of the organisational system (as represented by the variables included in each of the basic model components). Each alternative (or set of alternatives) can be applied to the model and its impact viewed without having to alter the original model.
specification.

There are two basic approaches to evaluating decisions. The first of these is where each decision alternative is considered individually (incremental), and the second is where all decision alternatives are evaluated simultaneously (synoptic). The first approach corresponds to deterministic simulation, whereas the second corresponds to discrete optimisation.

3.7 SUMMARY

This chapter has looked at models and the way they can be used in the decision-making processes of the organisation. From this, a conceptual framework for modelling organisational decisions was developed. A more rigorous discussion of this framework is given in the next chapter.
Chapter Four:
A Mathematical Framework for Modelling.

The requirements for a modelling system have been discussed in the preceding chapters. The translation of these concepts into practice requires a mathematical framework on which to base the model definition and solution. It is only after this framework has been established that the modelling approach can be formalised into a workable technique by means of a computer programme.

The modelling approach is centred around a kernel which has three basic components. The first is the internal model of the organisation. The second component is a model of the critical factors that characterise the environment in which the organisation operates. These are linked to the organisational component by means of the environmental interface, which is the third component of the kernel. A mathematical outline of these processes is described in section 4.1.

The ability to run the model over a set of possible scenarios which represent likely future states of the environment was a further requirement for a modelling system. Running the model under a different environment scenario is equivalent to replacing the environmental component of the kernel with that scenario. A mathematical framework for this is developed in sections 4.21.

A further requirement was the ability to model the impact of different decision options. This can be accomplished by modifying those parts of the kernel (probably either organisation or interface) that are

---

1. The structure of the decisions and scenarios used in this framework have been adapted from the "project tree" concept of Gear and Lockett (1973). The role of decisions and scenarios in optimisation and risk analysis is discussed in Chapter 5.
directly affected by the decision so that they reflect the impact of the decision. A mathematical description of this is developed in section 4.3.

The mathematical structures used to define the various components of the model must permit the model to be solved. Three basic solution mechanisms have been recognised: deterministic simulation, risk analysis and optimisation. The application of these techniques based on the mathematical framework is developed in section 4.5.

To specify and solve an actual model will require a variety of ancillary processes. The most basic of these is the translation of the mathematical framework, as developed in this chapter, into a usable and understandable tool. This is discussed in chapter 5. Other facilities, such as the calculating and reporting of results, will also be required. These are implementation details and are not concerned with the mathematical framework. As such these are discussed in chapter 8.

4.1 Model Definition

The mathematical basis for the basic modelling kernel is well defined, and is used in most model-oriented systems (e.g. IFPS, FPS, PLANCODE). The model is based on a set of variables \( V \), which can be represented as a matrix (see figure 4.1).

A model consists of a set of definitions (functions) \( F \) which map a set of the elements of \( V \) onto other elements in \( V \).

\[
F : V \rightarrow V
\]  
(4.1)

For each element \( v_{ij} \in V \) there may exist a corresponding element \( f_{ij} \in F \) such that

\[
f_{ij} : V + v_{ij}
\]  
(4.2)

Define \( X_{ij} \) as the minimum subset of \( V \) such that

\[
f_{ij} : X_{ij} \rightarrow v_{ij}
\]  
(4.3)

The nature of this kind of model is such that:-
1. the past can be used to predict the future, but the future cannot be used to determine the present\(^2\) (i.e. lag variables are valid, but lead are not).

i.e. \( f_{ij} : X_{ij} + v_{ij} \Rightarrow X_{ij} \subseteq V \wedge (v_{mn} \in X_{ij} \Rightarrow n < j) \) (4.4)

2. a variable may not be defined to be a (direct) function of itself.

i.e. \( f_{ij} : X_{ij} + v_{ij} \Rightarrow X_{ij} \subseteq V \wedge v_{ij} \notin X_{ij} \) (4.5)

Note that this does not exclude simultaneous definitions.

e.g. given \( f_{ij} : X_{ij} + v_{ij} \) and \( f_{rs} : X_{rs} + v_{rs} \)

where \( v_{rs} \in X_{ij} \) and \( v_{ij} \in X_{rs} \)

\[\begin{array}{cccccccc}
0 & 1 & 2 & \ldots & T-2 & T-1 & T \\
0 & v_{00} & & & & & \\
1 & v_{10} & & & & v_{1T} & \\
2 & & & v_{22} & & & \\
\vdots & & \vdots & & \vdots & & \vdots & \\
m-2 & & & & & & & \\
m-1 & & & & & & & \\
m & v_{m0} & & & & v_{mT} & \\
\end{array}\]

(Time)

Variables

\( m \) = number of variables.

\( V_{ij} \) = \( i \)th variable, \( j \)th period.

Figure 4.1: The Solution Matrix.

2. This does not hold for performance measures such as present value or rate of return.
Thus a model \( M \) is a set of functions \( F \) which operates on a set of variables \( V \) represented as a matrix

\[
M = \{ f_{ij} \mid \forall m, v_{mn} \in X_{ij} \Rightarrow (n \neq j) \lor (n = j \land m \neq i) \}
\]

\( (v_{ij} \in V, f_{ij} \in F) \) \hspace{1cm} (4.6)

There are three basic types of function:-

1. **Scalar functions**, where a particular element is defined to be a specified value (constant)

\[
f_{ij} : \mathbb{R} \rightarrow v_{ij}
\]

(where \( \mathbb{R} \) is the set of real numbers)

\[
e.g. \ v_{ij} = c \quad (c \text{ is a constant})
\]

2. **Intra-temporal functions**, where one element is defined in terms of other elements in the same time period (column)

\[
f_{it} : (v_{jt}) \rightarrow v_{it}, \quad j \neq i,
\]

\[
e.g. v_{ij} = v_{kj} + v_{ij}
\]

3. **Inter-temporal functions**, where one element is defined in terms of elements in other (preceding) time periods (columns)

\[
f_{it} : (v_{rs}) \rightarrow v_{it}, \ s < t
\]

\[
e.g. v_{ij} = v_{ik}
\]

These can, of course, be mixed

\[
e.g. v_{ij} = c \cdot v_{ij-1} + v_{kj}
\]

A model is completely defined (i.e. it can be solved) if a value can be found for each element in \( V \). This is clearly possible only if there exists a function for every element, \(^{3}\)

\[
i.e. \text{ a model is completely defined if}
\]

\[
\forall v_{ij} \in V, \exists f_{ij} \in F \hspace{1cm} (4.7)
\]

---

3. This is a necessary but not always sufficient requirement.
Since many of the functions for a particular row will be the same, it is useful to define a vector function
\[ F_i : X_i \rightarrow V \Rightarrow \forall n, f_{\text{in}} : X_{\text{in}} \rightarrow v_{\text{in}}, X_{\text{in}} \subseteq X_i, X_{\text{i}} \subseteq V \]  \hspace{1cm} (4.8)

### 4.1.1 Types of Model

The three components of the basic model (organisational, environmental, interface) can be represented as partitions of the matrix \( V \). (figure 4.2).

Define the sets \( V_O \), \( V_E \), and \( V_I \) to be subsets of \( V \) such that:

\[ V_O = \{v_{ij} | v_{ij} \in V, i \leq \ell \} \]  \hspace{1cm} (4.9)

\[ V_I = \{v_{ij} | v_{ij} \in V, \ell < i \leq \ell + \kappa \} \]  \hspace{1cm} (4.10)

\[ V_E = \{v_{ij} | v_{ij} \in V, i > \ell + \kappa \} \]  \hspace{1cm} (4.11)

where \( V_O \) represents the set of organisational variables, \( V_I \) the set of interface variables, and \( V_E \) the set of environmental variables.

The set of functions can also be divided into three subsets, \( F_O' \), \( F_I' \), \( F_E' \) representing the organisational, interface and environmental functions respectively, but the basis for the partition is different.

\[ \begin{array}{ccccccc}
 & 0 & 1 & 2 & \ldots & T-2 & T-1 & T \\
\hline \\
\text{Organisational} & & & & & & & \\
\text{(\( V_O \))} & & & & & & & \\
\hline \\
\ell & & & & & & & \\
\ell+1 & & & & & & & \\
\hline \\
\text{Interface} & & & & & & & \\
\text{(\( V_I \))} & & & & & & & \\
\hline \\
\ell+\kappa & & & & & & & \\
\ell+\kappa+1 & & & & & & & \\
\hline \\
\text{Environmental} & & & & & & & \\
\text{(\( V_E \))} & & & & & & & \\
\hline \\
\ell+\kappa+\nu & & & & & & & \\
\end{array} \]  

(Time)

Figure 4.2: The Partitioned Solution Matrix.
The set of organisational functions, $F_O$, contains all those functions whose domain and range are members of $V_O$ (i.e. all those functions that operate entirely on the organisational variables, for example depreciation calculations).

$$f_{ij} \in F_O \iff (f_{ij} : x_{ij} \rightarrow v_{ij} \Rightarrow (x_{ij} \subseteq V_O \land v_{ij} \in V_O)) \quad (4.12)$$

The set of environmental functions, $F_E$, contains all those functions whose domain and range are members of $V_E$ (i.e. all those functions that operate entirely on the environmental variables, e.g. inflation and interest rate relationships)

$$f_{ij} \in F_E \iff (f_{ij} : x_{ij} \rightarrow v_{ij} \Rightarrow (x_{ij} \subseteq V_E \land v_{ij} \in V_E)) \quad (4.13)$$

The set of interface functions $F_I$ contains all those functions whose range is contained in $V_O$, but whose domain spans $V_E$ and $V_I$, and possibly even $V_O$ (i.e. an interface function is one that assigns some value to an organisational variable, the value being derived at least in part from the environment, e.g. the effect of inflation on costs).

$$f_{ij} \in F_I \iff (f_{ij} : x_{ij} \rightarrow v_{ij} \Rightarrow (x_{ij} \subseteq V) \land (x_{ij} \cap V_E \neq \emptyset) \land (v_{ij} \in V_O)) \quad (4.14)$$

There are three other useful categories of functions:

1. where the organisation acts to alter the characteristics of its interface with the environment. This represents a change in the strategic posture of the organisation. The set $F_S$ contains all such functions:

$$f_{ij} \in F_S \iff (f_{ij} : x_{ij} \rightarrow v_{ij} \Rightarrow x_{ij} \subseteq (V_O \cup V_I) \land (v_{ij} \in V_I)) \quad (4.15)$$

2. where the organisation affects the environment proactively (i.e. the environmental variables are determined by organisational variables), for example where advertising a product increases the total market size for that product. The set $F_P$ contains
all such functions:-
\[ f_{ij} \in F_p \iff (f_{ij} : X_{ij} + v_{ij} \Rightarrow (x_{ij} \subseteq V) \land (v_{ij} \in V_e) \land (x_{ij} \cap V_o \neq \emptyset)) \quad (4.16) \]

3. where the characteristics of the interface are affected by the environment. This represents an unexpected change in the strategic posture of the organisation, e.g. marketshare suddenly increases when a competitor drops out. The set \( F_x \) contains all such functions:
\[ f_{ij} \in F_x \iff (f_{ij} : X + v_{ij} \Rightarrow (x \subseteq V_e) \land (v_{ij} \subseteq V_i)) \quad (4.17) \]

The possible types of interaction are shown in figure 4.3. Note that in practice, when the interface relationships are very simple, it is often convenient to map the environmental variables directly onto the organisational variables.

4.2 Scenario Definition

It is not possible to know the future state of the environment with certainty.\(^4\) \( F_e \) as represented thus far is an estimation of one possible future state of the world.\(^5\) A requirement of the modelling framework is the ability to explore the effects of a variety of future environmental scenarios.

Let there be \( n \) possible (mutually exclusive) future environmental scenarios, represented by:
\[ e = \{ F^1_e, F^2_e, F^3_e, \ldots, F^n_e \} \quad (4.18) \]
where \( F^k_e \) is the set of functions that represent the \( k \)th environmental scenario, and \( e \) is the set of all sets of environmental functions.

---

4. Similarly some aspects of the future organisation and its environment interface may not be known with certainty. The mathematical handling of these uncertainties is the same as those in the environment, and as such, will not be treated separately.
5. Strictly, \( F_e \) is a set of transition relationships, and \( V_e \) represents the future state of the world generated by these.
(F₀)
Organisational
L₁₁

(Fₑ)
Environmental
L₂₂

(Fᵢ)
Input/Output
L₁₂ & L₂₁

(Fₛ)
Planned Change of Interface.
Strategic Reposturing.

(Fᴾ)
Proactive Effects on Environment

(Fₓ)
Unexpected Change of Interface.
Strategic Surprise

Figure 4.3: Types of Interaction.
Let \( M_E \) represent the model as previously defined, but with no environment (i.e. \( F_E \)) specified,

\[
M_E = F_0 \cup F_1 \cup F_s \cup F_x \cup F_p
\]  \hspace{1cm} (4.20)

There exist \( n \) possible models, \( (M_E^i, i=1,\ldots,n) \) based on future possible states of the environment, where

\[
M_E^i = M_E \cup F_E^i, \quad F_E^i \in e
\]  \hspace{1cm} (4.21)

Suppose that each scenario \( F_E^i \) has a probability \( p_i \) of occurrence, such that:

\[
\sum_{i=1}^{n} p_i = 1
\]  \hspace{1cm} (4.22)

Then the mapping

\[
M_E^i : V \rightarrow V
\]  \hspace{1cm} (4.23)

will occur with a probability \( p_i \), and thus the value associated with each \( v_{qr} \in V \) by the set of functions \( M_E^i \) will also occur with a probability \( p_i \). Similarly when \( M_E^j \) is used, other values will be generated for each \( v_{qr} \in V \) with a probability \( p_j \). In this manner, the models \( M_E^1, M_E^2,\ldots,M_E^n \) can be used to define probability distributions for all of the elements in \( V \). These distributions represent the sensitivity of each element with respect to the environment.

4.2.1 Dynamic Scenarios

The scenarios described above are essentially composite in nature. Each scenario represents a complete statement as to one possible future state of the environment. It is easy to envisage however situations in which it is desirable or necessary to have different sets of scenarios for different segments or elements of the environ-
ment. Consider, for example, the environmental scenario in figure 4.4 which is based on market share and inflation. This composite scenario can be usefully broken into two smaller scenarios, one for market share and one for inflation. (see figure 4.5). These scenarios are essentially interacting and dynamic.

| Either | Increasing Market Share and High Inflation |
| Or    | Increasing Market Share and Medium Inflation |
| Or    | Increasing Market Share and Low Inflation |
| Or    | Static Market Share and High Inflation |
| Or    | Static Market Share and Medium Inflation |
| Or    | Static Market Share and Low Inflation |
| Or    | Decreasing Market Share and High Inflation |
| Or    | Decreasing Market Share and Medium Inflation |
| Or    | Decreasing Market Share and Low Inflation |

**Figure 4.4**
Composite scenario for Market Share and Inflation

One of
- Either Increasing Market Share
- Or Static Market Share
- Or Decreasing Market Share

And one of
- Either High Inflation
- Or Medium Inflation
- Or Low Inflation

**Figure 4.5**
Two Dynamic Scenarios for Market Share and Inflation
The set of dynamic scenarios can be described by defining $E$ as:

$$
E = \{ e_1, e_2, \ldots, e_m \} 
$$

(4.24)

where $e_i$ is the set of scenarios for factor $i$,

and

$$
e_i = \{ F_{E_i}^1, F_{E_i}^2, \ldots, F_{E_i}^n \} 
$$

(4.25)

where $F_{E_i}^j$ is the set of functions that define the $j$th possible future for that factor,

\begin{align*}
&\text{i.e. } F_{E_i}^j = \{ f_{rs}^j | f_{rs}^j : x_{rs} \Rightarrow (x_{rs} \subseteq v_E \\
&\quad \wedge v_{rs} \in v_E \wedge v_{rs} \notin x_{rs}^j) \} 
&\end{align*}

(4.26)

Each possible scenario represented by $F_{E_i}^j$ will have a probability of occurrence $p_i^j$, and

$$
\sum_{j=1}^{n} p_i^j = 1, \quad i = 1, \ldots, m 
$$

(4.27)

A specific future is composed of a combination of one element (i.e. scenario) from the sets of possible scenarios for each factor (see figure 4.6). For a future to be valid, no item must be defined ambiguously (i.e. more than once) in the combination of component scenarios, i.e.

$$
\forall (F_{E_i}^j \in e_i \wedge F_{E_r}^s \in e_r) \quad j, s = 1, \ldots, m, j \neq s
$$

$$
F_{E_i}^j \cap F_{E_r}^s = \emptyset 
$$

(4.28)

If the cardinality of $e_i$ is denoted by $|e_i|$, and $W$ is the number of possible states of the world described by $E$, then

$$
W = \prod_{i=1}^{m} |e_i| 
$$

(4.29)

6. For example, if scenario $k$ in $e_1$ (i.e. $F_{E_1}^k$) defines inflation to be 15%, and scenario $j$ in $e_2$ ($F_{E_2}^j$) defines inflation to be 20%, then the composite future $F_{E_1}^k \cup F_{E_2}^j$ will have inflation defined to be both 15% and 20%, which is clearly a conflict.
For example, let $E = \{e_1, e_2, e_3\}$

$e_1 = \{F_{E1}^1, F_{E1}^2\}$

$e_2 = \{F_{E2}^1, F_{E2}^2, F_{E2}^3, F_{E2}^4\}$

$e_3 = \{F_{E3}^1, F_{E3}^2, F_{E3}^3\}$

Then $|e_1| = 2, |e_2| = 4, |e_3| = 3$

$w = |e_1| \times |e_2| \times |e_3| = 24,$

and the possible combinations are shown in figure 4.6.

\[\begin{array}{cccc}
F_{E1}^1 \cup F_{E2}^1 \cup F_{E3}^1 & F_{E1}^2 \cup F_{E2}^1 \cup F_{E3}^1 \\
F_{E1}^1 \cup F_{E2}^1 \cup F_{E3}^2 & F_{E1}^2 \cup F_{E2}^1 \cup F_{E3}^2 \\
F_{E1}^1 \cup F_{E2}^1 \cup F_{E3}^3 & F_{E1}^2 \cup F_{E2}^1 \cup F_{E3}^3 \\
F_{E1}^1 \cup F_{E2}^2 \cup F_{E3}^1 & F_{E1}^2 \cup F_{E2}^2 \cup F_{E3}^1 \\
F_{E1}^1 \cup F_{E2}^2 \cup F_{E3}^2 & F_{E1}^2 \cup F_{E2}^2 \cup F_{E3}^2 \\
F_{E1}^1 \cup F_{E2}^2 \cup F_{E3}^3 & F_{E1}^2 \cup F_{E2}^2 \cup F_{E3}^3 \\
F_{E1}^1 \cup F_{E2}^3 \cup F_{E3}^1 & F_{E1}^2 \cup F_{E2}^3 \cup F_{E3}^1 \\
F_{E1}^1 \cup F_{E2}^3 \cup F_{E3}^2 & F_{E1}^2 \cup F_{E2}^3 \cup F_{E3}^2 \\
F_{E1}^1 \cup F_{E2}^3 \cup F_{E3}^3 & F_{E1}^2 \cup F_{E2}^3 \cup F_{E3}^3 \\
F_{E1}^1 \cup F_{E2}^4 \cup F_{E3}^1 & F_{E1}^2 \cup F_{E2}^4 \cup F_{E3}^1 \\
F_{E1}^1 \cup F_{E2}^4 \cup F_{E3}^2 & F_{E1}^2 \cup F_{E2}^4 \cup F_{E3}^2 \\
F_{E1}^1 \cup F_{E2}^4 \cup F_{E3}^3 & F_{E1}^2 \cup F_{E2}^4 \cup F_{E3}^3 \\
\end{array}\]

Figure 4.6
Possible Combinations of Scenarios
The probability of a particular combination future occurring is the product of the probabilities of each component scenario of that future.\(^7\)

e.g. the probability of the future composed of 
\[ F_{E_1}^1, F_{E_2}^4, F_{E_3}^2 \] is given by:

\[ P(F_{E_1}^1 \cup F_{E_2}^4 \cup F_{E_3}^2) = P_1^1 \times P_2^4 \times P_3^2 \]  
(4.30)

By simple probability theory, the sum of the probabilities of all possible futures will be equal to one.

\[ \sum_{i=1}^{W} P(\text{future}_i) = 1. \]  
(4.31)

4.2.2 Scenario Interdependencies

An underlying assumption behind the dynamic scenarios described above is that the probability of a given scenario occurring is not affected by the occurrence of another scenario, i.e. that the scenarios are independent. Such conditions will not hold in all situations.

Consider, for example, the situation where separate scenarios have been specified for both inflation and interest rates, as shown in figure 4.7. Here the events "high inflation" and "low inflation" are judged to have an equal probability of 0.5, as are the events "high interest rate" and "low interest rate". This gives the following joint probabilities:

- Probability of high inflation and low interest rate = 0.25
- Probability of high inflation and high interest rate = 0.25
- Probability of low inflation and low interest rate = 0.25
- Probability of low inflation and high interest rate = 0.25

---

7. Assuming the scenarios are independent.
In fact, one would intuitively expect there to be some positive correlation between interest and inflation rates, so that the cases "high inflation and high interest rate" and "low inflation and low interest rate" would have a higher probability of occurrence than the other two cases.

One method of taking such interdependencies into consideration is to combine all of the dependent factors together into one set of scenarios, with more realistic joint probabilities (see figure 4.8). This approach loses some of the advantages of dynamic scenarios, and may be cumbersome to use when there are a lot of factors.

A different approach is to express the probability of the occurrence of one set of scenarios in terms of the occurrence of the other scenarios (i.e. conditional probabilities). This is illustrated in figure 4.9. This approach preserves the benefits of dynamic scenarios, and enables interdependencies to be specified.

One of
\begin{align*}
\{ & \text{either high inflation (probability = 0.5) } \\
& \text{or low inflation (probability = 0.5) } \\
\}
\end{align*}

and one of
\begin{align*}
\{ & \text{either high interest rate (probability = 0.5) } \\
& \text{or low interest rate (probability = 0.5) } \\
\}
\end{align*}

**Figure 4.7**
Inflation and interest rate scenarios

One of
\begin{align*}
\{ & \text{either high inflation and high interest rate (probability = 0.35) } \\
& \text{or high inflation and low interest rate (probability = 0.10) } \\
& \text{or low inflation and high interest rate (probability = 0.15) } \\
& \text{or low inflation and low interest rate (probability = 0.40) } \\
\}
\end{align*}

**Figure 4.8**
Combined scenarios
The general case is:-

Let $e_i$ be the set of scenarios for factor $i$
and $e_j$ be the set of scenarios for factor $j$,
i.e. $e_i = \{F_{Ei}^1, F_{Ei}^2, \ldots, F_{Ei}^n\}$  \hspace{1cm} (4.32)
and $e_j = \{F_{Ej}^1, F_{Ej}^2, \ldots, F_{Ej}^m\}$  \hspace{1cm} (4.33)

Define $P_{rs}$ to be the probability of the compound scenario $F_{Ei}^r \cup F_{Ej}^s$ occurring,
then $P_{rs} = P(F_{Ei}^r)P(F_{Ej}^s|F_{Ei}^r)$  \hspace{1cm} (4.34)
where \[ \sum_{q=1}^{m} P(F_{Ej}^q|F_{Ei}^r) = 1, \] \hspace{1cm} $r = 1, \ldots, n$  \hspace{1cm} (4.35)

But if factors $i$ and $j$ are independent,
then \[ P(F_{Ej}^s|F_{Ei}^r) = P(F_{Ej}^s), \] \hspace{1cm} $i = 1, \ldots, m$
and \[ P_{rs} = P(F_{Ei}^r)P(F_{Ej}^s) \] \hspace{1cm} (4.36)

One of
\[
\begin{cases}
\text{either} & \text{high inflation (probability = 0.5)} \\
\text{or} & \text{low inflation (probability = 0.5)}
\end{cases}
\]

and one of
\[
\begin{cases}
\text{either} & \text{high interest rate (probability = if high inflation then 0.7 otherwise 0.2)} \\
\text{or} & \text{low interest rate (probability = if high inflation then 0.3 otherwise 0.8)}
\end{cases}
\]

\textbf{Figure 4.9}
\textit{Conditional Scenarios}
In terms of the inflation-interest rate example,

figure 4.7 represents the independent case,
so the \( P(E_i^r) \)'s are specified;

figure 4.8 represents the combined case,
so the \( P_{rs} \)'s are specified;

figure 4.9 represents the conditional case,
so the \( P(E_i^r|E_j^s) \)'s are specified.

4.3 DECISION DEFINITION

A requirement of the modelling system was the ability to model decisions and their effects on the organisation. The nature of strategic decisions is that they will, at the very least, alter the organisation's interface with its environment. Additionally, they may affect the functioning of the organisation and perhaps even alter the environment.

Let there be \( m \) possible decisions to be made, represented by the set \( D \)

\[
D = \{ d_1, d_2, d_3, \ldots, d_m \} \tag{4.38}
\]

where \( d_j \) is the \( j \)th decision.

Let each decision \( d_j \) be composed of a set of mutually exclusive choices or alternatives,

\[
d_j = \{ F_{Dj}^1, F_{Dj}^2, F_{Dj}^3, \ldots, F_{Dj}^n \} \tag{4.39}
\]

where \( F_{Dj}^k \) is the set of functions that represents the effect on the organisational system (i.e. the organisation-interface-environment) of the \( k \)th alternative of the \( j \)th decision.
The decision-maker must choose one option \( F_{Dj}^k \) from each possible decision \( dj \). Note that the null decision (continue with the status quo) can be represented either explicitly as one of the \( F_{Dj}^k \), or implicitly by ignoring the decision \( d_j \).

There are two kinds of effects which a decision option may have. The first of these is a change in the actual structure of the organisational system. This can be represented by redefining the appropriate functions in the model. The second effect is incremental, where the basic structure is not changed, but the level of activity is. An example of this is where, if a certain decision alternative is selected, the capital expenditure for the year will increase by a given amount.

Let \( I \) be the set of incremental (additive) functions \( I_{jk} \) such that:

\[
I_{jk} \in I \Rightarrow f_{jk}(X_{jk}') + I_{jk}(X_{jk}) = v_{jk}'
\]

\[
(X_{jk} \subset V) \land (v_{jk} \in V) \land (v_{jk} \not\in X_{jk}')
\]

(4.40)

Note that incremental functions are meaningful not only in the context of decision options but also in scenarios.

Thus \( F_{Di}^k \) is the set of definitional functions and incremental functions that represent the effects of the \( k \)th option of decision \( i \).

\[
i.e. \ f_{mn}^k \in F_{Di}^k \Rightarrow f_{mn}^k : X \rightarrow v_{mn}'
\]

\[
(X_{mn} \subset V) \land (v_{mn} \in V) \land (v_{mn} \not\in X_{mn})
\]

(4.41)

and \( I_{mn}^k \in F_{Di}^k \Rightarrow f_{mn}(X_{mn}) + I_{mn}^k(X_{mn}) = v_{mn}
\]

\[
(X_{mn} \subset V) \land (v_{mn} \in V) \land (v_{mn} \not\in X_{mn})
\]

(4.42)

Let \( M \) define the original model,

\[
i.e. M = F_0 \cup F_I \cup F_S \cup F_X \cup F_F \cup F_E
\]

(4.43)
Then the model \( M_D \) that represents the impact of a set of decision options \( F_D \) upon the structure of the basic model (with or without scenarios) is given by:

\[
M_D = M \cup F_D \tag{4.44}
\]

where \( F_D = \bigcup_{i=1}^{m} F_{D_i} \), \( F_{D_i} \in d_i \).

(4.45)

Note that, just as for dynamic scenarios, the combination of decision options must be unambiguously defined, i.e.

\[
\{ f_{ij} | f_{ij} \in F_{D_k} \} \cap \{ f_{ij} | f_{ij} \in F_{D_p} \} = \emptyset
\]

\[
\forall F_{D_k} \in d_k, F_{D_p} \in d_p,
\]

\[
k, p = 1, \ldots, m, k \neq p
\]

(4.46)

If the cardinality of \( d_i \) (including the null decision) is denoted by \( |d_i| \), then the number of possible decision combinations \( Y \) is given by:

\[
Y = \prod_{i=1}^{m} |d_i| \tag{4.47}
\]

An example of a decision is shown in figure 4.10.

**Options**

<table>
<thead>
<tr>
<th>Either</th>
<th>Sample impacts:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build smelter now</td>
<td>Increase capital expenditure (1981) by $400 million; Increase revenue (1985) by $100 million.</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>Don't build smelter (null decision)</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td>Increase capital expenditure (1985) by $400 million x inflation; Increase revenue (1989) by $100 million x inflation.</td>
</tr>
<tr>
<td>Delay smelter until 1985</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.10**

Sample decision to build an aluminium smelter
4.4 SCENARIOS AND DECISIONS

From the preceding discussion of scenarios and decisions, it is clear that they are both similar in concept. Both have the same structure and the same requirements. The only difference in fact is that decisions are under the control of the organisation. It is assumed that, should the organisation select a particular option, then that course of action will occur.

Scenarios, on the other hand, represent events which are outside of the control of the organisation. Thus there is no certainty as to which (if any) of the possible futures will occur. This uncertainty, as represented by a probability measure, is the only difference between scenarios and decisions. From all other aspects, both mathematical and intuitive, the concepts are identical.

4.5 SOLUTION TECHNIQUES

Three basic methods of solving models were recognised in the previous chapter as being potentially useful. The methods were deterministic simulation, risk analysis, and optimisation. The manner in which these fit into the mathematical framework is discussed in this section.

4.5.1 Deterministic Simulation

The primary method of solution is deterministic simulation. For this method to be applicable the model must be completely defined\(^8\) (cf equation 4.7). The solution problem for deterministic simulation can be stated as:-

\[
\forall v_{ij} \in V \exists f_{ij} \in F,
\]

find the value for a particular \(v_{ij}\)

---

8. This is a necessary condition, but not in itself sufficient. If the model contains simultaneous relationships then these must have a unique solution. Simultaneous relationships occur naturally in financial models, such as in calculating the debt required to finance the interest on the debt.
The method proceeds by observing that there exists a function \( f_{ij} \) that defines \( v_{ij} \) in terms of other members of the set \( V \),

\[
i.e. f_{ij}(X_{ij}) = v_{ij}, \quad X_{ij} \subseteq V, \quad v_{ij} \notin X_{ij}
\]

Should \( X_{ij} \) be completely defined in terms of a real number (i.e. \( X_{ij} \subseteq \mathbb{R} \)), then the solution is found, i.e. \( v_{ij} = X_{ij} \). However, typically \( X_{ij} \) will contain other elements of the set \( V \). Assume that \( X_{ij} = \{v_{mn}, v_{pq}\} \). It is therefore necessary to find a value for \( v_{mn} \) and \( v_{pq} \) before a value can be calculated for \( v_{ij} \). These are calculated in exactly the same manner as used to find \( v_{ij} \) (i.e. the process is one of recursive substitution). Ultimately a value will be found for a particular \( v_{rs} \) (when \( X_{rs} \subseteq \mathbb{R} \)), and this can be substituted into all other functions that use \( f_{rs} \).

This method of recursive substitution is sufficient to solve any model (or part thereof) that is completely defined without simultaneous relationships. Where such relationships exist, it is necessary to use special techniques such as Gauss-Siedel or Gaussian elimination. For example:-

find \( v_{ij} \) where

\[
f_{ij}(X_{ij}) = v_{ij}, \quad v_{kj} \in X_{ij}, \quad v_{ij} \notin X_{ij}
\]

and \( f_{kj}(X_{kj}) = v_{kj}, \quad v_{ij} \in X_{kj}, \quad v_{kj} \notin X_{kj} \).

Recursive substitution cannot be used here as the value of \( v_{ij} \) is needed to calculate that of \( v_{kj} \), and vice versa.

An example of recursive substitution as illustrated in figure 4.11.

4.5.2 Simulation for Risk Analysis

The organisation does not exist in isolation from its environment. Many fluctuations in the environment will directly affect the organisation. The nature of the organisation is such, however, that its future state cannot be predicted with certainty. Risk analysis is a method which gives an indication of the effects of possible future environmental states on the critical variables of the organisation.
\[ f_{ij}(x_{ij}) = v_{ij} \]
\[ f_{kt}(x_{kt}) = v_{kt} \]
\[ x_{ij} = \{v_{kt}, v_{mn}\} \]
\[ x_{kt} = \{v_{pq}, v_{rs}\} \]
\[ f_{pq}(x_{pq}) = v_{pq} \]
\[ f_{mn}(x_{mn}) = v_{mn} \]
\[ f_{rs}(R) = v_{rs} \]
\[ x_{pq} = \{v_{rs}\} \]
\[ x_{mn} = \{v_{rs}, v_{xy}\} \]
\[ f_{xy}(x_{xy}) = v_{xy} \]
\[ x_{xy} = \{v_{ab}, v_{mn}\} \]
\[ f_{ab}(R) = v_{ab} \]

Figure 4.11: Deterministic Simulation by Recursive Substitution.
The use of probabilistic scenarios was discussed in section 4.2. The solution by recursive substitution of a model incorporating one probabilistic future will give a certain value for each model variable. The probability of that set of values occurring is the same as the probability of that future occurring.

When the model is solved over all futures, or even "many" futures, a set of probabilistic values for each variable is generated. These represent a probability distribution, the shape of which can be used as an indicator of the risk.

4.5.3 Optimisation

Optimisation techniques can be used to generate "good" solutions quickly. These can be used as starting points in the evaluation-choice phase of the decision-making process.

In an optimising framework there is a set of variables whose values are not defined uniquely. Instead, the values are defined to lie on some range delimited by an upper and lower bound. The solution technique finds values for these variables that best fulfil some optimisation criteria (represented by an objective function which is to be either maximised or minimised).

Consider the model

\[ \mathbf{M} : Y \rightarrow X, \quad Y, X \subseteq V \]

define \[ W = V - X, \]

i.e. \[ W \] is the set of all variables which are not defined by the model.

The optimisation problem can be stated as:-

maximise \( g(V) \)

st \[ \mathbf{M} : Y \rightarrow X, \quad Y, X \subseteq V \]

and \( h(W) \geq 0 \), \( W = V - X \)

where \( g(V) \) represents the objective function

and \( h(W) \geq 0 \) represents a set of functions that defines limits for the unknown
Optimisation considerations are discussed in more detail in the next chapter.

4.6 SUMMARY

In this chapter a mathematical description has been developed for modelling organisations. The modelling framework is based on the concepts discussed in chapter three. Three methods for solving models were also outlined.

The next chapter discusses in more detail the use of optimisation methods in this framework. To illustrate the framework, an example model is presented in chapter six.
Chapter Five: Optimisation and Modelling.

Discussion thus far has centred on the development of an integrated framework for modelling based on concepts of organisational decision processes. Little attention has been given to the potential role of optimisation techniques in decision support models.

This chapter studies the background of optimising techniques and existing models which have used these techniques, especially in the area of financial planning. From these models, a consistent framework is developed which integrates simulation and optimisation techniques.

There are very few reports of optimising models that have been used to evaluate strategic alternatives. The appendix to this chapter discusses a small system built to investigate the structure and computer implementation of such time-based systems.

5.1 TYPES OF OPTIMISATION

Optimisation techniques have been used for many years as aids in corporate decision-making. The range of managerial activities affected by these methods includes production scheduling, inventory control, fleet scheduling, and many investment and financial applications.

There are numerous types of optimisation method available. Probably the most commonly used is mathematical programming, such as linear programming, quadratic programming, and integer-linear programming.
A general form of a mathematical programme is:

Minimise \ c(X)
Subject to \ AX \geq d
and \ X > 0

where \ X \ is \ a \ column \ vector \ of \ decision \ variables,
\ c(X) \ is \ a \ cost \ function \ in \ X \ (the \ objective \ function),
\ A \ is \ a \ matrix \ of \ coefficients,
\ d \ is \ a \ (column) \ vector \ of \ real \ numbers.

Other types of optimisation methods available include dynamic programming, Lagrange-multiplier techniques, geometric programming, and methods for unconstrained optimisation\(^1\).

The most widely used optimisation techniques are those, such as linear programming, which are based on the Simplex method. Their wide use is attributable in part to the ready availability of computer programmes based on this relatively simple method.

5.2 OPTIMISATION MODELS IN CORPORATE FINANCIAL PLANNING

A wide variety of these models has been proposed in the literature over the past two decades. Included in this are models by Weingartner (1963) and subsequent derivatives (for example, Hamilton and Moses (1973), Lockett and Gear (1975), Rychel (1977)), which are based on project selection under conditions of capital rationing. Other models have attempted to draw more heavily on finance theory (Eubank (1972), Carleton, Dick and Downes (1973), Myers and Pogue (1974)).

Parallel developments in optimisation modelling include attempts to handle risk and uncertainty (Naslund (1966), Eubank (1972), Lockett and Gear (1975)), and to handle multiple objective problems, typically with variants of goal programming (Merville and Tavis (1974), Ashton and Atkins (1979)). Non-mathematical programming techniques, such as dynamic programming, have also been applied to financial planning.

---

1. For a more complete discussion of optimisation techniques see Dixon (1972).
problems (e.g. Krouse (1969)).

It is perhaps indicative of the complexities involved in building and running mathematical programmes that most reported models are theoretical, with few actual working implementations² (Snowden et al (1979)).

5.3 SIMULATION AND OPTIMISATION IN FINANCIAL POLICY MAKING

Carleton et al (1973) have examined the role of both simulation and optimisation models in financial planning. In their view, simulation models (or "budget generators") fall short in the area of financial policy in three principal ways:

1. they tend to encourage "bottom-up" modelling which focuses attention on low level instead of high level matters;

2. they tend to avoid a formal treatment of uncertainty, simulating a number of alternative, non-probabilistic scenarios;³

3. they are not good screeners of financial plans, relying upon repeated reruns of the model in an unguided "hunt and peck" manner.

Carleton et al then describe a usable financial optimising system which they designed "... guided by the principle that contemporary finance theory could only be usable if the mind numbing effects of large scale model size, computer mechanics, and the mathematics of linear programming were removed".

The resultant model is of a fixed structure (i.e. it was a generalised model which could only be altered by changing the model parameters). It was intended that its outputs would act as the inputs to a more detailed simulation model, thus providing an elementary form of risk and sensitivity analysis.

---

2. A notable exception is Hamilton and Moses (1973).
3. The increasing availability of Monte-Carlo routines in simulation packages may have overcome this problem.
5.4 OPTIMISATION FRAMEWORK: THE GEAR-LOCKETT MODEL

The optimisation framework selected for implementation in this study is based on that developed by Gear and Lockett (1973). The Gear-Lockett (G-L) model has the advantage that, in its original form, it closely parallels the approach taken by Weingartner (1963) and Hamilton and Moses (1973). In the more generalised form developed here, it appears to be able to handle a wide variety of published linear optimisation models.

The G-L model is based on a set of multi-period decision trees (termed 'project trees'). In each period, each branch of every tree has indicated on it a resource requirement, representing the amount of resources required from the resource pool should that branch be selected. A solution is found by selecting the optimal combination of branches (according to some payoff function) such that the resource constraints are not broken. A small example is shown in figure 5.1.

5.4.1 Formulation of Gear-Lockett Model

The basic visual statement of the problem, as represented by the sets of project trees, can be transformed into a zero-one mathematical programming problem by the following sequence of steps. Working in from the initial period:

1. each decision node is represented by a set of zero-one variables, one for each branch of the node. An "exclusion" row is emitted for each node to ensure that only one branch is selected for each decision.

2. constraint rows are emitted for the period, each column coefficient corresponding to the resource requirement for the decision branch for that period.

3. for each chance node, one branch only is selected, and the process repeats itself for that period from step (1) until the end is reached.

4. For a rigorous exposition of the method see Cowie (1977).
Figure 5.1 An Example of the Gear-Lockett Model.
4. at the end the benefit values for each branch are multiplied by the compound probability of that set of branches and added to the cumulative benefit value for each column (thus forming an objective function). The process returns to the last chance node with unprocessed branches. The next branch from this node is selected and the process returns to step (1).

5. when all the branches from the chance nodes have been done the process is completed and the result is a zero-one programme.

Although this process is extremely difficult to perform manually, it is possible to automate it and thus relieve much of the computational burden (Cowie (1977)). An example of a zero-one programme generated by this method is shown in figure 5.2.

The resulting zero-one programme is functionally similar to the Weingartner model with no budget deferrals, but extended to allow recourse decisions to be made under conditions of uncertainty (termed "contingency programming" - Gear and Cowie (1980)). The solution to the zero-one programme is a sequence of decisions to make (i.e. branches to select) under every relevant probabilistic outcome.

5.4.2 Strengths of Gear-Lockett Model

The principal strength of the G-L model is the simple visual statement of the problem. This is easy to understand and can give valuable insights into the nature of the problem under consideration.

The visual statement of the problem has one other critical property: it does not suggest or rely on any particular solution technique. Thus, although the approach taken by Lockett and Gear was zero-one programming, other approaches, such as dynamic programming or simulation, would have

5. This is illustrated by the "WRONZ problem"(Cowie (1977)), which, although mathematically intractible using zero-one techniques, was able to be "solved" by visual inspection. The solution is believed to be very nearly optimal.
<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X21A</th>
<th>X22A</th>
<th>X21B</th>
<th>X22B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclusion</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Inclusion</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Inclusion</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Resource, Period 1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>&lt; 9</td>
</tr>
<tr>
<td>Exclusion, Future A</td>
<td>-1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource, Period 2, Future A</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td>&lt; 8</td>
</tr>
<tr>
<td>Exclusion, Future B</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Resource, Period 2, Future B</td>
<td>1</td>
<td>3</td>
<td></td>
<td>5</td>
<td>1</td>
<td></td>
<td>&lt; 8</td>
</tr>
<tr>
<td>Objective: Maximise</td>
<td>12</td>
<td>6</td>
<td>7.5</td>
<td>2.5</td>
<td>7.5</td>
<td>2.5</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2  The Formulation of the Example in Figure 5.1 into a 0-1 Programme.
been equally applicable.

A further strength of the approach is that the trees may represent a variety of different items. In fact each tree can represent a "decision category" and each branch possible strategies within that category. For example, a tree could represent a division of a company with each branch representing a possible development strategy (as in Hamilton and Moses (1974)).

5.4.3 Shortcomings of Gear-Lockett Model

The G-L model has several major drawbacks:

1. The zero-one programmes generated are "exhaustive" in that every possible decision in every possible combination of chance futures is represented. As each chance branch doubles the matrix size for the remainder of the problem (i.e. all subsequent decision variables and constraints), the mathematical programme rapidly becomes intractibly large. This means that not only might it become impossible to solve the resulting zero-one programming problem, but that it might not even be possible to formulate it (Cowie (1977)).

2. The exhaustive data required for the problem, representing a complete set of decisions and probable events, is difficult to obtain. Furthermore, since the solution represents a complete set of future contingency decisions under all circumstances, and since all circumstances cannot be known with certainty at the outset, the value of such a complete solution seems dubious at best.

3. The output from the zero-one programme is extremely hard to relate back to the original problem.

6. In fact a mixture of optimisation and simulation was initially used to solve this type of problem (Lockett and Gear (1973)).
4. Complete independence is assumed between the project trees. In many situations this is unrealistic as:-
   (i) there may be synergistic effects between different projects;
   (ii) there may be some correlation among the chance events;
   (iii) resource consumption among trees may be correlated to external factors.

5. The branches can only represent incremental changes in resource usage. This is too simplistic for corporate modelling where a wide variety of variables are required, and where the selection of a particular project branch may, especially at the strategic level, alter the structure of the model (e.g. introduce a complete new variable or constraint).

A potential method of overcoming the size problem was suggested by Lockett and Muhlemann (1974) and extended by Cowie (1977). Basically the approach involves "pushing back" all chance and decision nodes to the very first period, so that each project tree contains a single node only. This approach, while losing the contingent feature of the model, does allow good initial solutions to be taken (Cowie) and, from a practical standpoint, recognises that subsequent decisions will almost certainly be dependent upon information which only becomes available as time progresses.

An attempt was made to overcome the interdependence problem by recognising external (environmental) and internal uncertainties (Gear and Cowie (1980)). In this extension to the basic formulation, separate stochastic trees were constructed for different environmental factors, each tree representing a set of probabilistic forecasts of a specific environmental indicator. These were then related to the branch resource requirements in the project trees by indices similar to those proposed by Sharpe (1963). Unfortunately this approach, while overcoming the problem of resource usage correlated to environmental factors, does not address the problems of project synergies and correlated uncertainties.
5.5 MODIFICATIONS TO THE GEAR-LOCKETT MODEL

In order to overcome the problems described in the previous section, it was necessary to make some extensions and some simplifications to the G-L model. The result is a powerful modelling technique which completes the framework described in the previous chapter.

5.5.1 The Problem of Model Size

To overcome the problems of large model size and concomitant computational difficulties, the pushed-back framework of Lockett and Muhlemann was adopted as the basis for all model inputs. Thus all decision and stochastic trees must be specified as single node structures, but with no limit on the number of branches. This differs from the Lockett-Muhlemann approach in that, instead of collecting a complete set of contingent data and reducing the problem by push-back, the data is collected in pushed-back form.

All nodes are considered to occur after period zero (the present period) but before period one (the next period). This does not impose any restrictions on the actual timing of decisions as the branches can be empty until the actual impacts are felt.

To further improve upon the mathematical tractability of the models the decision-maker must select a set of possible futures from the stochastic trees over which to optimise the model. The resulting mathematical programme is thus no longer chance constrained. However a different form of risk analysis of the optimal solution is made possible. This is discussed further in section 5.8.

5.5.2 The Problem of Contingency Data Collection

As previously stated, data is collected in "pushed-back" form. This overcomes the onerous data requirements of the original contingency programming formulation.
5.5.3 The Problem of Output Readability

The mathematical programming process is only one part of an integrated set of modelling techniques. Associated with the implementation of this framework is a set of report generators and other associated presentation methods designed according to precepts of decision support systems and software engineering. Thus the problem of output readability is overcome in the implementation of the framework.

5.5.4 The Problem on Non-Independence of Project Trees

The three types of interdependencies noted in the previous section have been overcome by a variety of language and implementation features.

1. Project Synergy: The wide range of language features in the implementation have largely overcome the problem of modelling project synergies. Calculations can, for example, be conditional on the level of other model variables, or special "synergy variables" may be introduced to model conjoint effects.

2. Chance Event Correlation: The provision of subjective conditional probability estimates on the chance branches allows correlations to be modelled.\(^7\)

3. Environmental Correlations: These may be modelled either by using simple language features of the implementation (e.g. multiplying cost variables by the inflation rate), or by a series of "response curves", which are discussed in section 5.6.

5.5.5 The Problem of Structural Alterations

In order to provide for structural changes to the model instead of incremental resource usages it is necessary to move away from zero-

---

\(^7\) Using Bayes theorem it is possible to calculate the a priori probability estimates.
one programming as a solution technique. This is because, in order to reflect a structural change that is conditional upon a decision being taken, the zero-one (decision) variable must be multiplied by the decision variable(s) involved in the structural alteration. This results in a non-linear problem.

To overcome this, the decision-maker selects the set of decisions which he wishes to consider. The model may then be formulated as a linear programme with a user-specified objective function. Other combinations of decisions can then be selected and optimised in this manner so that the decision-maker can quickly explore a large variety of likely combinations, mentally screening out those which are likely to prove unsatisfactory. This approach is in keeping with the basic aim of decision support systems which is to combine the judgmental abilities of man with the computational capabilities of computers.

5.6 PROJECT INTERDEPENDENCIES WITH THE ENVIRONMENT

An approach to modelling project interdependencies with the environment was outlined by Gear and Cowie (1980). The relevant external factors were represented as Sharpe-type indicators on separate stochastic trees. Each project tree (or resource type) has associated with it a set of indices which measured the impact of a unit change by the indicators on the elements of that project. Extending this approach to the general case gives the following:

Let $I$ be the set of external indicators such that

$$I = \{i_1, i_2, \ldots, i_m\}$$

where $i_j \in I$ is the indicator associated with environmental factor $j$.

Let $T$ be the set of project trees such that

$$T = \{t_1, t_2, \ldots, t_n\}$$

where $t_j \in T$ is the $j$th project tree.

---

8. These environmental influences are in fact the $L_{21}$ and $L_{12}$ effects discussed in section 1.5.
Let $\beta$ be the set of indices such that

$$\beta = \begin{pmatrix} \beta_{11}, \beta_{12}, \ldots \beta_{1m} \\ \beta_{21}, \beta_{22}, \ldots \beta_{2m} \\ \beta_{n1}, \beta_{n2}, \ldots \beta_{nm} \end{pmatrix}$$

where $\beta_{jk}$ is the impact of a unit increase in $i_k$ upon one unit of $t_j$, all other factors being equal.

Then the impact of a set of changing external factors on $t_j \in T$ is given by:

$$\prod_{k=1}^{n} \left( 1 + \beta_{jk} \Delta i_k \right)$$

(5.1)

where $\Delta i_k$ is the inter-temporal change in the level of the $k$th exogenous indicator,

i.e. $\Delta i_k = i_k^t - i_k^{t-1}$

where $i_k^t$ is the level of the indicator in period $t$.

This gives the exogenously influenced value ($t'_j$) for $t_j$ as

$$t'_j = t_j \prod_{k=1}^{n} \left( 1 + \beta_{jk} \Delta i_k \right)$$

(5.2)

The above relationship assumes a linear change in $t_j$ regardless of the degree of change in the level of the indicator (see figure 5.3). In fact it is easy to visualise situations in which this response curve is not linear\textsuperscript{9}, as illustrated in figure 5.4.

The linear case, as described above, can be extended as follows:

1. a number of estimates of the impacts upon each $t_j$ over the likely range of changes in the indicators are made. This results in a set of points similar to that shown in figure 5.5.

\textsuperscript{9} Economic "laws", such as diminishing marginal returns and economies of scale, as well as business concepts such as the experience curve, illustrate this point.
Figure 5.3: Linear Response Curve as Used in Gear-Cowie Model.

Figure 5.4: Non-Linear, Negatively Correlated Response Curve.

Figure 5.5: Estimated Point Impacts of Indicator Changes.

Figure 5.6: Interpolation of Intermediate Impacts.
2. for a given change in an indicator level 
\( (\Delta i_k) \), the corresponding impact can be 
estimated by interpolation (figure 5.6).

In this manner varying degrees of impact can be modelled. This
approach is analogous to the response curves used in marketing models
such as Brandaid (Keen (1980)).

The preferred method of interpolation to use would be one that
preserves the continuous shape of the curve (e.g. cubic spline). However,
recognising that the points that generate the curve are
essentially subjective and hence not accurate, simple linear interpola-
tion is used in the final implementation (figure 5.7).

5.7 THE ADVANTAGES OF RESPONSE CURVES

The use of response curves as described in the previous section has
several advantages over traditional modelling methods, especially
where more than one data source is used, such as in multi-divisional
companies.

1. The approach enforces an explicit treatment of the
environment and its organisational interface, as
discussed in chapter one.

2. Preparing projections is made easier as these are
based entirely on the continuation of operations
given no changes in the environment. It is
likely that this data would be able to be
prepared more readily than data which fully
accounts for fluctuations due to a dynamic
environment.

3. The data supplied from all sources is more
likely to be consistent in that it is all
based on one easily comprehended set of
environmental assumptions.

4. Any number of independently prepared environ-
mental forecasts can be run against the base
data.

Thus, it is envisaged that models utilising environmental response
Figure 5.7: Linear Interpolation of Intermediate Points.
curves will be quicker to prepare, more consistent, and more flexible than those prepared in the traditional manner.

5.8 RISK ANALYSIS

The inclusion of stochastic trees in the framework means that it is possible to describe many probabilistic forecasts or expectations in a single model. Typically when the model is being solved, either by deterministic simulation or by optimisation, it is solved for a pre-selected subset of these possible futures. This subset is composed of a single branch from each stochastic tree. The probability of the resultant state of the world occurring is given by the product of the probabilities on each of the selected stochastic branches.

The nature of the uncertainty, as represented by the sets of probabilistic scenarios, means that it is not possible to predict with certainty which actual combination of branches (if any) will occur. Thus a policy which looks promising under one future may be disastrous under another. Provision has therefore been made in the framework to test any given policy under any selected combination of scenarios. Thus it is possible for example to test a policy under combinations that represent pessimistic, optimistic, and likely futures, each with an associated probability of occurrence.

If a policy is tested under all possible futures (or even "a lot" of futures), it is feasible to construct a probability distribution for any of the variables in the model that are affected by the stochastic branches. This is achieved by noting that different scenario combinations will produce for each variable a value which has a probability equal to the probability of that future occurring. The accumulation of these results over all futures will result in a distribution.

After finding a good solution (policy) for a specific future by linear programming, the decision-maker is able to simulate that policy over all possible futures and thus build a series of distributions for key variables and constraints. The range and shape of the
distribution is an indication of the risk involved in following that particular policy. Furthermore the distribution corresponding to a particular constraint indicates the probability of the relationship being broken. For critical financial or legal relationships, this can be a valuable indication of the likelihood of a particular policy violating some crucial limit and thus confronting the organisation with a crisis.

5.9 SUMMARY

This chapter has studied the structure of time-based corporate optimising models. A framework has been developed based on that initially proposed by Gear and Lockett. This framework has been extended to overcome many of the shortcomings in the original Gear-Lockett model. It has been possible to include in the framework techniques for handling a diverse range of synergy and other inter-dependencies. A feature of the resultant framework is that it is not specifically technique oriented, providing for both optimisation and deterministic simulation. By exploiting this feature, it has been possible to develop a simple yet powerful method for investigating risk.

The appendix to this chapter discusses LPGen, a simple modelling system which was constructed to investigate the implementation characteristics of optimising models. The findings from this system will enable the construction of a comprehensive system which embodies the modelling framework developed in the previous chapters of this thesis. An example model based on this framework is described in the next chapter.
APPENDIX TO CHAPTER FIVE

IMPLEMENTATION TECHNIQUES FOR OPTIMISATION MODELS

The very few reported implementation successes for optimisation-based corporate models (see section 5.2) suggested that it would be useful to construct a small, prototype optimising system to investigate the basic nature of this type of model. The aim of this system, named LPGen (Linear Programme Generator), was not to implement the framework discussed in the previous sections of this chapter, but instead to investigate the effects of language and system implementation on optimising models. Specifically the aims were:

1. to provide a simple high-level system for the construction of time-based optimisation models;
2. by studying the models built using LPGen, to establish the basic underlying structure of such models and their relationship with simulation models;
3. to synthesise the structure of a high level language that would be able to implement this structure in the previously discussed modelling framework.

LPGen belongs to a class of computer programme termed "matrix generators" (Fourer and Harrison (1978)). It differs from other matrix generators in that it is specifically designed to handle inter-temporal models. Thus LPGen is easier to use on this type of model than other matrix generators. However for more complex problems the other generalised systems are superior.

10. A system which does implement the framework is discussed in chapter 8.
11. Other matrix generators include ALPS (Steinberg (1977)), EZLP (Jarvis and Papaconstadopoulos (1978)), LPMODEL (Katz et al (1980)).
5A.1 Structure of Financial Linear Programming Models

A financial linear programming model will consist of the following:-

1. A model horizon, representing the number of time periods in the model;
2. A set of variables, each variable occurring in each time period of the model;
3. A set of relationships and constraints that represent the logical structure of the model;
4. Some initial data;
5. An objective function.

A mathematical statement of this is:-

\[
\begin{align*}
\text{Min } & \sum_{i=1}^{n} w_i x_{it} & 0 \leq t \leq T \\
\text{s.t. } & \sum_{i=1}^{n} a_{it} x_{it} = c_{it} & 0 \leq t \leq T \\
& x_{i0} = b_i & 1 \leq i \leq n \\
& x_{it} \geq 0 & 1 \leq i \leq n, 0 \leq t \leq T
\end{align*}
\]  

(1) (2) (3) (4)

where

- \( n \) is the number of variables,
- \( T \) is the model horizon,
- \( x_{it} \) is variable \( i \) in period \( t \),
- \( t = 0 \) represents the present period.

The objective function is represented by (1), the relationships and constraints by (2), and the initial data by (3). The requirement in linear programming that all variables are non-negative is represented by (4).

5A.2 LPGen Language Structure

The language embodied in LPGen was designed to represent the structure of financial linear programming models in a simple manner. Because LPGen was seen to be an experimental system, it was not intended that the language be particularly rich or flexible. The simple form chosen however proved sufficiently versatile for most problems. An LPGen example is shown in figure 5A.1.
5 A.2.1 Identifiers

LPGen identifiers were constructed using the same rules as FORTRAN identifiers (i.e. a maximum of six alphanumeric characters, the first one of which must be alphabetic). These were passed directly into the output file with the only change being the appending of two digits to represent the period number.

5 A.2.2 Variables

All variables in LPGen were declared as belonging to one of four types:

- \textit{Indv} - a time independent continuous variable;
- \textit{Indi} - a time independent integer variable;
- \textit{Timev} - a time dependent continuous variable;
- \textit{Timei} - a time dependent integer variable.

All timev and timei variables would have a two digit period number appended after them on the output file.

5 A.2.3 Expressions

LPGen could handle only a very simple type of expression, the form of which was:

\[ c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \]

where \( c_1, c_2, \ldots, c_n \) represent constants in a standard FORTRAN format (except that the exponent character E was, for syntactic reasons, represented as @);

\( x_1, x_2, \ldots, x_n \) represent variable names (a given variable could only appear once in each expression, a source of great frustration);

the linear operator (shown above as a +) could be either a + or a -.

Note that because of the simpler linear nature of expressions, a multiplication operator was not required (and was in fact illegal).
5 A.2.4 Statements

The format of the statements that comprise the model logic was:

<expression> = c₁ [ , c₂ [ , c₃ ... ] ]

As is standard in linear programming the operators > and < refer to 'greater than or equal to' and 'less than or equal to' respectively. c₁, c₂, and c₃ are all constants. The multiple form of the right hand side allowed different right hand side values in different time periods.

5 A.2.5 Time periods

It was necessary to specify the time periods to which a particular timei or timev variable referred. This was accomplished by means of a FORTRAN type subscript. The form that this could take was:

(T-c) - used to specify a lagged period relative to the present period (T) (c is an integer constant);

(T) - used to specify the present time period. This was the default where no subscript was used;

(T+c) - used to specify a lead period relative to the present period;

(c) - used to specify a particular period c.

No indi or indv variable could be subscripted.

5 A.2.6 Initial data

All data, relationships and constraints that referred to a specific time period, or which were composed entirely of INDV or INDI variables were defined in a section headed by the keyword INIT. All TIMEI and TIMEV variables in this section had to be subscripted with an absolute period indicator (i.e. a subscript of form (c)).

5 A.2.7 Time relationships

The bulk of most financial models was composed of a set of statements or constraints that were repeated for each time period in the model. These were specified in a section preceded by the key word CYCLE(n),
where \( n \) was the number of time periods for which to generate statements. The cycle statement acted in a similar fashion to a FORTRAN DO loop, except that only one pass was made through the set of statements, each statement being replicated \( n \) times varying the subscript \( T \) from 1 to \( n \).

5 A.2.8 Objective function

The key word OBJ was used to denote the start of the objective function. The objective function took the form of a simple expression composed of IND or INDV variables, or TIMEI or TIMEV variables with absolute subscripting (i.e. subscripts not involving \( T \)).

5 A.2.9 Statement delimiters

All statements were input in free format spread over the first 72 columns of a card. Statements were delimited by semicolons.

5 A.2.10 Comments

Column 73 of a card or a percent symbol marked the end of a logical input record. Thus, comments could be included by preceding them with a percent symbol.

5 A.2.11 Options

Various run time options were available. These included:

- `MAXIMISE` - to cause the objective function to be maximised (this was the default);
- `MINIMISE` - to cause the objective function to be minimised;
- `AUTO` - to cause the automatic invocation of the linear programming package (Burroughs Tempo);
- `RANGE` - to print sensitivity data;
- `NAME` - to provide identification to the model.

5 A.2.12 Output

No special provisions were made for output, the main function of the system being to act as a linear programming preprocessor. However all variables were sorted into alphabetic order to assist reading the
Tempo output. Additionally a row "map" was printed so that LPGen statements could easily be matched with the appropriate Tempo row.

5 A.3 Review of LPGen

The construction of LPGen proved a very valuable exercise. Not only did it provide useful insights into the structure and use of financial models and modelling languages, but it also provided an extremely useful tool for the construction of linear programmes. The following were the main points that arose from LPGen.

5 A.3.1 Structure of financial linear programmes

A very clear distinction exists between the structural elements of a financial linear programme and the constraints. The structure of a model is made up mainly of accounting identities and financial relationships. Some of these identities and relationships will contain variables whose value is undefined. These represent the decision variables in the model. The constraints on the other hand operate as bounds on the solution space of these decision variables. It is very helpful in constructing models if the structural items can be kept separate from the constraints. Once the structure of the model is defined, it remains unchanged. Sensitivity analyses are conducted by varying parameters and constraints, not by varying structure. Initial data and forecast data should always be represented as parameters.

5 A.3.2 Relationship between simulation and optimisation models

The key distinction between a simulation and optimisation model is the number of undefined or decision variables. A linear programming model with no decision variables is, in fact, a simulation model and is calculated entirely using the structural relationships. It is not necessarily feasible with respect to the stated constraints.

5 A.3.3 Structural relationships acting as constraints

It was discovered that a carelessly constructed structural relationship could, in fact, act as an unwanted constraint, almost invariably producing an incorrect model. These implicit constraints were extremely difficult to find, even if it was realised that they in fact were present. A typical example is the use of a variable to
calculate the difference between two other variables. Consider, for example, the following statement to calculate the change in working capital over a period:

\[ \text{CWC} = \text{WC} - \text{WC(T-1)}; \]

where CWC is the change in working capital

WC is the level of working capital.

\[(\text{CWC} - \text{WC} + \text{WC(T-1)} = 0 \text{ is the LPGen representation}).\]

Because of the non-negativity requirements, CWC must be greater than or equal to zero, which implies that:

\[ \text{WC - WC(T-1)} \geq 0, \]

and hence \[ \text{WC} \geq \text{WC(T-1)}. \]

The net result is an implicit constraint stating that working capital may not decrease. Any high level linear programming system should remove all such implicit constraints, so that apart from non-negativity requirements, all constraints must be explicitly stated. This means that the structural relationships will never act as constraints.

5 A.3.4 Expression structure

The simple form of expressions used by LPGen was too limiting and inflexible (see, for example, the CWC example above). Full arithmetic expressions of the FORTRAN kind are essential if the full potential of a high level system is to be realised. For structural definitions the form should be similar to a FORTRAN assignment statement, i.e.

\[ <\text{variable}> <\text{assignment}> <\text{FORTRAN-type expression}>. \]

For constraints the form should be:

\[ <\text{FORTRAN-type expression}> <\text{relational operator}> <\text{FORTRAN-type expression}>. \]

This would make the following statements syntactically correct (although linearity would have to be proved):

\[ \text{SALES} + \text{SALES(T-1)} \times \text{INFLATION} \times \text{REALGROWTH}; \]

(structural definition)

\[ \text{DEBT} / \text{EQUITY} < \text{MAXDEBT/EQUITY}; \]

(constraint).
5 A.3.5 Identifier structure

Six characters is generally insufficient to form meaningful variable names. This restriction should be relaxed so that longer names may be used.

5 A.3.6 Output

Most of the output from financial models will be ultimately presented as some sort of financial statement. A linear programming system should have the facility to do this automatically. This would assist model debugging.

5 A.3.7 Balancing constraints

This is more of a modelling detail than a systems detail. A properly constructed model will have all sources and uses of funds properly accounted for, and thus will (if correct) balance. Thus there is no need for the inclusion of a balancing constraint (of the form assets = equity + liabilities). Experience has in fact shown that the inclusion of such a constraint can force the system to balance when it is incorrectly specified. Thus results which may seem to be correct may in fact be wrong, this being hidden by the fact that the system balances.
| NAME   | 0001 | 0002 | 0003 | 0004 | 0005 | 0006 | 0007 | 0008 | 0009 | 0010 | 0011 | 0012 | 0013 | 0014 | 0015 | 0016 | 0017 | 0018 | 0019 | 0020 | 0021 | 0022 | 0023 | 0024 | 0025 | 0026 | 0027 | 0028 | 0029 | 0030 | 0031 | 0032 | 0033 | 0034 | 0035 | 0036 | 0037 | 0038 | 0039 | 0040 | 0041 | 0042 | 0043 | 0044 | 0045 | 0046 | 0047 | 0048 | 0049 | 0050 | 0051 | 0052 | 0053 | 0054 | 0055 | 0056 | 0057 | 0058 |
|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| MAXI   | 0001 | 0002 | 0003 | 0004 | 0005 | 0006 | 0007 | 0008 | 0009 | 0010 | 0011 | 0012 | 0013 | 0014 | 0015 | 0016 | 0017 | 0018 | 0019 | 0020 | 0021 | 0022 | 0023 | 0024 | 0025 | 0026 | 0027 | 0028 | 0029 | 0030 | 0031 | 0032 | 0033 | 0034 | 0035 | 0036 | 0037 | 0038 | 0039 | 0040 | 0041 | 0042 | 0043 | 0044 | 0045 | 0046 | 0047 | 0048 | 0049 | 0050 | 0051 | 0052 | 0053 | 0054 | 0055 | 0056 | 0057 | 0058 | 0059 | 0060 |
Figu re 5.1.

- MGRAD = .17 GTRAD(T-1) = 01
- HPROP = .17 PCROP = 01
- GDTA = .09 GTRAD = 01

*INIT
- RNTPAD(1) = 01
- RNTPAD(2) = 101
- RNTPAD(3) = 101
- RNTPAD(4) = 101
- RNTPAD(5) = 101
- RNTPAD(6) = 01

*PROP(1) = .06 PROP(1) = 01
- PROP(2) = .32 PROP(2) = 01
- PROP(3) = .62 PROP(3) = 01
- PROP(4) = .53 PROP(4) = 01
- PROP(5) = .46 PROP(5) = 01

====================================================================================================
NUMBER OF ERRORS DETECTED = 0. NUMBER OF WARNING MESSAGES = 0.
FILE GRANT CONTAINS 721 RECORDS.
MATRIX CONTAINS 771 ENTRIES IN 259 ROWS. MPS/ALL INITIATED.
COMPILATION TIME = 130.8866 SECONDS ELAPSED: 5.7207 SECONDS PROCESSING: 1.6038 SECONDS I/O.
====================================================================================================
Chapter Six: Illustrative Example of Framework.

A small example model was built to illustrate the modelling framework developed in the previous chapters. The example, although hypothetical, is sufficiently representative of actual situations to be of some interest in its own right. The model has been deliberately kept simple to preserve its expository nature.

The model and accompanying analysis have been developed and explained as if they were the only inputs into the decision-making process. Clearly in actual situations this would not be so, as there are always many relevant factors and issues which cannot be modelled quantitatively. The use of models in a "decision support" environment, discussed in the next chapter, is one approach that attempts to overcome this problem.

6.1 MODEL BACKGROUND

The model is of a single division company which has been presented with an opportunity to expand. The purpose of the model is to evaluate the likely impact of the various decision alternatives which confront the company. Thus the model is primarily to aid the evaluation-choice routine in the decision process.

For simplicity, only the financial characteristics of the company will be considered. The company balance sheet and income statement for period zero (i.e. the "initial data" for the model) are shown below.
Balance Sheet
(at end of period 0)

Fixed Assets at Cost 10.0
Less : Provision for Depreciation 0.0

Book Value of Assets 10.0

Current Assets:
Stock 4.0
Debtors 3.0
Cash 0.0

7.0

Share Capital 9.5
Retained Earnings 0.0
Term Liabilities 4.0

Current Liabilities:
Bank overdraft 3.0
Creditors 0.5
Provision for Tax 0.0
Provision for Dividend 0.0

3.5

17.0

Income Statement
(for period 0)

Sales 20.0

Less Variable Costs:
Materials 7.0
Labour 3.0
Other Variable Costs 1.0

11.0

Gross Margin 9.0

Less : Fixed Costs 5.2
Depreciation 1.0
Interest 0.8

7.0

Net Profit before tax 2.0
Less tax 1.0

Net Profit after tax 1.0
6.2 STRUCTURE OF MODEL

The structure of the model follows the framework developed in chapter 4. It is composed of three basic components:

1. organisational, representing the internal operations of the organisation;
2. environmental, representing the environmental factors of interest;
3. interface, which relates the environment to the organisation.

Each of these components is explained further in the following sections.

6.2.1 Organisational Variables

The set of organisational variables \( (V_o) \) for this model contains the following:

- CA: current assets
- CASH: cash balance
- CL: current liabilities
- CRS: creditors
- DBTRS: debtors
- DEPN: depreciation
- DIV: dividend declared
- DIVC: cash paid out in dividends
- FA: book value of fixed assets
- FAAC: fixed assets at cost
- FC: fixed costs
- FCG: percentage growth in fixed costs
- GM: gross margin
- INT: interest paid
- LAB: labour cost
- MATS: materials cost
- NEWCAP: new share capital issued
- NEWFA: new fixed assets
- NOD: new overdraft
- NPAT: net profit after tax
- NPBT: net profit before tax
- NTDEBT: new term debt
- OC: other costs
- OD: overdraft
- OVC: other variable costs
- PDEPN: provision for depreciation
- PDIV: provision for dividends
- PTAX: provision for tax
- RETPR: retained profit
- ROD: overdraft repaid
- RTDEBT: term debt repaid
- SALES: value of sales
SALESG  percentage growth in sales
SCAP    issued capital
STOCK   inventories
TAX     assessed tax
TAXC    cash paid out in tax
TDEBT    term debt
VARCST    total variable costs
WC     working capital (excluding cash)

6.2.2 Organisational Functions

It will be recalled from chapter 5 that the set of organisational functions \( F_o \) contains all those functions whose domain and range are members of \( V_o \). Thus those functions which operate entirely on organisational variables are organisational functions.

The set of organisational functions for this model is as follows. Note that the functions contain parameters (the lower case items). These are discussed in the next section.

1. Materials, labour and other variable costs may be expressed as a percentage of sales.

\[
\text{MATS ARE } y\text{mats } \times \text{SALES} \quad (6.1)
\]
\[
\text{LAB ARE } y\text{dlab } \times \text{SALES} \quad (6.2)
\]
\[
\text{OVC ARE } y\text{ovc } \times \text{SALES} \quad (6.3)
\]

2. Total variable costs are the sum of all variable costs.

\[
\text{VARCST IS MATS } + \text{LAB } + \text{OVC} \quad (6.4)
\]

3. Gross margin is then given by:-

\[
\text{GM IS } \text{SALES-VARCST} \quad (6.5)
\]

4. Fixed assets at cost are the sum of the previous period's value and new fixed assets purchased

\[
\text{FAAC IS FAAC}(T-1) + \text{NEWFA} \quad (6.6)
\]

5. Fixed assets are depreciated using diminishing value.

\[
\text{DEPN is } y\text{depn} \times \text{FA}(T-1) \quad (6.7)
\]
\[
\text{PDEPN is } P\text{DEPN}(T-1) + \text{DEPN} \quad (6.8)
\]
\[
\text{FA IS } \text{FAAC} - \text{PDEPN} \quad (6.9)
\]
6. Stock, debtors and creditors are calculated as a percentage of sales.
   \[
   \text{STOCK IS } ystock \times \text{SALES} \quad (6.10) \\
   \text{DBTRS IS } ydtrs \times \text{SALES} \quad (6.11) \\
   \text{CRS IS } ycrs \times \text{SALES} \quad (6.12)
   \]

7. Current assets are the sum of stock, debtors and cash
   \[
   \text{CA IS } \text{STOCK} + \text{DBTRS} + \text{CASH} \quad (6.13)
   \]

8. Finance is available from long-term debt, bank overdraft and new share issues.
   \[
   \text{TDEBT IS } \text{TDEBT(T-1)} + \text{NTDEBT-RTDEBT} \quad (6.14) \\
   \text{OD IS } \text{OD(T-1)} + \text{NOD - ROD} \quad (6.15) \\
   \text{SCAP IS } \text{SCAP(T-1)} + \text{NEWCAP} \quad (6.16)
   \]

9. The dividend payable is a percentage of the previous years share capital.
   \[
   \text{DIV IS } ydiv \times \text{SCAP(T-1)} \quad (6.17)
   \]

10. Term debt is repaid at a percentage of the level of debt in the previous period.
    \[
    \text{RTDEBT IS } yrtdbt \times \text{TDEBT(T-1)} \quad (6.18)
    \]

11. Interest is payable on term debt and bank overdraft.
    \[
    \text{INT IS } yintd \times \text{TDEBT} + yintod \times \text{OD} \quad (6.19)
    \]

12. Other costs are calculated as
    \[
    \text{OC is } \text{FC} + \text{DEPN} + \text{INT} \quad (6.20)
    \]

13. Current liabilities are calculated as
    \[
    \text{CL ARE } \text{OD} + \text{CRS} + \text{PTAX} + \text{PDIV} \quad (6.21)
    \]

14. Net profit before tax
    \[
    \text{NPBT IS } \text{GM} - \text{OC} \quad (6.22)
    \]

15. Tax is paid as a percentage of profits after tax
    \[
    \text{TAX IS } ytax \times \text{NPBT} \quad (6.23)
    \]
16. Net Profit after tax

\[ \text{NPAT IS NPBT - TAX} \] (6.24)

17. A proportion of the tax is paid in the year incurred, and the remainder in the following year.

\[ \text{TAXC IS PTAX(T-1) + ytaxc \times TAX} \] (6.25)
\[ \text{PTAX IS (1-ytaxc) \times TAX} \] (6.26)

18. A percentage of the dividend is paid in the year it is declared, and the remainder in the following year.

\[ \text{DIVC IS PDIV(T-1) + ydivc \times DIV} \] (6.27)
\[ \text{PDIV IS (1-ydivc) \times DIV} \] (6.28)

19. Cash is calculated as the starting cash balance plus the cash inflows less the cash outflows.

\[ \text{CASH IS CASH(T-1) + NPBT + DEPN + NEWCAP} \]
\[ \text{+ NOD + NTDEBT - TAXC - DIVC - RTDEBT} \]
\[ \text{- ROD - NEWFA + WC(T-1) - WC} \] (6.29)

20. Working capital (excluding cash) is calculated as

\[ \text{WC IS STOCK + DBTRS - CRS} \] (6.30)

6.2.3 Model Parameters

The parameters used in the preceding section have been assigned the following values:

- \( ycrs \) : 10%
- \( ydtrs \) : 15%
- \( ydepn \) : 10%
- \( ydiv \) : 4%
- \( ydivc \) : 50%
- \( ydlab \) : 15%
- \( yintd \) : 13%
- \( yintod \) : 10%
- \( ymats \) : 35%
- \( yovc \) : 5%
- \( yrtdebt \) : 10%
- \( ystock \) : 20%
- \( ytax \) : 50%
- \( ytaxc \) : 30%
6.2.4 **Environmental Variables**

The organisation has identified two measures of the environment as being important to its operations. These then compose the set of environmental variables ($V_E$):

- **INFLATION**  the inflation rate
- **ECONOMIC**  the "level of economic activity".

Note that the ECONOMIC variable is a surrogate measure for more usual econometric-based indices. The projections for such variables in practice would be based on subjective estimates by the decision-makers. Thus they are a distillation of experience and expectation, based on a wide variety of inputs, whereas the more traditional economic indicators (such as, perhaps, INFLATION) are based on forecasting estimates.

6.2.5 **Environmental Scenarios**

A scenario has been prepared for each environmental variable (see figure 6.1). It has been assumed that the scenarios are independent (i.e. the level of economic activity does not affect the rate of inflation, and vice versa). Six possible states of the world are defined by these scenarios. The probability estimates, enclosed in parentheses, are subjective. Note that the functions shown on the scenarios operate entirely on environmental variables, and hence are environmental functions.

6.2.6 **Interface Relationships**

The set of interface functions relates changes in the environment onto the organisation. For simplicity linear response curves, as discussed in section 5.6, have been assumed.

The interface variables used are:

- **BES**  the impact upon sales of a percentage change in the level of economic activity
- **BIS**  the impact upon sales of a percentage change in the inflation level
- **BIFC**  the impact upon fixed costs of a percentage change in the inflation level.
Inflation Scenarios

<table>
<thead>
<tr>
<th>Period 1</th>
<th>Period 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Inflation [0.5]</td>
<td></td>
</tr>
<tr>
<td>Inflation : 7%</td>
<td>Inflation : 7%</td>
</tr>
<tr>
<td>High Inflation [0.25]</td>
<td></td>
</tr>
<tr>
<td>Inflation : 12%</td>
<td>Inflation : 12%</td>
</tr>
<tr>
<td>Falling Inflation [0.25]</td>
<td></td>
</tr>
<tr>
<td>Inflation : 12%</td>
<td>Inflation : 10%</td>
</tr>
</tbody>
</table>

Economic Scenarios

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Buoyant [0.5]</td>
<td></td>
</tr>
<tr>
<td>Economic : 5%</td>
<td>Economic : 7%</td>
</tr>
<tr>
<td>Depressed [0.5]</td>
<td></td>
</tr>
<tr>
<td>Economic : 5%</td>
<td>Economic : 4%</td>
</tr>
</tbody>
</table>

Figure 6.1: Environmental Scenarios.
From equation 5.1, the impact on sales is

\[(1 + \beta_{ES} \times \text{ECONOMIC})(1 + \beta_{IS} \times \text{INFLATION})\]

and for fixed costs is

\[(1 + \beta_{IFC} \times \text{INFLATION})\]

The response curves corresponding to the interface variables are shown in figure 6.2. These are linear and correspond to the following values for the interface variables:

\[
\begin{align*}
\beta_{ES} & : 1.2 \\
\beta_{IS} & : 0.8 \\
\beta_{IFC} & : 1.0
\end{align*}
\]

The interface functions are:

\[
\begin{align*}
\text{SALES} & \text{ ARE } \text{SALES}(T-1) \times (1 + \text{SALESG}) \times (1 + \beta_{ES}\text{ECONOMIC}) \\
& \times (1 + \beta_{IS}\text{INFLATION}) & \text{(6.31)} \\
\text{FC} & \text{ ARE } \text{FC}(T-1) \times (1 + \text{FCG}) \times (1 + \beta_{IFC}\text{INFLATION}) & \text{(6.32)}
\end{align*}
\]

### 6.2.7 Decision Alternatives

The company wishes to consider an expansion strategy. This strategy is composed of three alternatives:

1. **no expansion, continue as at present (the null decision);**
2. **expand in period 1;**
3. **continue as at present until period 2, then expand.**

The expansion is going to impact directly the following variables:

1. **Sales.** At present these increase by 3% real in each period. In the period in which the expansion is made it is expected that sales will increase by 15%.
2. **Fixed assets.** The expansion involves an investment in fixed assets to boost sales capability. Typically fixed assets are replaced at a rate equal to 2% of the previous period's sales. If the expansion takes place, however, it will be necessary to increase the fixed assets by 25%.
Figure 6.2: Response Curves.
3. Fixed costs, which normally increase at 2% each period, will, in the period of expansion, increase by 8%.

The decision and its primary impacts are shown in figure 6.3. The secondary inputs, such as increased requirements for finance, are reflected automatically in the organisational model. All the impacts in this situation are incremental; the decision imposes no structural changes to the organisational model.

6.2.8 Policy Limitations

There are several financial policy requirements which must be met:

1. Debt-equity ratio is limited. Total debt may not exceed shareholders' funds.

   \[ TDEBT + OD + CRS < SCAP + RETPR \]  \hspace{1cm} (6.33)

2. Short-term debt is limited and may not exceed 50% of current assets.

   \[ OD + CRS < 50\% \times CA \]  \hspace{1cm} (6.34)

3. Dividends may not exceed 40% of accounting profit after tax.

   \[ \text{DIV} < 40\% \times \text{NPAT} \]  \hspace{1cm} (6.35)

4. Interest payments cannot exceed 75% of NPAT.

   \[ \text{INT} < 75\% \times \text{NPAT} \]  \hspace{1cm} (6.36)

5. Share issues are limited to 10% of the share capital in the previous period.

   \[ \text{NEWCAP} < 10\% \times \text{SCAP}(T-1) \]  \hspace{1cm} (6.37)

Additionally, it is required that cash cannot become negative (to provide unlimited finance at no cost), and that overdraft cannot become negative (thus providing income from interest received).

   \[ \text{CASH} > 0 \]  \hspace{1cm} (6.38)

   \[ \text{OD} > 0 \]  \hspace{1cm} (6.39)
<table>
<thead>
<tr>
<th>Period 1</th>
<th>Period 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Continue)</td>
<td>(Continue)</td>
</tr>
<tr>
<td>SALESG IS 3%</td>
<td>SALESG IS 3%</td>
</tr>
<tr>
<td>FCG IS 2%</td>
<td>FCG IS 2%</td>
</tr>
<tr>
<td>NEWFA IS 2%*SALES(T-1)</td>
<td>NEWFA IS 2%*SALES(T-1)</td>
</tr>
<tr>
<td>(Continue)</td>
<td>(Expand)</td>
</tr>
<tr>
<td>SALESG IS 3%</td>
<td>SALESG IS 15%</td>
</tr>
<tr>
<td>FCG IS 2%</td>
<td>FCG IS 8%</td>
</tr>
<tr>
<td>NEWFA IS 2%*SALES(T-1)</td>
<td>NEWFA IS 25%*FA(T-1)</td>
</tr>
<tr>
<td>(Expand)</td>
<td>(Continue)</td>
</tr>
<tr>
<td>SALESG IS 15%</td>
<td>SALESG IS 3%</td>
</tr>
<tr>
<td>FCG IS 8%</td>
<td>FCG IS 2%</td>
</tr>
<tr>
<td>NEWFA IS 25%*FA(T-1)</td>
<td>NEWFA IS 2%*SALES(T-1)</td>
</tr>
</tbody>
</table>

Figure 6.3: Possible Strategies and Their Impact.
6.2.9 Summary of Model

The above sections completely define the modelling problem independent of any solution technique. There are two categories of variables with no uniquely defined values:

1. those variables whose value is derived from a scenario, in this case sales and fixed costs. For deterministic simulation or optimisation the set of scenarios to use must be specified. Risk analysis considers all scenario sets in succession.

2. the decision variables, in this case new and repaid overdraft, new term debt and new share issues. In deterministic simulation values must be supplied for these, whereas optimisation chooses its own values according to the objective function. The risk analysis requires the values, from either simulation or optimisation, to be specified a priori.

The next sections discuss the formulations required to solve the model using either simulation or optimisation.

6.3 SOLUTION BY SIMULATION

The method of deterministic simulation when applied to a model such as this is well understood (Khoury and Nelson (1965), Boulden and Buffa (1970)). The implementation in this framework is, however, different from traditional methods. In most systems it is necessary to specify not only the logic but the order in which the model is to be solved. These are the so-called "procedural systems", and are derived from traditional computer programming languages.

In contrast to procedural systems are non-procedural systems, in which the system is itself able to establish the correct order of solution. This eliminates the need for complex control structures,

1. Examples include SIMPLAN, FPS, PLANCODE, FORESIGHT.
2. Examples include IFFS, CUFFS, as well as the framework developed in this study.
and makes modelling considerably easier. Typically such models are solved by evaluating all of the variables.

The solution method developed for this framework, however, only requires those portions of the model that are of immediate interest to be solved, thus improving modelling efficiency. The method, termed recursive substitution, was outlined in section 4.5.1. The following section illustrates this method.

6.3.1 An Example of Recursive Substitution

Consider the situation in which it is required to calculate the value of interest in period one. Assume that the scenarios "low inflation" and "depressed" have been selected for evaluation, along with the decision to expand in period one.

The definition of interest in period one, with parameter values inserted, is (from definition 6.19):-

\[ \text{INT}(1) \quad \text{IS} \quad 13\% \times \text{TDEBT}(1) + 10\% \times \text{OD}(1) \]

In order to calculate interest in period one it is necessary to calculate the value of both TDEBT and OD in period one. The former of these is (from 6.14):-

\[ \text{TDEBT}(1) \quad \text{IS} \quad \text{TDEBT}(0) + \text{NTDEBT}(1) - \text{RTDEBT}(1) \]

The value of TDEBT in period zero is known (see balance sheet in section 6.1), reducing the definition to the form

\[ \text{TDEBT}(1) \quad \text{IS} \quad 4 + \text{NTDEBT}(1) - \text{RTDEBT}(1). \]

No definition has been supplied for NTDEBT(1) as it is a decision variable. Therefore, at this point, the user must supply a value. Assume that a value of 2 is supplied, then

\[ \text{TDEBT}(1) \quad \text{IS} \quad 6 - \text{RTDEBT}(1). \]

The definition for RTDEBT(1) is (from 6.18):-

\[ \text{RTDEBT}(1) \quad \text{IS} \quad 10\% \times \text{TDEBT}(0), \]

which reduces to

\[ \text{RTDEBT}(1) \quad \text{IS} \quad 0.4. \]
Returning to the calculation of TDEBT and substituting this value gives

\[ \text{TDEBT}(1) \text{ is 5.6} \]

and substituting this back into the interest calculation gives

\[ \text{INT}(1) \text{ IS } 0.728 + 10\% \times \text{OD}(1). \]

The process continues in a similar manner for the calculation of the period one overdraft.

A calculation sequence by recursive substitution can be shown as a tree. The tree for the complete calculation for \( \text{INT}(1) \) is shown in figure 6.4.

The recursive substitution method can be used to solve any portion of a model which does not contain a set of simultaneous definitions. Such simultaneous systems can be detected by the method\(^3\) and solved by some other method (e.g. Gaussian elimination).

### 6.4 SOLUTION BY OPTIMISATION

The optimisation method, linear programming, requires that the model be formulated into a matrix, the structure of which is suitable for the optimisation algorithm. The matrix should be as small as possible (for efficiency), and should not contain any implicit constraints of the type discussed in section 5 A.3.

To achieve this, a variant of the recursive substitution technique was developed\(^4\). The method is termed recursive factorisation, and like recursive substitution, only considers those portions of the model which are relevant for the particular solution required.

The structure of the linear programme generated by recursive factorisation is:

- one row for the objective function, representing

---

3. A loop in the tree structure is indicative of a simultaneous relationship (see figure 4.11).
4. A discussion of the computer implementation of this technique is discussed in appendix A.
Figure 6.4: Recursive Substitution Sequence for Interest Calculation.
the entire model reduced to a single linear set of factors;
   - one row for each constraint in each time period;
   - one column for every decision variable encountered;
   - in certain circumstances, one extra column and row for simultaneous relationships.

Those model variables which do not have a corresponding column in the linear programming formulation may be solved by recursive substitution after the optimisation solution has been found.

6.4.1 An Example of Recursive Factorisation

Consider the situation in which it is required to minimise the value of interest in period one. As in section 6.3.1, assume that the scenarios "low inflation" and "depressed" have been selected for evaluation, along with the decision to expand in period one.

The definition for interest in period one is (from 6.19)

\[\text{INT}(1) \quad \text{IS} \quad 13\% \times \text{TDEBT}(1) + 10\% \times \text{OD}(1).\]

To calculate \text{INT}(1) it is first necessary to calculate \text{TDEBT}(1). The definition for this is (from 6.14):

\[\text{TDEBT}(1) \quad \text{IS} \quad \text{TDEBT}(0) + \text{NTDEBT}(1) - \text{RTDEBT}(1),\]

which reduces to

\[\text{TDEBT}(1) \quad \text{IS} \quad 4 + \text{NTDEBT}(1) - \text{RTDEBT}(1).\]

There is no definition for \text{NTDEBT}(1) so it becomes a decision variable, and is left in the calculation with a coefficient of 1. From section 6.3.1, \text{RTDEBT}(1) reduces to 0.4, leaving the partially evaluated definition for \text{TDEBT} as

\[\text{TDEBT}(1) \quad \text{IS} \quad 3.6 + 1 \times \text{NTDEBT}(1).\]

Substituting this back into the interest definition gives

\[\text{INT}(1) \quad \text{IS} \quad 0.468 + 0.13 \times \text{NTDEBT}(1) + 10\% \times \text{OD}(1).\]

5. A decision variable is one for which there is no definition.
6. This objective function is for illustration only and does not purport to be a suitable organisational objective.
It remains to evaluate overdraft in period one. This is given by (from 6.15):-

\[ \text{OD}(l) = \text{OD}(0) + \text{NOD}(l) - \text{ROD}(l) \]

The overdraft for period zero is known (see balance sheet, section 6.1), but both \( \text{NOD}(l) \) and \( \text{ROD}(l) \) are decision variables. The partially evaluated definition for period one overdraft is:-

\[ \text{OD}(l) = 3 + 1 \times \text{NOD}(l) + 1 \times \text{ROD}(l). \]

Substituting this back into the interest calculation results in the following objective function:-

\[
\text{Minimise } 0.768 + 0.13 \times \text{NTDEBT}(l) + 0.1 \times \text{NOD}(l) \\
+ 0.1 \times \text{ROD}(l)
\]

The process is then applied to all constraints in all time periods of the model (in this case two periods), with the exception of period zero\(^7\). This results in a matrix of 15 rows and 8 decision variables (see figure 6.5).

### 6.5 RISK ANALYSIS

Risk analysis requires that the model be "solved" for each possible combination of scenarios. For each combination, the variable values are calculated by recursive substitution. These values, in association with the probability of that particular combination of scenarios, enable the expected value, standard deviation, variance and range of the distribution to be calculated.

#### 6.5.1 An Example of Risk Analysis

In section 6.4.1 a linear programme was generated to minimise interest for the decision to expand in period one over the scenario combination of "low inflation" and "depressed". When solved, this

---

\(^7\) Period zero is used to define the initial data for the model, and hence is completely defined. Experience shows that frequently the constraints in period zero are broken a priori, so including them only forces infeasibility.
<table>
<thead>
<tr>
<th>Definition Number/Per</th>
<th>NTDEBT (1)</th>
<th>NTDEBT (2)</th>
<th>NOD (1)</th>
<th>NOD (2)</th>
<th>ROD (1)</th>
<th>ROD (2)</th>
<th>NEWCAP (1)</th>
<th>NEWCAP (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(6.33) 1</td>
<td>1.065</td>
<td></td>
<td>1.050</td>
<td></td>
<td>-1.050</td>
<td></td>
<td>-1.000</td>
<td>-1.72</td>
</tr>
<tr>
<td>(6.33) 2</td>
<td>1.024</td>
<td>1.065</td>
<td>1.100</td>
<td>1.050</td>
<td>-1.100</td>
<td>-1.050</td>
<td>-0.960</td>
<td>-3.72</td>
</tr>
<tr>
<td>(6.34) 1</td>
<td>-0.445</td>
<td>0.543</td>
<td></td>
<td></td>
<td>-0.543</td>
<td></td>
<td>-0.500</td>
<td>0.45</td>
</tr>
<tr>
<td>(6.34) 2</td>
<td>-0.368</td>
<td>-0.445</td>
<td>0.568</td>
<td>0.543</td>
<td>-0.568</td>
<td>-0.543</td>
<td>-0.490</td>
<td>-0.73</td>
</tr>
<tr>
<td>(6.35) 1</td>
<td>0.026</td>
<td>0.026</td>
<td>0.020</td>
<td>0.020</td>
<td>-0.020</td>
<td></td>
<td>0.040</td>
<td>-0.57</td>
</tr>
<tr>
<td>(6.35) 2</td>
<td>0.023</td>
<td>0.026</td>
<td>0.020</td>
<td>0.020</td>
<td>-0.020</td>
<td>-0.020</td>
<td>0.040</td>
<td>-0.57</td>
</tr>
<tr>
<td>(6.36) 1</td>
<td>0.179</td>
<td>0.138</td>
<td></td>
<td></td>
<td>-0.138</td>
<td></td>
<td></td>
<td>-0.57</td>
</tr>
<tr>
<td>(6.36) 2</td>
<td>0.161</td>
<td>0.179</td>
<td>0.138</td>
<td>0.138</td>
<td>-0.138</td>
<td>-0.138</td>
<td></td>
<td>-1.06</td>
</tr>
<tr>
<td>(6.37) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td>-0.95</td>
</tr>
<tr>
<td>(6.37) 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.010</td>
<td>1.000</td>
</tr>
<tr>
<td>(6.38) 1</td>
<td>0.890</td>
<td></td>
<td>0.915</td>
<td></td>
<td>-0.915</td>
<td></td>
<td>1.000</td>
<td>1.25</td>
</tr>
<tr>
<td>(6.38) 2</td>
<td>0.736</td>
<td>0.890</td>
<td>0.865</td>
<td>0.915</td>
<td>-0.865</td>
<td>-0.915</td>
<td>0.980</td>
<td>3.06</td>
</tr>
<tr>
<td>(6.39) 1</td>
<td></td>
<td>1.000</td>
<td></td>
<td></td>
<td>-1.000</td>
<td></td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>(6.39) 2</td>
<td></td>
<td>1.000</td>
<td>1.000</td>
<td></td>
<td>-1.000</td>
<td>-1.000</td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>Minimise</td>
<td>0.13</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+0.768</td>
</tr>
</tbody>
</table>

**Figure 6.5**

Linear Programme Tableau Generated by Recursive Factorisation
formulation gave the variable CASH a value of 0.8

An indication of the "riskiness" of cash in period one is given by:-

1. the distribution of CASH(1)
2. the probability that CASH(1) > 0.9

These can be calculated as follows:-

1. select the first chance future (scenario combination)
2. calculate the expressions and their probability
3. select the next chance future and return to
   step 2, until all possible chance futures have
   been calculated. At this point the cumulative
   probability must be one.

The first chance future is the combination of branches "low
inflation" and "buoyant". This has a probability of:-

\[ P(\text{low inflation}) \times P(\text{buoyant}) = 0.5 \times 0.5 \]
\[ = 0.25 \]

To calculate the value of CASH(1) the method of recursive substitu-
tion is used. Note however, that instead of the value of the decision
variables being supplied by the user, they are supplied by the solu-
tion to the previously run linear programme. This gives a value of
zero for CASH(1) in this future.

The expression CASH(1) > 0 is given a value of 0 if false, and 1 if
true. In this case the relationship is true, and so the value is 1.

Having calculated the expressions for this scenario combination, the
next combination is done, and so on until all combinations have been
evaluated. The complete set of calculations are shown in figure 6.6.

---

8. CASH is not in itself a decision variable, but its value
   depends upon those of decision variables, and hence can be
   calculated by recursive substitution after the optimal solution
   has been found.

9. The expression CASH(1) > 0 reads "CASH in period one is greater
   than or equal to zero".
<table>
<thead>
<tr>
<th>Scenario Combination</th>
<th>Probability ( P_i )</th>
<th>CASH(1)</th>
<th>CASH(1) &gt; 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>( P_i \times Value )</td>
<td>( P_i \times Value^2 )</td>
</tr>
<tr>
<td>Low inflation Buoyant</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low inflation Depressed</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High inflation Buoyant</td>
<td>0.125</td>
<td>-0.11</td>
<td>-0.0138</td>
</tr>
<tr>
<td>High inflation Depressed</td>
<td>0.125</td>
<td>-0.11</td>
<td>-0.0138</td>
</tr>
<tr>
<td>Falling inflation Buoyant</td>
<td>0.125</td>
<td>-0.11</td>
<td>-0.0138</td>
</tr>
<tr>
<td>Falling inflation Depressed</td>
<td>0.125</td>
<td>-0.11</td>
<td>-0.0138</td>
</tr>
<tr>
<td>Totals</td>
<td>1.00</td>
<td>-</td>
<td>-0.055</td>
</tr>
</tbody>
</table>

**CASH(1)**

- Expected Value \( = \sum P_i \times Value \) = -0.055
- Variance \( = \sum P_i \times Value^2 - (\sum P_i \times Value)^2 \) = 0.003
- Standard deviation = Variance \( = 0.055 \)
- Range is from -0.11 to 0

**CASH(1) > 0**

- Probability \( (CASH(1) > 0) = \sum P_i \times Value = 0.5 \)

**Figure 6.6: Summary of Risk Analysis Results**
6.6 REVIEW

In this chapter a hypothetic model has been constructed based upon the modelling framework developed in the previous two chapters. The model was a simplified but non-trivial example of one type of model that is likely to be built using this framework. Three types of solution technique, deterministic simulation, optimisation, and risk analysis, were then applied to the model.

This exercise shows:

1. that the framework as developed is mathematically valid;
2. that the calculations involved in solving even a simple model are of sufficient complexity that, if the framework is to be implemented in practice, the implementation must be computer-based.

The next chapter discusses some computer-related requirements for such a computer implementation, and the actual implementation is discussed in chapter eight.
Chapter Seven:
Requirements of a Computer-Based Modelling System.

A framework for the use of models as decision-making aids in organisations has been developed in the preceding chapters. This framework was based on the structure of the organisational decision processes. The framework was described mathematically, and was illustrated by a small, non-trivial example. This illustration showed that, if the modelling framework is to be used by management, it must be implemented as a computer-based system.

The use by management of computer-based tools, such as modelling systems, as an aid to decision-making is the centre of a field of study called decision support systems (DSS). These systems are specially tailored to the requirements of users who are not experienced in, or particularly knowledgeable about the details and workings of computers, but who only require their use as a tool or a means to an end. The nature of DSS is discussed in the first part of this chapter.

The lack of experience by their intended users imposes special requirements on the design of such systems. Certain of these requirements are fundamental to all DSS regardless of the intended application. Others are derived principally from the task which the DSS is to support, and as such apply only to specific DSS implementations. Both these types of requirement are discussed in the latter parts of this chapter.
7.1 DECISION SUPPORT SYSTEMS

The concept of decision support systems (DSS) is relatively new. It has emerged from the traditional areas of data processing and management information systems (MIS). Fundamental to DSS is the concept of a man-computer symbiosis: man (the decision-maker) provides the judgment and creative abilities while the computer supplies rapid calculation and information retrieval power. The combination provides a synergy effect, the net result being to augment the user's decision-making ability. Equally important is the idea that the computer acts only to support the decision-maker. It is never in control of the decision-making process, and is used only at the discretion of the decision-maker. (Keen and Scott Morton (1978), Sprague (1980)).

A taxonomy for DSS has been suggested by Alter (1977a). This is shown diagrammatically in figure 7.1. A DSS can range from a pure data retrieval system, where the raw data is accessed by request with no analysis whatsoever, to a system which suggests answers to a specific problem. A strategic model is in the simulation-suggestion portion of this taxonomy.

---

File drawer systems
Data analysis systems
Analysis information systems
Accounting models
Representation models
Optimisation models
Suggestion models

Data retrieval
Data analysis
Simulation
Suggestion

---

Figure 7.1: Taxonomy of Decision Support Systems.

(From Alter (1977a)).
Bonczek et al (1980b) have identified three principal components of a DSS. Two of these, the knowledge system and problem-processing system are directly related to Alter's taxonomy. The third, the language system, is the method by which the user (decision-maker) accesses the other two (i.e. the system's syntactic interface). The knowledge system contains information in the form of files or data bases about the decision-maker's problem domain, whereas the problem processing system converts input from the language system (i.e. problems) and data from the knowledge system into information that supports the decision process. In pure data retrieval systems (such as file drawer systems) there is at most a minimal problem processing system, whereas in model oriented DSS's the problem processing system is dominant (figure 7.2).

Sprague (1980) suggests that there are three discernable levels of technology associated with DSS. At the highest level is the specific DSS itself, which is an arrangement of hardware and software that allows a decision-maker to deal with a specific set of related problems. The next level is the DSS generator, which is the hardware and software that provides the capabilities to quickly and easily build a specific DSS. The lowest level, DSS tools, are those hardware and software elements that facilitate the development of either a specific DSS or a DSS generator.

The philosophy of DSS implies that such systems are interactive (i.e. they are used on-line and respond in a timely manner). Interaction is important in applications such as DSS because:¹

- answers to specific questions can be obtained more or less immediately;
- the concentration and attention of the user with respect to his problem are not interrupted by delays caused by waiting for output;
- more alternatives can be studied and models debugged and tested quickly and more easily;
- courses of action suggested by the system itself can be evaluated immediately.

¹. See for example, Alter (1977b)
The degree and scope of the interaction is determined not only by the language system but also (in some cases) by the problem-processing system.²

In many situations the ideal for DSS is to have the actual decision-maker using and interacting with the system. However, given the present state of the art of computer languages this is rarely achieved.³ In lieu of this there are generally two other patterns of use (Sprague (1980), Alter (1980)).

1. expert use through an intermediary. The decision-maker is remote from the (specific) DSS itself, and all communication between them is done via an expert intermediary. This person translates his perceptions of the decision-maker's problem into a format suitable for the DSS, and interprets the output for the decision-maker. The advantage of direct interaction with the system is largely lost to the decision-maker, although the interactive nature of the system allows the intermediary to function more efficiently. Additionally, the intermediary's perceptions of the problem may markedly differ from those of the decision-maker, thus distorting the decision process.

2. chauffeur use. The decision-maker interacts with the DSS directly, except that all input (and possibly translation of the output) to the system is handled by an expert chauffeur. The decision-maker and the chauffeur are both working with the DSS at the same time and in the same place so that the advantages of interaction are maintained, and the likelihood of potential problems arising through perception differences is minimised.

² For example, in building decision structures (see Leal (1976)).
³ Where CEO's, for example, do use a DSS directly they have generally had a technical background (Rockart and Treacy (1982) McKenney and Keen (1973)).
The type of use that is best will vary from situation to situation. However, as Alter (1977b) points out:

"... the key issue was not whether the user or manager could talk to a computer to get the answers he needed, but rather whether a combination of people, data, models and technical tools could provide these answers in a convenient, timely, and cost effective manner."

7.2 TECHNOLOGICAL REQUIREMENTS OF A DECISION SUPPORT SYSTEM

The intended applications and use (i.e. interactive by non-computer experts) of DSS imposes some general requirements on the technology of DSS. Boulden and Buffa (1970) have listed a set of requirements for on-line corporate models (see table 7.1). Although these were formulated before the concept of DSS evolved, they continue to form a valid basis for DSS.

![Diagram: Components and Users of DSS](image)

**Figure 7.2: Components and Users of DSS.**

4. For more detailed patterns of use see Alter (1980), Chapter 4.
Simplicity - The manager must understand the logic exercised by the computer and be able to verify the calculations when desired.

Secrecy - Confidential data and analyses must be protected. Data such as corporate profit plans and merger analyses are highly proprietary.

Conversational - The manager must be able to interact directly with the computer in his own language. An extensive programme must be available to guide the manager in asking meaningful questions with reasonable protection against operating errors.

Fast Response - The answer to one question normally suggests another question. The manager must be able to establish a meaningful dialogue with the computing system to facilitate the creative process of problem solving.

Management Control - The decision process must be under the control of the manager, not the computer. A manager should be able to change assumptions, alter logic, and specify exact output desired without resorting to reprogramming.

Accessibility - The system should be accessible from the manager's office or from any other point within his sphere of operation so that problems may be explored from a conference environment or during a business trip.

Data Availability - Current as well as historical and projected operating information must be available through the system. Accounting, financial, statistical, and mathematical tools should be readily available for analysing and transforming data.

Flexibility - The system must have the flexibility to explore a wide range of problems encountered by the manager. The system should be available for use by a number of staff and operating personnel with diverse interests without reprogramming.

Economy - The manager needs a system which will give results now and at a reasonable cost. He does not have two years to develop a system which may or may not function. Cost should be dependent on time usage of the system.

Table 7.1: Requirements of an On-Line, Time-Shared Computer System.

From: Boulden and Buffa (1970).
Figure 7.3: DSS Types, Technology and Users.
A further set of requirements has been outlined by Sterling (1975). Termed the "Stanley House criteria", these were established by interested parties during meetings in Canada. Although intended to apply mainly to the use of public and semi-public data bases, some of the criteria are applicable in the more narrow area of DSS. These have been shown in table 7.2.

In discussing the technological requirements for DSS, it is useful to structure them into the framework of Bonczek et al (1980b). Some criteria, such as system response time and system accessibility, pertain to the general environment in which the DSS operates, and not to the DSS itself. The requirements are summarised in figure 7.4.

7.2.1 Language System Requirements

7.2.1.1 Simplicity

The underlying reason for the necessity for a DSS to be simple (from the user's point of view) is that it will mostly be used on a casual basis by a naive user. Eason (1976) says of the naive user that:

![Diagram](image)

Figure 7.4: Technological Requirements of a Decision Support System.

5. "The naive computer user ... is a person who is not an expert in computer technology but who uses a computer system to assist him in the performance of a task", Eason (1976).
A1) The language of a system should be easy to understand.
A2) Transactions with a system should be courteous.
A3) A system should be quick to react.
A4) A system should respond quickly to users.
A5) A system should relieve the user of unnecessary chores.
A7) A system should include provision for correction.
E5) A system should treat with consideration all individuals who come into contact with it.

Table 7.2: "Stanley House Criteria".
(From Sterling (1975)).
1. "They need the computer for the performance of specific tasks and will evaluate its performance according to its ability to serve these tasks;

2. The interest and aspirations of most naive users lies in the work they do and not in the tools they are obliged to use. They seek only the necessary knowledge and skills to fulfil their task needs, and any time and effort devoted to studying issues of only peripheral importance is to be kept at a minimum;

3. At the same time as they seek to minimise the necessary learning, they also seek to minimise the time and effort devoted to using the computer system. It is a means to an end, not an end in itself."

He goes on to say that:-

"He [the naive user] needs access to a set of computer facilities commensurate with his information processing needs and these facilities must be easy to understand and easy to use".

Simplicity is also important because usage of the DSS is casual (i.e. infrequent) and generally voluntary (Alter (1980); p.178). Because the system is being used only infrequently by naive users there is a danger that, if the system is too complicated, they will tend to forget how to use it (Cuff (1980)). Additionally, as the system use will be largely voluntary, the users need to be motivated to use it, and this can be more easily achieved by a simple system than a complex one (Schilling (1979)).

There are several facets to the concept of simplicity:-

- ease of learning : the system should be such as to minimise the time necessary for a user to become sufficiently familiar with the system to use it with confidence (Eason (1976), Jarvis and Papaconstadapoulos (1978), Martin (1973), Newman (1978)). This is especially important in casual, voluntary systems such as DSS because potential users may not be prepared to undergo long, rigorous training programmes (Martin (1973)).

- ease of use : the system should be easy to use. (Eason (1976), Little (1970), Schilling (1979), Schneiderman (1980), Sterling (1975)). One
aspect of this is that as casual users
generally have limited keyboard experience,
the user time on the keyboard should be kept
as short as possible (Cuffs (1980), Martin
(1973)).

- ease of remembering: since use of the DSS will be
  infrequent, the system must be easy to remember
  (Schneiderman (1980)).

- ease of understanding: decision-makers cannot be
  expected to place a great degree of commitment
  on processes that they do not understand.
  Therefore a DSS must present these processes in
  such a manner as to facilitate their under-
  standing (Eason (1976)).

The achievement of "simplicity", as represented by these four
facts, can be realised by:-

1. consistent structure of the language (Gaines and
   Facey (1975), Schofield et al (1980), Schneiderman
   (1980)). The system and command terminology should
   be based on a common structure with no exceptions.

2. the use of familiar concepts (Barbosa and Hirko
   (1980), Keen and Wagner (1979)). The terminology,
   vocabulary and concepts imbedded in the system
   should be those of the targeted users of the DSS,
   and should avoid computer jargon.

7.2.1.2 Conversational

Decision support systems should be friendly and helpful so as to
encourage their use in voluntary situations. To be pleasant to use,
systems must avoid being:-

- patronising or rude. The message "VACUOUS DESCRIPTION"
  by the DEACON system (Martin (1973), Meadow
  (1970, p.120-122)) in reply to incorrect
  input is hardly likely to engender user
  goodwill.
- cryptic. Messages from the system should be helpful (Schofield et al (1980)). Thus, error messages such as:

   ERROR BD LINE 900

which is typical of many BASIC systems\(^6\) should be avoided, and a full explanation given instead.

- verbose. Because casual users will be inexperienced at the keyboard, input should not need to be too long and complex. Thus, while the input

   "SUBTRACT 850.0 FROM THE CASH FLOW OF PERIOD 1"

(Meader and Ness (1974)) is simple, easy to understand, and conversational, it is too verbose for most naive users to be bothered typing. At the other extreme, such as in airline reservation systems (Martin (1973)), keystrokes have been reduced by using short, difficult to remember mnemonics. An aspect of conversational systems for casual users is a tradeoff between short, easy to key, cryptic input and verbose, English-like statements.

7.2.1.3 User Control

The user must at all times feel that he controls the system, and not vice-versa (Barbosa (1980)). This has important ramifications on the way in which a system can be operated. There are three basic modes of input to a system (Keen (1980), Martin (1973), Newman (1978)):-

1. Question and answer dialogues. The user is asked if he requires a certain operation (such as a particular type of analysis or data retrieval), and he replies by typing either "Yes" or "No". This type of dialogue leaves control entirely with the computer, and denies the user the ability to alter the sequence of operations. Although very simple for the operator, it is very limited in scope.

---

6. This particular error message is from PRIME BASIC (cf BASIC Interpretive Language Guides, Revision A, May 1975, Prime Computer, Inc.).
2. Menu selection. The system displays a numbered list of possible choices, and the user enters the number of his preferred choice. This may lead to another menu being displayed and the process repeating itself. "Difficult (inflexible) for the user who knows exactly what area he likes, but has to go through the entire full menu selection process to get it" (Newman (1978)). "Simple for the operator" but "limited in scope" (Martin (1973)).

3. Command languages. The user specifies what he wants to do (such as plot a histogram or display a table). "Can be concise and precise" but "operator must be familiar with mnemonics and formats" (Martin (1973)).

The mode which provides the user with the greatest degree of control is that of command languages. However, it is also the only mode which requires the user to have prior knowledge of the system in order to use it.

A further aspect of control is the ability to escape from a task, such as when the user realizes that he has chosen an incorrect procedure (Schofield et al (1980)). The escape should return the user to a recognisable position in the system.

7.2.2 Knowledge System Requirements

7.2.2.1 Data Availability

The data that is needed to run the system should be readily available to the user. As a rule he should not have to key in long sequences of numbers (minimise keyboard time). The system should have the ability for the user to supply his own subjective estimates, especially in forecasting situations (Barbosa and Hirko (1980)).

Alter (1980; pp.127-132) has identified five types of data problems that can occur with a DSS:-
1. Data is not correct;
2. Data is not timely;
3. Data is not measured or indexed properly (e.g., it is only available in cents/litre and is required in dollars/tonne);
4. Too much data is needed;
5. Needed data simply doesn't exist.

In designing a DSS it is therefore important to consider what data is really required and will be available when the system is run. The ability to use subjective data could, in certain situations, help overcome some of the data problems.

7.2.2.2 Confidentiality

Clearly, data confidentiality can be of great importance, especially when sensitive data is being used, such as when a major investment or divestment decision is being analysed. Small (micro or mini) computers which are dedicated to a particular DSS have perhaps the ultimate in security: when the user has finished his analysis he can physically remove the files (if, for example, they are held on floppy disk or equivalent) and thus always remain in control of them. This is generally not possible on larger, centralised, time-sharing computers. However, with such systems there are usually several levels of security:-

1. Physical access to the computer itself is normally tightly controlled, so that only specially authorised staff have the right to enter the computer room.

2. Accessing the computer by means of a remote terminal is normally controlled by means of a system of user codes and/or passwords. These are checked by the computer before it allows the user access to any programmes or files.

3. Individual data files or programmes may have their own special passwords which must be specified before they can be accessed. Similarly individual fields or records within a file, or routines within a programme, may be protected by a password.
4. Data may only be available in aggregated or statistical form to certain categories of users. Thus the managing director may be able to inquire of the system a list of all those employees who earn more than a certain amount, whereas the response to the query by the personnel manager may be only the number of people in that category. The same query made by a person who has no need for such information could result in no information being supplied.

7.2.3 Problem Processing System Requirements

7.2.3.1 Flexibility

A DSS should be flexible enough to undertake a variety of tasks in the defined problem domain. The system should be able to adapt itself to the changing nature of the problem (Barbosa and Hirko (1980), Little (1970)). Both the data, parameters and the structure should be able to be easily changed to accommodate new information. Similarly it should be possible to use any solution processes that are available in a different sequence (Banczeck et al (1980a), Barbosa and Hirko (1980)). A system should also relieve the user of unnecessary chores (Sterling (1975)).

A DSS should not, however, be expected to perform outside its problem domain. If the problem is to find information then a system which specialises in information retrieval should be used (a data oriented system). If it is an investment problem then a specialised investment system should be used (possibly after and distinct from the information retrieval).

7.2.3.2 Feedback

Immediate, unambiguous feedback from the system is important to provide the user, especially a casual or naive user, with reassurance that the system is functioning as expected (Cuffs (1980)). Feedback

7. Note that information systems in this category must be protected against queries which attempt to narrow the possible response down from a statistic to an individual measure.
in the form of immediate response supports the user's control over the system. This is reinforced if the user is able to ascertain the state of the solution procedure at any time (Barbosa and Hirko (1980)).

7.2.4 Decision Support System Environmental Requirements

7.2.4.1 Response Time

Fast response is a desirable characteristic of an interactive system. Miller (1977) conducted experiments in which the measured user satisfaction with a system on the basis of (amongst others) a variable response rate. He states:

"This analysis allows an unequivocal view that users experiencing the high variability versions of the system expressed a significantly lower view of the system, its commands, its display, its speed and its overall utility than those experiencing the low variability versions of the system."

This is supported by Gaines and Facey (1975) who state that:

"The sheer variability of system response time without apparent cause was demoralising to the remote user."

This suggests that low variability in response times is an important attribute of an interactive system, probably more important than a low average response time.

Response time is not something that can be controlled easily on a large time-sharing system. Programme efficiency, in terms of core usage and process time, can speed up response times. The variability of response time, however, is a function of all programmes currently running on the computer and so is not within the control of the system designer. The use of a small dedicated system will, of course, give no variability in the response time for a specific task.

7.2.4.2 Protection

The user needs to be protected from unexpected system actions (Newman (1978)). This is especially true of new casual users who expect to make mistakes, but also expect the system to catch those mistakes (Cuffs (1980)).
"By virtue of their limited knowledge, naive users are "at risk" in the use of computer systems, especially when the system is on-line. If the system malfunctions or an error leads to an unusual response by the system, the user can experience the incident as traumatic and it can colour his attitude to the system for a considerable period."

Eason (1976).

7.2.4.3 Accessibility

Gaines and Facey (1975) state that "the importance of remote interactive users finding the system available when they try to access it cannot be over-emphasised". They argue that if the system is not available when required, especially if the time for which it is unavailable is not known, this will present the user with "disturbing uncertainty", which the user can avoid by ceasing to use the computer based system.

The terminal which is to be used for access should be located near (but not necessarily in) the user's work area. It should be in an area which offers privacy. If the terminal is a screen device, a printer should be available in the same area for hard copy output.

7.2.4.4 Economy

The system should provide results in an economical way. It should not be extravagant in terms of computer resources required (even although these are relatively inexpensive) because this may result in slower response times and reduced system availability.

7.3 USER REQUIREMENTS

The user of a DSS will be a unique individual with his own preferences, prejudices, cognitive style and abilities (McKenney and Keen (1974), Mason and Mitroff (1973)). Differences between the user's problem solving processing and those of the system may result in the system not being used, or its full potential not being realised (Keen and Scott Morton (1978); pp.73-77, Henderson and Nutt (1980)). Ideally, a computer system should be able to take account of these factors, adapting itself to the user as it learns about him. Such requirements as these are outside the scope of this study.
7.4 REQUIREMENTS OF A SPECIFIC DECISION SUPPORT SYSTEM

If a DSS is to be successful in a particular situation it must have certain abilities. These are largely determined by what the system will be expected to do, and as such are task specific. The application under study here is the support that such systems can give to certain aspects of the strategic processes within an organisation. This section studies the requirements that will be imposed upon a model oriented DSS in such applications.

7.4.1 Basic Requirements

The fundamental requirement of a model oriented system is the ability to define and solve models. Model definition requires, as a minimum, provision for:

1. defining the variables that are to be used in the model;
2. defining the relationships between the variables (i.e. the model logic).

Provision should be made for defining new variables and relationships at any point in the modelling process, so that the model can adapt to the user's perception of the problem. Thus, model definition is concerned primarily with specifying the knowledge system of the specific DSS.

A basic facility that should also be available is the ability to construct and combine internal and environmental models (see section 3.6). The rationale for having separate models (or components) is that they represent fundamentally different things. The internal model represents the operating structure of the organisation. As such it will alter if there is a structural change in the organisation, which will presumably be brought about by some kind of planned strategic decision8 (the interface model may be changed in similar

8. The model parameters (such as process conversion rates or realisable profit margins) may change to reflect operational factors (such as improved efficiencies). However operational changes should not affect the structure of a strategically-oriented model.
circumstances). The external model on the other hand will be constantly updated to reflect the most current information available on the state of the environment. In other words, the internal model represents factors which are largely controlled by the organisation, whereas the external model represents factors which are largely outside of the organisation's influence.

When a model has been defined, it must also be solved. This is the most basic function of the problem processing system. The mechanism for solving a model is to calculate a specific value for each variable, the value being derived from the relationship of the variable to the other variables in the model.

7.4.2 Task Requirements

The task for which the DSS is designed will clearly have a major impact on the structure and processing abilities of the DSS. The role of this DSS is to support aspects of the strategic processes within an organisation. These have been discussed in chapter three. The problem processing system of the DSS will have certain unique requirements imposed upon it by these processes. These in turn will affect the design of the language and knowledge systems. The requirements are summarised in figure 7.5.

7.4.2.1 Decision Recognition Routine

The uses of models as an aid to decision recognition fall into two broad categories (see section 3.2):

1. as a predictor of possible organisational performances in the future under various probable future scenarios (anticipatory mode);
2. as a control mechanism, where the actual results are significantly different from the predictions made by previous model runs (introspective mode).

The solution method that is necessary for these analyses is deterministic simulation.
<table>
<thead>
<tr>
<th>Decision Process</th>
<th>Decision Recognition</th>
<th>Decision Diagnosis</th>
<th>Evaln/Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anticipatory</td>
<td>Intro-</td>
<td>Anticipatory</td>
</tr>
<tr>
<td>Requirements</td>
<td>retrospective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Familiar Language</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Specify Sets of Futures</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario Comparison</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Prediction Comparison</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Update Scenarios</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update Models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision Options</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Optimisation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk Analysis</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.5: Summary of Requirements for DSS.
7.4.2.2 Diagnosis Routine

This routine is concerned with establishing the cause of the problem, and it can be invoked in one of three ways (section 3.3):

1. the control model predicts results which are in an unacceptable area (anticipatory);
2. the actual results experienced differ significantly from those that were predicted by the control model (introspective);
3. an event occurs which it is believed will have significant effects (either good or bad) on the organisation (reactive).

A type (1) result, being based on the latest available environmental forecasts, will probably have resulted from a potential future shift in the environment, causing it and the organisation to become out of step. It will be necessary to explore various possible strategies, probably using the model, in an effort to find a way to sidestep or to minimise the effects of this anticipated problem.

A type (2) result, based as it is on actual versus (previously) predicted results, indicates that the predicted model may not have in fact represented reality. A possible cause of this may have been unforeseen events in the environment. This can be checked by running the original model ex post with the actual environmental data. If there is a significant difference between these results and the actual results, then the model does not represent reality. As the model is based on the best understanding of the organisation, the implication is that this understanding is incomplete or incorrect.

It will be necessary to re-evaluate the assumptions that have been made with respect to the organisation (and its environmental interface) in an effort to explain the discrepancies. Similarly, if the environment did not act as expected, it would be desirable to establish why this was so and thus enhance the understanding of the environment.

A type (3) occurrence is a totally unforeseen event. The environmental scenarios, and possibly the environmental interface, will have to be altered to reflect the event. The model can then be run to evaluate the likely impact of this event on the organisation to see if problems
are likely to occur as a result. Again it would be useful to try and establish the cause of the event so as to enhance the understanding of the environment.

The actions described above suggest the following additional requirements:

1. to be able to alter and update an environmental scenario;
2. to be able to save a model and its results for subsequent ex post analysis;
3. to be able to update a model to reflect perceived changes in the organisation and its environmental interface.

Note that the use of the model is again largely predictive, so that deterministic simulation techniques are sufficient to run the model.

7.4.2.3 Evaluation-Choice Routine

The primary function of a model in this routine is to aid the evaluation of several different decision alternatives over an uncertain future. This suggests that a system should have the ability to:

1. express distinct decision alternatives or sets of alternatives, each one representing the impact on the structure or parameters of the model (the organisational component) of a particular decision option;
2. select any non-mutually-exclusive set of alternatives and investigate their effect on the model. Thus selecting any combination of decision options should automatically alter the structure of the model to represent the effects of implementing those options;
3. analyse the effects of the options simultaneously, so that interrelationships and synergistic aspects can be analysed;

9. Note that a decision will only effect the environment if it is proactive (section 2.5).
4. test the sensitivity of any particular set of alternatives over the possible set of environmental scenarios.

These requirements suggest the use of four separate modelling techniques:

1. deterministic simulation, to generate predictions based on a particular set of decision options in a specified future scenario;

2. simultaneous equation solver, to resolve systems of simultaneous equations that can occur naturally in many modelling situations;10

3. optimisation, to select that subset of alternatives which best meets some stated criteria;

4. risk analysis, to test the sensitivity of a specified set of decision options to changes in the environment, as represented by the environmental scenarios.

These techniques should be able to be used as and when required by the user. The model definition system should be such that models can be specified in such a way that they are technique independent.11

7.5 SUMMARY

This chapter has reviewed some facets of decision support systems (DSS) and their use as aids to the management decision-making process. Both the general requirements of such systems and the specific requirements of tasks supporting the strategic processes have been discussed.

The next chapter describes a computer system constructed to integrate the modelling framework (described in chapters four to six) with the decision support concepts reviewed in this chapter. Two applications of this modelling system are discussed in chapter nine.

10. Many models have in fact been based entirely on sets of simultaneous equations (see, for example, Francis and Rowell, (1978) and Warren and Shelton (1971)).

11. This assumes that the end models meet any specific technique requirements, such as linearity for optimisation, and known probabilities for the risk analysis.
APPENDIX TO CHAPTER SEVEN

A REVIEW OF EXISTING SYSTEMS

There are a number of commercially available modelling systems. Although these vary considerably in design and scope, they all purported to address the problem of strategic modelling. A cross-section of these systems\(^\text{12}\) was evaluated with respect to the technological and task requirements discussed in this chapter.

The systems ranged from being principally data-base oriented with some modelling power (FOCUS) to model oriented with limited data facilities (CUFFS). Most systems were procedural, whereas a few (CUFFS, IFPS, FCS-EPS) were non-procedural\(^\text{13}\). They were all interactive and had the basic modelling and reporting functions. Some included graphics capabilities, statistical analysis and forecasting routines (EMPIRE, EXPRESS, XSIM, REVEAL). One (REVEAL) included special capabilities based on fuzzy set theory.

"Ease of use" varies amongst systems. This is a very subjective criterion based not only on the language, but also on the command structure and the philosophy and architecture of the system (for example, compiler versus interpreter, procedural versus non-procedural, incremental versus holistic etc.). Model statements from some of the systems are shown in table 7A.2. This illustrates how "ease of learning" and "ease of understanding" can be influenced by the naturalness of the language structure.

12. The systems and their suppliers are listed in table 7A.1. Note that both the quantity and the quality of information available differed between systems, making a rigorous evaluation impossible. Systems such as MAPS (Ross Systems, Inc.), FORESIGHT (Uniting Computer Services) and BPS (Burroughs Corporation) were not included as they are principally budget oriented. Other systems, such as Prosper (ICL) and Plancode (IBM) were obsolete or insufficiently differentiated from other systems. A detailed discussion of these systems is given in Grimyer and Wooller (1975)

13. "Procedural languages tell the computer what to do - open a file, get a piece of data, add a number (procedures). Non-procedural languages tell the computer what the user wants - a list of late payments, analysis of sales results - and the software figures out how to produce it" - Dyson (1982).
<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUFFS</td>
<td>CUFFS Planning &amp; Models, Ltd, 44E 84th Street, Suite 10c, New York, NY 10028.</td>
</tr>
<tr>
<td>EMPIRE</td>
<td>Applied Data Research, Inc, Princeton, NJ 08540.</td>
</tr>
<tr>
<td>EXPRESS</td>
<td>Management Decision Systems, Inc, 200 Fifth Avenue, Waltham, Massachusetts 02254.</td>
</tr>
<tr>
<td>FCS-EPS</td>
<td>EFS Consultants, 35 Soho Square, London Wlv 5DG.</td>
</tr>
<tr>
<td>FINAR</td>
<td>Scicon Computer Services Ltd, Brick Close Kiln Farm, Milton Keynes MK11 3EJ.</td>
</tr>
<tr>
<td>FOCUS</td>
<td>Information Builders, Inc, 1250 Broadway, New York, NY 10001.</td>
</tr>
<tr>
<td>FPS</td>
<td>Scientific Time Sharing Corp, 7316 Wisconsin Avenue, Bethesda, Maryland 20014.</td>
</tr>
<tr>
<td>IFPS</td>
<td>EXECUCOM Systems Corporation, 3409 Executive Center Drive, Austin, Texas 78731.</td>
</tr>
<tr>
<td>REVEAL</td>
<td>Decision Products, Inc, 2700 Augustine Drive, Suite 155, Santa Clara, California 95051.</td>
</tr>
<tr>
<td>SIMPLAN</td>
<td>Social Systems. Inc, P.O. Box 2809, Chapel Hill, NC 27514.</td>
</tr>
<tr>
<td>XSIM</td>
<td>Interactive Data Corporation, 1033 Massachusetts Avenue, Cambridge, MA 02138.</td>
</tr>
</tbody>
</table>

Table 7A.1: Systems and Suppliers.
Apart from the absolutely basic requirements (such as the ability to
define a model), the specialised task requirements were generally not
met in any of the systems. No system provided special mechanisms to:-

1. specify, use and update sets of scenarios;
2. specify and model the impact of different
decision options;
3. compare the output from different model runs
(either previous runs or runs based on different
scenario or decision options).

It would be possible to circumvent the lack of scenario or decision
structures by using separately prepared "sub-models" or alternative
data sets, and loading these into the main model. This is at best
a tedious and inefficient method, and is not conducive to a DSS
environment aimed at naive and casual users. It would be particularly
cumbersome in procedural systems.

The systems all used deterministic simulation as the basic means of
solving a model. Most systems supplied risk analysis in the form of
Monte-Carlo simulation. IFPS has a separate optimisation capability
(IFPOS). Although IFPOS uses the same input formats (augmented by
some extra directives) as IFPS, the two systems are not integrated
which makes optimisation an unnecessarily difficult method to use.

<table>
<thead>
<tr>
<th>Model</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPS</td>
<td>80 EQ CUM L65</td>
</tr>
<tr>
<td>FCS-EPS</td>
<td>80 'RETEARN' = ('RETEARN')LAGI + 'PROFIT'</td>
</tr>
<tr>
<td>REVEAL</td>
<td>80 RETEARN = OFFSET(RETEARN,-1) + PROFIT</td>
</tr>
<tr>
<td>SIMPLAN</td>
<td>80 RETEARN = RETEARN(-1) + PROFIT</td>
</tr>
<tr>
<td>CUFFS</td>
<td>#80 RETEARN = PREV(RETEARN) + PROFIT</td>
</tr>
<tr>
<td>IFPS</td>
<td>80 RETEARN = PREVIOUS RETEARN + PROFIT</td>
</tr>
</tbody>
</table>

Table 7A.2
Model Statements

The statements represent the retained earnings
calculation (the previous period's retained earnings
plus this period's profit) in various modelling
languages. Note that for FPS, the profit calculation
is done in statement number 65.
Chapter Eight:
The Implementation of a Computer-Based Modelling System.

This chapter discusses the implementation of "Stratagem", a decision support system generator which incorporates the modelling framework developed in this research. The implementation is based on the decision support concepts described in the previous chapter.

The first part of this chapter gives a brief overview of the system philosophy and implementation. This is followed by a description of how the system meets the requirements of user-oriented computer systems. Two applications of the system are discussed in the next chapter.

8.1 SYSTEM OVERVIEW

A general computer-based system designed to embody the modelling framework developed in this research must of necessity be large and complex. This section provides a brief overview of the system and the reasons it was developed as it was. A discussion of some of the more novel computer science aspects of the implementation is given in Appendix A. A complete description of the system in the form of a user reference guide is given in Appendix B.
8.1.1 Implementation Philosophy

Stratagem was developed to make the modelling framework readily available to the non-specialised user. Without the aid of an easy-to-use support system such as Stratagem, the modelling framework would be too complex to be of practical use.

Many aspects of the framework can be exploited in the system design to facilitate its use. One such consideration is that each variable in a model can be defined by one and only one function. Furthermore, the framework does not specify that the functions must be solved in any particular order. These points have the following ramifications on the system implementation:

1) each function is uniquely identifiable by, and hence may be referenced by, its subject variable (i.e. the variable which it defines);

2) because the order of the functions is irrelevant, there is no need to number them.

These two points mean that the system can be non-procedural, so that it and not the user can determine the required order of solution. A corollary of this is that there is no need for artificial control structures, such as "GO TO" statements and iteration loops. All these factors serve to make the system significantly easier to use.

An aspect of the non-procedurality is that the system can be implemented as an incremental compiler. This means that as each statement is entered by the user it is checked and converted (compiled) into a form which can be used by the system. Thus not only is the user given instant feedback if an error occurs, but there is no need for a special set of instructions to initiate and control the compilation. Additionally, each function is available for use immediately it is entered, so that a model may be easily changed.
8.1.2 System Structure

The structure of Stratagem is shown in figure 8.1. The main part of the system is contained in the component labelled "Modelling System (1)". This is composed of several subsystems which are discussed in the next section.

The system can be operated in either batch or interactive mode. In batch mode the system reads a sequential file (card or disc file (2) in figure 8.1) of input statements and processes them one at a time. In interactive mode a special co-operating process (3) is initiated to act as a buffer between the user (4) and the modelling system (see section 8.1.8). The input formats expected by the system are the same regardless of whether batch or interactive mode is used.

Previously created disc files (5) can be accessed by Stratagem. Similarly it is possible to direct the system to save models on disc for use in later sessions.

A record of all input given to the system and all output generated by the system is kept in a log file (6). This is automatically printed at the end of each session so that a record is available of all the session's transactions.

An additional print file (7) is available. Prefixing any input to the system by the keyword "PRINT" will direct to this file any output generated as a response to the input. This is especially useful in preparing hard copies of reports. Alternatively, such output can be directed to a remote printing device (if available). In this manner copies of reports are available instantly.

---

1. When initiated, the system automatically detects whether it is operating in batch or interactive mode.
Figure 8.1: Structure of Stratagem.
8.1.3 Composition of System

Stratagem can be viewed as being composed of five integrated subsystems, linked by a simple and consistent language structure. (see figure 8.2). The object of the language is to permit maximum use of the system with as few commands as possible, and also to provide a vocabulary with which the user is familiar. To this end the vocabulary (but not the grammar) is completely user definable.

The central component of the system is the kernel (cf section 4.1). This contains the solution matrix and the means for defining and solving the model functions (i.e. the incremental compiler and interpreter). Housekeeping routines, such as saving and loading models, also reside here.

When the system is run interactively, all input from the user is handled by means of the input/output monitor. This module, run as a separate process from the remainder of the system, allows the user to queue up input in advance of processing and thus acts as a buffer between him and the rest of the system. It is discussed in more detail in section 8.1.8.

The report subsystem provides facilities for the specification and preparation of specialist user reports.

The scenario/decision subsystem provides the management capabilities for these structures. These include routines for compiling the structures, for using them in evaluating the model, and for risk analysis.

The optimisation subsystem is a set of specialist routines that formulate a model into a linear programming matrix. This matrix is then passed to a separate mathematical programming system. Processing of the modelling system is suspended until the optimisation is completed, at which point the results are read back into the modelling system, and made available for further analysis.

---

Figure 8.2: Basic System Structure.
3.1.4 Basic System Components

The system is centred around a set of data structures which correspond to the principal components of the modelling framework. The components are:

1) Variables: These reference the elements of the solution matrix (section 4.1). Each row of the matrix is represented by a single variable, and each element in the row corresponds to the variable in a particular time period. Any element in the solution matrix can be directly referenced by using the appropriate time period (column) as a subscript to the variable (row).

2) Constraints: These are mainly used to specify the feasible region in optimisation, although they may also be used to test conditions under simulations. They are represented by rows of the solution matrix in the same way as variables.

3) Reports: A report is a format for output specification. It is used for preparing specialised output, such as financial statements.

4) Scenarios: A scenario is used to refer to a set of probabilistic futures, as discussed in section 4.2.

5) Futures: A future is a set of definitions that describe a (probabilistic) future state of the environment.

6) Decisions: These are used to refer to sets of mutually-exclusive courses of action (options) which the decision maker has open to him (see section 4.3).
7) Options: An option is a set of functions which describes the impact of a particular course of action.

Associated with each component is the following set of attributes.

1) Name: This defines a unique identity for a particular element. Each element is referenced by its name.

2) Type: This specifies the type of component (e.g. variable, report). The type determines the context in which an element may be referenced, and the action taken when the element is referenced.

3) Description: This is used to describe the element and is primarily for documentation.

4) Value: This represents the status of the element, and its nature depends upon the type attribute. For example, for a constraint it can be "true", "false", or "undefined", whereas for a report it is a display (or listing) of the report. (see figure 8.3 for possible values)

5) Definition: This usually defines the mechanism by which the value for the element is calculated. For scenarios, decisions, futures and options, it describes the members of the appropriate set. The structure of the definition for each type must follow certain guidelines (see Appendix B).

Of all the component types, that of variable is the most fundamental. All other types use this as a basic building block. A complete list of model components may be generated by use of the "DOCUMENT" statement (see Appendix B).
<table>
<thead>
<tr>
<th>ELEMENT TYPE</th>
<th>POSSIBLE VALUES</th>
<th>MECHANISM FOR DETERMINING VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>Real numbers, Undefined</td>
<td>Definition Attribute</td>
</tr>
<tr>
<td>Constraint</td>
<td>True, False, Undefined</td>
<td>Definition Attribute</td>
</tr>
<tr>
<td>Report</td>
<td>Display or Printout</td>
<td>Definition Attribute</td>
</tr>
<tr>
<td>Scenario</td>
<td>Selected, Ignored^1</td>
<td>Select and Ignore Statement</td>
</tr>
<tr>
<td>Future</td>
<td>Selected, Inactive^2</td>
<td>Select Statement</td>
</tr>
<tr>
<td>Decision</td>
<td>Selected, Ignored</td>
<td>Select and Ignore Statement</td>
</tr>
<tr>
<td>Option</td>
<td>Selected, Inactive</td>
<td>Select Statement</td>
</tr>
</tbody>
</table>

Note:  
1) Scenarios and Decisions may be ignored so that their effects are not reflected in the model.  
2) Only one future in each scenario, and one option in each decision may be selected at any one time, so the others become inactive.

Figure 8.3: Range of, and Mechanism for Determining, the Attribute "Value".
8.1.5 Element Declaration

The name and type of an element must be declared to the system before the element can be used. Attempts to use elements which have not been declared will cause an error message to be displayed.

In practice it is quite feasible to deduce the type of most undeclared elements by the context on which they are first used (such as in Fortran or IFPS). Although this saves the user an extra step, it has one major drawback: names which are misspelt will be treated as different elements. Such spelling errors are often extremely difficult to detect, making model debugging unnecessarily difficult.\(^3\)

The forced declaration of elements has two major advantages in addition to overcoming the misspelling problem:

1) it imposes a discipline on the modeller, forcing him to specify in advance the elements which he intends to use, and thus helping to define the scope of the model;

2) it provides a useful reference for documentation purposes.

The type of an element (i.e. report, future etc.) dictates the usages to which the element can be put. The system has a degree of knowledge about how each element type can be used, thus reducing the possibility that they will be used out of their intended context.

The name that is given to the element when it is declared is the main method of referencing that item. Although each name must be unique (i.e. refer to only one element), the use of synonyms allows more than one name to refer to the same item. In this way, model users can build their own, individualised vocabularies.

---

3. A very common example of this is when the digit zero is used instead of the letter "O", e.g. J0 instead of JO.
8.1.6 Element Definitions

The nature of the definition attribute depends upon the element's type. These are summarised in figure 8.4.

The set of all variable definitions defines the logic of the model "kernel" (see section 4.1). These definitions may be altered by uncertain events in the environment (as represented by "futures"), and by planned changes (as represented by "options").

The constraint definitions define the allowable region in which the values for the variables may lie. In simulation these are not restrictive, but when optimising they define the feasible region for the solution, and hence are binding.

The report definition specifies the layout of a specialised report. At its simplest it consists of a list of variables and/or constraints which are to be output. Special formatting directives may also be included in a report.

8.1.7 Attribute Interrogation

The status of the attributes of any model element may be interrogated at any time. There are two types of attribute interrogation possible. The first of these displays all the attributes for the item except for value. It is invoked by entering the name of the element followed by a question mark,

   e.g. REVENUE?

The system responds by displaying the description, definition and type of the element.

The value is interrogated by entering the name of the element followed by a semicolon,

   e.g. REVENUE;
<table>
<thead>
<tr>
<th>ELEMENT TYPE</th>
<th>STRUCTURE OF DEFINITION</th>
<th>COMPOSITION OF DEFINITION</th>
<th>DEFINITION SPECIFIES:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>Function</td>
<td>Variables, Constants</td>
<td>Value</td>
</tr>
<tr>
<td>Constraint</td>
<td>Function</td>
<td>Variables, Constants</td>
<td>Value</td>
</tr>
<tr>
<td>Report</td>
<td>List</td>
<td>Variables, Constraints</td>
<td>Report contents</td>
</tr>
<tr>
<td>Scenario</td>
<td>Set</td>
<td>Futures</td>
<td>Possible future states of world</td>
</tr>
<tr>
<td>Future</td>
<td>Set</td>
<td>Variable/Constraint Definitions</td>
<td>Impact of state of world</td>
</tr>
<tr>
<td>Decision</td>
<td>Set</td>
<td>Options</td>
<td>Decision alternatives</td>
</tr>
<tr>
<td>Option</td>
<td>Set</td>
<td>Variable/Constraint Definitions</td>
<td>Impact of decision alternative</td>
</tr>
</tbody>
</table>

*Figure 8.4: The Structure and Function of Definitions.*
This causes the system to calculate and display the value for the element. For an element of type "variable" or "constraint", a set of values is displayed, representing the value of the element in each time period (i.e. the row from the solution matrix that corresponds to the element). For elements of type "report", the values of the variables and constraints that constitute the report are displayed in the format specified. The range of values that each type may have is shown in figure 8.3.

This facility of attribute interrogation is in itself sufficient to run most pre-specified models. Thus it provides a simple but adequate solution mechanism for the naive user, obviating the need for the complex set of commands which is typically required in modelling systems.

8.1.8 Input-Output Monitor

The input-output monitor is a special asynchronous process that runs concurrently with the remainder of the modelling system when the system is being used interactively. It has several purposes:

1) to act as a buffer between the modelling system and the user. This was intended to give the user additional control over the modelling process, and to provide him with the ability to interrogate the status of the modelling system at any time. It was expected that such control would be extremely useful when the system was involved in long calculations, such as in optimisation, risk analysis, and when solving large models.

2) to allow the user to queue multiple inputs. The user may wish to enter additional commands while the system is processing some previous input. The monitor can queue these until the system is ready to accept them.
3) to provide for alternate forms of input, such as menu or dialogue. Although these modes are seen as being less satisfactory than the command mode employed (section 7.2.1), it was none the less felt that provision should be made for them in case they prove desirable at some future stage.

4) to provide special facilities for formatting the output onto the user's terminal. The output could, for example, be displayed as pages of information, with the user having the ability to return to previous pages.

Because the input-output monitor is geared specifically for interactive users, it is not available under batch mode.

8.2 DECISION SUPPORT SYSTEM IMPLEMENTATION

Stratagem is a decision support system (DSS) generator whose function is to aid the construction of models which will support aspects of the strategic decision making processes in an organisation. The reason for developing Stratagem was to make accessible to the user the modelling framework developed in chapters three to five. This can only be achieved by following the precepts of DSS which were discussed in the previous chapter. This section discusses how Stratagem fulfils these DSS requirements.

8.2.1 Language System Requirements

The language system in a DSS is the interface between the user and the knowledge and problem processing components of the DSS. (see section 7.1). Because of this, the language system is highly visible to the user, and hence will be a major factor in the acceptability of the system to the user.

4. When the system was developed the specialised terminals required to implement these facilities were not available, and as such this feature was not implemented.
8.2.1.1 Simplicity

In section 7.2.1.1, "simplicity" was defined as having four facets: ease of learning, ease of use, ease of remembering, and ease of understanding. It was suggested that these could be achieved by basing the language on familiar concepts and a consistent structure.

The following features of Stratagem contribute towards simplicity:

1) the use of a user-defined vocabulary. Apart from a minimum of predefined key-words, the user is responsible for defining his own vocabulary. In this way he will be familiar with the terminology used which will help him to remember it. Additionally the vocabulary can be extended at any time by the use of synonyms, so that the user can, if desired, respecify the predefined key-words to his own satisfaction. This also means that different users can refer to the same model, each using his own preferred vocabulary.

2) the use of simple structures. The system is structured around the seven components discussed in section 8.1.4. To use the system successfully it is sufficient only to know how the variable data structure is specified. To aid understanding of this, all variable definitions use the same form of arithmetic expression as that used in the Fortran and Basic programming languages. While users may not be familiar with these per se, they will almost certainly be familiar with the simple algebraic form on which the expression structure is based.

As the user becomes more experienced with the system and gains confidence, he can start using the more complex structures such as reports and scenarios.
3) **language consistency.** Every effort has been made to keep the language and its actions consistent throughout the system. All commands have a simple but meaningful syntax, and may be used at any point in the modelling process. Similarly the concept of interrogation (section 8.1.7) is applicable throughout the system.

4) **the non-procedural orientation of the system.** This removes the need for artificial control structures, such as "GO TO" statements and iteration loops. The resulting reduction in complexity not only makes the language easier to learn and to use, but also speeds up the model building process. A significant side benefit is that because the order of the model logic (i.e. the variable definitions) is immaterial, no special modelling editor is required.

5) **the use of an incremental compiler.** Because each statement is automatically compiled as it is entered, there is no need to have separate "COMPILE" and "SOLVE" commands. This reduces the complexity of the language and simplifies and speeds up the model building process.

### 8.2.1.2 Conversational

The incremental compiler design enhances the conversational aspects of the system. As each statement is input it is checked so that the user is notified immediately if an error is detected. All error messages are given in full English with a description of the error and where it occurred, so that the meaning is quite clear.

Wherever possible, input to the system is in simple, English-like phrases. There is always a tradeoff in this approach between naturalness of language and verbosity. The ability to specify synonyms provides the user with the capability of shortening the input strings if desired.
8.2.1.3 User Control

The use of the system is based on a command language (section 7.2.1.3), so that the user is in control at all times. Because most of the commands are based on the variable names, they will be in the user's own vocabulary, thus reducing the learning effort required.

The input-output monitor (section 8.1.8) was also intended to enhance user control. Because it is implemented as an asynchronous process, it is technically possible for it to interrogate or interrupt the main modelling system. Although these features were not fully implemented, they would have provided the user with an extra level of control. For example, the user would be able to halt a long solution sequence (such as risk analysis) should he change his mind half way through. At present he must wait until the analysis is complete.

8.2.2 Knowledge System Requirements

Stratagem is a model oriented DSS generator, and hence the knowledge system requirements (in terms of data) are limited (section 7.1). Furthermore, the aim of the system is to support strategic decision making. Because the data required for this level of decision making is highly aggregated (see for example Lorange and Vancil, (1977; pp 265-266)), there will not be a large data requirement. Therefore it is not necessary for this type of system to interface directly into the "corporate data base", even if such an information resource exists.

8.2.3 Problem Processing System

Stratagem provides a variety of mechanisms to aid in the construction and solution of models. The non-procedural orientation coupled with the incremental compiler makes it easy to build and update a model. A variety of solution techniques are available to solve a model when it is constructed (see section 8.2.5).
The system attempts to relieve the user of as many chores as possible by having a lot of built in "intelligence". For example the system is able to work out the correct order in which to solve the model equations, and it also automatically detects and solves any linear simultaneous equations which are present in the model. Additionally the system can operate as a calculator (in much the same way as Basic), so that ad hoc calculations can be performed on the model variables at the terminal.

8.2.4 Specific DSS Requirements

The requirements for a specific decision support system were discussed in section 7.4. The most basic of these requirements was the ability to define the variables used in the model, and the logic of the model. How Stratagem fulfils these basic requirements was discussed in section 8.1. The degree to which Stratagem meets the other requirements is discussed below.

8.2.4.1 Sets of Scenarios

The mathematical structure for the scenarios was developed in section 4.2. A scenario is composed of a set of "futures", each future having an assigned probability of occurrence. A future in turn describes the impact on the model that will result from a particular combination of events occurring. The impact is represented by a set of functions.

Scenarios are implemented in Stratagem by the use of two basic data structures, "scenarios" and "futures" (see section 8.1.4). As with all Stratagem data structures, each scenario and future in the model is given a unique name by the user for reference purposes. Two types of definition (function) can be specified in a future:

1) a structural change, where an entirely new definition is specified for a variable. When the scenario is being used, this new definition replaces the definition for the variable in the main model. This type of definition is equivalent to functions of the
form shown in equation 4.19.

2) a change in the level of the variable, where the value of the variable as calculated elsewhere in the model is increased or decreased by a certain amount. This form of definition gives rise to functions of the type shown in equation 4.40.

An example of a scenario is shown in figure 8.5.

A particular future is selected for use by the "SELECT" instruction. When a future is selected, the model logic is automatically adjusted to represent the original model as impacted by the future (i.e. the model as given by equation 4.21). Using the "SELECT" instruction, it is possible for the user to run quickly through a variety of likely scenario combinations without needing to alter any of the model logic. In a similar manner, the effect of a scenario can be (temporarily) removed from the model by the use of the "IGNORE" instruction.

8.2.4.2 Display Results in a Familiar Form

Two modes of displaying the results of a modelling run have been provided in Stratagem. The first of these is generated in response to an interrogation of the value attribute for a set of variables or constraints. The response is a display of the values arranged in a simple table.

The second method is that of a user specified report. A user may specify named lists of variables and constraints, along with optional formatting requirements. Each list defines a report which is one of the Stratagem data structures. By interrogating the value attribute of a report, the contents of the report are displayed at the terminal. This is useful for defining output in the form of financial statements.
SCENARIO NEWCOMPETITION "NEW COMPETITION IN MARKET PLACE";
FUTURE NEWCOMPETITOR "NEW COMPETITOR";
  PROBABILITY IS 95%;
  SALES ARE 90% * FORECASTSALES;
  INCREASE UNITCOSTS BY 4% * UNITCOSTS(T-1);
ENDFUTURE;
FUTURE NONEWCOMPETITOR "NO NEW COMPETITOR";
  PROBABILITY IS 5%;
  SALES ARE FORECASTSALES;
ENDFUTURE;
ENDSCENARIO;

Figure 8.5: Example of a Stratagem Scenario.
A method of output which would have been desirable from a user point of view was simple graphics. This would give the user a visual representation of trends of variables over time and the relationships between specified variables. Implementation of this type of output has been deferred for future development.

8.2.4.3 Comparison of Results

When comparing the results from actual versus predicted futures, or evaluating different decision policies, it would be helpful to have some mechanism for comparing the results from different runs. Although this was not fully implemented in Stratagem, the proposed mechanism was as follows:

1) Any report could be saved on disc for printing at a later session by an instruction in the following format:

   `<reportname> AS <savedname> ;`

   where `<reportname>` is the name of the report
   and `<savedname>` is the title of the disc file in which the report is saved.

2) A subsequent model could be compared with the saved report by an instruction of the form:

   `COMPARE WITH <savedname> ;`

   The comparison would be presented in the same format as the saved report, except that both the data contained in the original report and the current data would be displayed side by side, with options for unit and percentage variance.

This form of comparison report, although simple in concept and relatively easy to implement, does not appear to be available on any modelling system currently available. Yet for highlighting the differences between alternative possible courses of action, it would appear to be a tool that is both easy to use and extremely useful for the decision maker.
8.2.4.4 Updating and Saving Models

The non-procedural implementation of the modelling system means that it is easy to update or alter a model. For example to change a variable definition it is necessary only to enter the revised definition. In other modelling systems, such as IFPS or SIMPLAN, it is necessary to go into a special edit mode, find the line where the definition occurs in the model, change the definition, and then recompile or solve the model. The proper implementation of a non-procedural system eliminates the need for an editor.

A model may be saved on disc for use in a later session by means of the "SAVE" command, the format of which is:

```
SAVE AS <filename> ;
```

where `<filename>` is the name of the disc file in which to save the model

The model is saved in symbolic form.

8.2.4.5 Decision Alternatives

Decisions and options, as described in section 4.3, have been implemented in Stratagem as basic data structures (see section 8.1.4). Because of their similarity with scenarios and futures (section 4.4), they have been implemented in the same fashion, except that they are non-probabilistic. Thus an option may be selected in exactly the same way as a future. When an option is selected, the model is automatically altered to reflect the impact of that option.

8.2.4.6 Analyse the Effects of Options Simultaneously

It was suggested in section 7.4.2.3 that it would be useful in the evaluation-choice routine of the decision making process to be able to consider all decision options simultaneously. In this way interrelationships between strategies can be considered from a synoptic viewpoint.
Since the modelling framework used in Stratagem is derived from the Gear-Lockett (G-L) model (chapter five), it is appropriate to use their method of project selection for this. The method, based on zero-one integer programming, is described in detail in Gear and Lockett (1973). An algorithm for automatically formulating a zero-one mixed integer programme for the general case of the G-L model is given in Cowie (1977). This demonstrates that it is both feasible and practical to use the G-L framework.

The formulation developed here, however, is an extension of the G-L model. Not only does it permit changes in the level of variables (as in equation 4.40) to appear on the branches of the trees (i.e. the futures and options), as in the G-L model, but it also allows the complete respecification of variable values (as in equation 4.19). If either of these types of equation involve a decision variable (such as level of debt), the result is non-linear and hence cannot be formulated as a zero-one problem. The non-linearity occurs because it is necessary to multiply the decision variable by the accept-reject (zero-one) variable. A further non-linearity arises in situations where the selection of a particular option imposes further constraints on the model.\(^5\)

Apart from the two situations described above, it is possible to map the model into a zero-one mixed integer programme of a form similar to that of the original G-L problem. An algorithm for doing this has already been developed (Cowie (1977)). Its implementation in this modelling system would involve a major increase in the size of the system, and as such it was left for a future enhancement.

8.2.4.7 Sensitivity over Environmental Scenarios

When a solution that looks attractive for a model for a particular set of environmental futures has been found, either by optimisation or simulation, it is desirable to be able to test that policy under

---

5. An example of this is given in the option "GOALS" in the decision "MANAGERIAL GOALS" in the Real Estate example given in Appendix C.
different environmental assumptions. Stratagem provides two methods for doing this. The first method, risk analysis, is discussed in more detail in section 8.2.5.3.

The second method is to use the "SELECT" command to switch over to a different set of environmental futures. When an "optimal" solution for example has been found under one set of futures, switching to a different set of futures preserves the value of the decision variables but causes the value of other variables to change to reflect the impact of the different futures. The model then reflects the results of a particular strategy, as represented by the value of the decision variables, applied under a different set of environmental assumptions. An example of this analysis is given in Appendix C.

8.2.5 Solution Techniques

A variety of different solution techniques was recognised as being necessary for a system over its intended decision support functions (section 7.4.2.3). The implementation of these in Stratagem is discussed below.

8.2.5.1 Deterministic Simulation

Deterministic simulation is the most basic method for solving a model. Stratagem uses the technique of recursive substitution which was developed in chapter four. This technique implicitly recognises the non-procedurality of the language, and has the added advantage of being able to automatically detect simultaneous equations.

Central to the technique is finding the next definition to be solved, as the modelling language does not require the definitions to be ordered in any particular fashion. The initial variable to be solved is always specified by the user, either as an interrogation through the terminal or as part of a report. To establish which definition to use it is necessary for the system first to check through all those futures and options which have currently been
selected. If no definition for the variable is found here, the remainder of the model definitions are "searched". If no definition is found, the variable is a decision item, and Stratagem asks the user to supply the value.

8.2.5.2 Optimisation

The optimisation method employed by Stratagem is linear programming. A model is formulated into a linear programme (LP) by the technique of recursive factorisation, which was described in section 6.4. To implement this in Stratagem involved the development of a complex new algorithm which is discussed in Appendix A.

The optimisation routine is invoked by entering "MAXIMISE" (or "MINIMISE"), followed by a Fortran-like arithmetic expression. This expression becomes the objective function of the LP. Stratagem uses recursive factorisation to reformulate this expression in terms of the decision variables in the model, and then displays the resultant function. The decision variables are all those variables for which no definition or value has been supplied. Variables which are defined in simultaneous equations are also treated as decision variables. The constraints for the LP are generated in terms of the decision variables by recursive factorisation using the constraint definitions in the model.

When the linear programme has been successfully formulated, Stratagem automatically invokes the Burrough's mathematical programming system TEMPO, passing it the LP by means of a disc file. Stratagem suspends itself while waiting for TEMPO to solve the LP, after which Stratagem restarts and reads the solution off a disc file created by TEMPO. Stratagem then displays the objective function value and makes the results available for further analysis by simulation. While Stratagem is suspended awaiting the LP results, the input-output monitor is still active, so that messages may be input to the system.
8.2.5.3 Risk Analysis

The risk analysis used in Stratagem is based on the probabilistic scenarios discussed in section 4.2, and is outlined in section 4.5.2. It is invoked by entering the keyword "RISK", followed by a list of variables, constraints, and/or expressions. This causes the system to go through the following sequence:

1) save the present status of the selected futures;
2) select the first combination of futures;
3) check that the probabilities of the futures in each scenario sum to one;
4) calculate the probability of this combination of futures occurring, being the product of the probability of each selected future in each scenario;
5) calculate by recursive substitution the value of each item in the list;
6) select the next combination of futures and return to step three;
7) if all combinations of futures have been done, the initial status of the selected futures as saved in step one is restored, so that the original model is preserved. This marks the end of the risk analysis, so the results are displayed and the analysis terminated.

The result of this analysis is a distribution for each item in the list. Stratagem displays the range, expected value, variance and standard deviation for each arithmetic item in the list, and for logical items it displays the probability that the condition will succeed or fail.

---

6. This step is necessary because of the facility for conditional probabilities. These can only be checked at run time.
7. This assumes the probabilities are independent, or if not, are expressed conditionally.
8.2.5.4 Simultaneous Equations

Although models do exist which are based entirely on a set of simultaneous equations (e.g. Francis and Rowell (1978), Warren and Shelton (1971)), a mechanism for solving simultaneous equations is not usually in itself sufficient for solving models. However because relationships within models can be simultaneously defined, a mechanism is required for solving those portions of the model.

Stratagem handles simultaneous equations in a manner that is completely invisible to the user. The technique of recursive substitution is able to detect any simultaneous relationship (section 4.5.1). Upon finding such a relationship, a special process is invoked that transforms the relationships into matrix form (see Appendix A). This is achieved by using recursive factorisation, and the results given back to the recursive substitution process. Unless the solution mechanism fails, which can be caused either by a non-linearity or by the matrix being singular, the user is completely unaware of the process. The advantages of this approach over those where the simultaneous relationship must be specified are illustrated in figure 8.6.

8.3 SUMMARY

This chapter has reviewed the structure of Stratagem, a computer programme which implements the modelling framework developed in this research. The review has concentrated on the general philosophy of the system, and how it fulfils the requirements for decision support systems so that it is accessible to the naive user. A complete description of the system is given in Appendix B, and some interesting implementation aspects are discussed in Appendix A.

The next chapter discusses two different applications of Stratagem.

8. For a discussion of systems of linear equations see Blum (1972).
CONTROL.

A

DATA.
create a
A IS NOW THE ACTIVE RECORD
DATA.
1969=2.45 2.67 2.98 3.27 3.61 3.98 4.39 4.84 5.34 5.89
DATA.
create b
B IS NOW THE ACTIVE RECORD
DATA.
create c
C IS NOW THE ACTIVE RECORD
DATA.
end
CONTROL.

b)
THE DEFAULT TIME RANGE IS 1969 - 1978
ANALYSIS.
edit model-for-b-and-c
THIS IS A NEW MODEL
EDIT.
10 b=a+c
EDIT.
20 c=b/a
EDIT.
saved
EDIT.
end
ANALYSIS.
model simul on save on using 0
ANALYSIS.
solve model-for-b-and-c
RESULTS OF MODEL-FOR-B-AND-C -- DEFAULT FILE IS FILE 1

TIME A(1) B(1) C(1)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.45</td>
<td>4.14</td>
</tr>
<tr>
<td>1</td>
<td>2.67</td>
<td>4.27</td>
</tr>
<tr>
<td>2</td>
<td>2.98</td>
<td>4.49</td>
</tr>
<tr>
<td>3</td>
<td>3.27</td>
<td>4.71</td>
</tr>
<tr>
<td>4</td>
<td>3.61</td>
<td>5.22</td>
</tr>
<tr>
<td>5</td>
<td>3.98</td>
<td>5.84</td>
</tr>
<tr>
<td>6</td>
<td>4.39</td>
<td>6.58</td>
</tr>
<tr>
<td>7</td>
<td>4.84</td>
<td>7.34</td>
</tr>
<tr>
<td>8</td>
<td>5.34</td>
<td>8.10</td>
</tr>
<tr>
<td>9</td>
<td>5.89</td>
<td>8.89</td>
</tr>
</tbody>
</table>

b) Stratagem input to create and run the simultaneous model.

Note:
1) User input is shown in lower case, computer output in upper case.
2) Simplan uses four modes, "data", "edit", "analysis" and "control". Stratagem does not use separate modes.
3) Simplan requires 14 lines of input to Stratagem's 7. (The symbol = acts as a prompt in Stratagem).
4) Simplan requires that the user indicate the initial and supply an initial value.

Figure 8.6: Comparison of Simplan and Stratagem Approaches to Solving a Simultaneous Model.
Chapter Nine:
Two Applications.

Two applications of the modelling system described in the previous chapters are discussed in this chapter. The applications represented two entirely different modelling philosophies, and their implementation demonstrated the flexibility and utility of the modelling framework developed in this research.

The first application was a simulation of a major strategic decision for Rockgas Ltd, a New Zealand company. The decision involved a large capital expenditure in order to enter into a new and growing market. Limited time was available in which to conduct the decision analysis, so there was considerable managerial pressure exerted throughout the modelling process.

The second application differs from the first in that it represents the reconstruction of an "advanced" optimisation model. The new implementation not only made it considerably easier to run the model, but also increased the range of useful analytical techniques available, thus providing for a better understanding of the model implications than was previously possible.
9.1 APPLICATION ONE: ROCKGAS LIMITED

Rockgas Ltd is a small company whose principal activities were in the nationwide distribution of imported bottled liquid petroleum gas (LPG) and ancillary equipment (such as Primus products). It is a wholly-owned subsidiary company of Fletcher-Challenge Ltd, which is New Zealand's largest public company (1981 revenue $2,082 million).

The development of an indigenous source of LPG, coupled with a government move towards energy self-sufficiency, created new opportunities for Rockgas. It was these opportunities it wished to explore using Stratagem.

9.1.1 Organisational Structure

Fletcher-Challenge is a large diversified company employing over 19,000 staff. It is divided into six business sectors, each more or less characterised by a common business area. Rockgas is a member of the largest sector, Rural and Trading, and in 1981 contributed just under 2% of that sector's revenue.

Each sector has its own senior management team, headed by an executive director from the Fletcher-Challenge board. All decisions involving an expenditure above a certain amount must be authorised at the sector level. In addition, major policy decisions, such as that contemplated by Rockgas, must be considered by the Fletcher-Challenge board.

9.1.2 Proposed Investment

The availability of an indigenous source of LPG plus the government's commitment to the development of alternative transport fuels (other than imported oil-based products) presented Rockgas with many new opportunities. These included the expansion of the product line to include LPG car conversion equipment, LPG bulk storage equipment, and other LPG related
products. There was also the opportunity to enter into the bulk
distribution of LPG and to establish retail outlets. It was
these last two possibilities that were investigated using Stratagem.

A national LPG distribution network would involve an investment
that was too big for any one company to contemplate. The network
would be centred on the LPG source at New Plymouth. Bulk depots
would be set up at strategic locations throughout the country.
LPG would be transported to these sites by specially constructed
road and rail tankers. In addition there was to be a custom built
LPG coastal tanker to deliver the fuel to coastal depots. Because
of the scale of these operations, it was proposed that a new
corporation, Liquigas Ltd, be established by a consortium of
interested companies to undertake this primary distribution.

Rockgas was not only planning to become a participant in Liquigas
but was also interested in establishing a secondary distribution
network. This network would use the Liquigas bulk depots as its
principal sources and would distribute LPG from these to customers.
To undertake this, it would be necessary to purchase a fleet of LPG
road tankers.

A further possibility was the establishing of a series of retail
outlets. These were to act as service stations, selling motor
spirits, CNG and LPG, as well as a wide range of automobile
accessories.

There were thus three decisions to evaluate:

1) Should Rockgas invest in Liquigas Ltd;
2) Should Rockgas develop a secondary distribution
   network;
3) Should Rockgas establish retail outlets.
The decision to develop a secondary distribution network was dependent upon a Rockgas involvement in Liquigas, and similarly, the establishment of retail outlets was dependent upon a Rockgas distribution network.

These were strategic decisions for Rockgas, involving a move away from the business of wholesale distribution of LPG accessories such as barbecues and camping equipment, into the distribution of LPG itself. This new business would completely dominate the old; the secondary distribution network alone would generate additional revenue in the order of two and a half times the existing revenue in its first year of operation, and would require a direct investment equivalent to twenty times the issued capital of Rockgas. The participation in Liquigas would require a further investment of around five million dollars, and would be the first move by Fletcher-Challenge into the primary energy area.

9.1.3 Stratagem Model

A Stratagem model was built to investigate the desirability of establishing a secondary network. The decision feature of Stratagem was then used to demonstrate the likely financial implications of vertical integration into retail operations. A model of Liquigas had already been built by another of the prospective participants in that company.

The Stratagem model was based on traditional simulation. The model entities included the principal sources of revenue and expenses, assets, liabilities, and cashflows. The implementation of the secondary network could take on a variety of different "shades". For example, the transport fleet could be either purchased or leased, and the drivers could be either employed or contracted. These "implementation options" are shown in figure 9.1. They were incorporated into the model using the Stratagem decision feature.
Implementation Options:

A: Transport.
   Buy
   □ Lease
   □
   { Capital Expenditure
   □ Transport Costs

B: Employment.
   Employ
   □ Contract
   □
   { Capital Expenditure
   □ Transport Costs
   □ Fleet Size
   □ Transport Costs

C: Vertical Integration (Retailing).
   None
   Implement
   □ Delay
   □
   { Revenue
   □ Capital Expenditure
   □ Revenue
   □ Capital Expenditure

Environmental Uncertainties:

Inflation Rates
Interest Rates
Margins
Volumes

Figure 9.1: Rockgas Implementation Options.
There were also a variety of future uncertainties to be considered. The ones that were felt to be most critical were the inflation rate, the interest rate, the LPG volume available, and the margin attainable on LPG (the selling price of which is indirectly controlled by the government). These factors were modelled using the Stratagem scenario feature, and the model was tested for sensitivity to these environmental factors by running it against selected combinations (e.g. high inflation and interest rates, low margins and volume sales).

Only one week was available to construct and run the model before the proposal was to be considered by the Fletcher-Challenge board. The construction details of this initial model are shown in figure 9.2, and its structure is shown in schematic form in figure 9.3. Because of the lack of material available on the time taken to build, debug and run models, it is not possible to make any definite conclusions on the relative performance of Stratagem with respect to other modelling systems.

The model was built to run in "chauffeur" mode, with the financial controller of Rockgas acting as the expert on the problem situation (the manager also took part on occasions). The construction of the model, requiring as it did a rigorous examination of all operations, highlighted several shortcomings in the current policy of Fletcher-Challenge with respect to its subsidiaries (for example there was no formalised corporate policy on dividends from subsidiary companies).

1. A recent survey of 42 IFPS users (Wagner (1981)) reports that the median length of a model was 200 lines, and the median time to build a model was 5 days. The Stratagem model for Rockgas was just under 300 lines and took 5 days to build and run. This compares very favourably with IFPS.

2. Many of these factors were attributable to the fact that Fletcher-Challenge had existed in that form for less than six months, being created by a merger of two other companies.
<table>
<thead>
<tr>
<th>Day</th>
<th>Hours</th>
<th>Activity</th>
<th>Variables</th>
<th>Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>Construct Revenue Logic</td>
<td>58</td>
<td>148</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>Construct Cashflow and</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Income Logic</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>Construct Asset Logic</td>
<td>59</td>
<td>115</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>Build Options and</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scenarios</td>
<td></td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>Debug Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>Run</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>38</td>
<td>Totals</td>
<td>150²</td>
<td>296³</td>
</tr>
</tbody>
</table>

Total time using Stratagem: 14½ hours elapsed time, 20 minutes process time.

Average session: 43 minutes elapsed time, 1 minute process time.

Longest Session: 1 hour 42 minutes elapsed, 3 minutes 20 seconds process.

Notes: 1) There is a certain amount of overlap between the various parts of the model (e.g. revenue, cashflow). These parts represent the order of construction, and not independent functional requirements.

2) Total number of variables in model, including parameters.

3) This does not include the decisions or scenarios.

4) The initial model input was done using the system editor (CANDE).

Figure 9.2: Summary of Model Construction.
Figure 9.3a: Revenue Calculations.
Figure 9.3b: Balance Sheet Calculations.
Running the model showed that the decision to proceed with the project was not sensitive to any one particular combination of the implementation options. These were therefore disregarded with respect to the accept/reject decision analysis, and were to be analysed in depth at a later stage should the project be given approval to proceed.

The expected availability and price of LPG, both critical inputs to the model, were determined by a model of Liquigas that had been constructed by another of the proposed partners in Liquigas. Unfortunately the output from this model was difficult to understand, being presented in tabular form with cryptic labels. Additionally the model itself was not available to Rockgas, so they did not know on what assumptions it had been based, and could not test some desired sensitivities. It was therefore decided to build a Stratagem model of Liquigas in an effort to expose the assumptions and to provide a means to test the sensitivities. This extra model (102 variables and 204 relationships) took three days to construct and debug. Its construction highlighted many of the assumptions in the other company's model, and also revealed some anomalies in its formulation. Although none of these affected the model results significantly, their discovery did enable the model outputs to be used with more confidence.

9.1.4 Model Results

The planning horizon for the model was seventeen years (the project life), and its output was given in four reports; two revenue reports (one being a summary of the other), a cashflow report, and a balance sheet. The outline of these is shown in figure 9.4. The results were all available within the limited time permitted. Based on the initial results and a list of assumptions in the model, the sector manager asked for a variety of sensitivity analyses before the proposal went to the board for final consideration. The reports generated by Stratagem were of sufficient quality to be used by the board.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Revenue</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LPG revenue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNG revenue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motor spirit revenue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other revenue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other income</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interest received</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gross revenue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LPG margin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNG margin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motor spirit margin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other margin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gross margin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other income</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interest received</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net revenue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Transport Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repairs and maintenance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Road user charges</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lease costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insurance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total transport cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Depreciation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport depreciation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Car depreciation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch depreciation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reception building</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Head office depreciation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total depreciation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Admin Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total salaries</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interest expense</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other expenses</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Administration costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Expenses</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Profit</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tax</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net profit after tax</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Less dividends</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Profit carried forward</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tax credit b/f</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Return on assets (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9.4: Outline of Rockgas Reports.**

a) Revenue Report.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Revenue Statement</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Gross Revenue</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Interest Received</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Net Revenue</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Transport Cost</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Depreciation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Administration Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Expenses</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Profit</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tax</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Net Profit After Tax</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Less Dividends</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Profit Carried Forward</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tax Credit B/F</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Return on Assets (%)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 9.4: Outline of Rockgas Reports.

b) Summary Revenue Report.
**Figure 9.4: Outline of Rockgas Reports.**

c) Balance Sheet.
### CASH FLOW STATEMENT

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH INFLUENCES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NET PROFIT AFTER TAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL DEPRECIATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHAREHOLDERS ADVANCES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEW ISSUED CAPITAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH IN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LESS CASH OUTFLOW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAPITAL EXPENDITURE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVIDENDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH OUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LESS CHANGES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NET CASH FLOW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9.4: Outline of Rockgas Reports.**

**d) Cash Flow Report.**
The decisions made were that:

1) Fletcher-Challenge (instead of Rockgas) would take up a 16.5% shareholding in Liquigas;

2) Rockgas would build and operate a secondary LPG distribution network;

3) Rockgas would not enter the retail market.

9.1.5 Ongoing Modelling

Several weeks after the decision to proceed with the secondary distribution network was made, it became necessary to evaluate in more detail the transport and employment components of the decision. These were specified as individual options within the model using the Stratagem decision feature. Each option represented the impact on the model structure, parameters and variables of each possible decision alternative (see figure 9.1). The model was run over a variety of environmental scenarios (such as limited availability of LPG, high inflation rates, etc) to investigate sensitivity. From this it was possible to eliminate many of the alternatives and select a broad transport and employment strategy. Some months later several possible tactics had been isolated for realising this strategy, and these were modelled using the decision features over a variety of likely scenarios.

At the same time as the broad strategies were being evaluated, the opportunity was taken to tidy up parts of the model. The balance forcing item\(^3\) was replaced by proper cash flow calculations, so that the model would only balance if proper account

---

3. A balance forcing item is a variable whose value is defined to be the difference between assets and equity plus liabilities. The inclusion of such an item will always make a model balance. However if proper account has been taken of the source and usage of funds, the model should always balance. A balance forcing item can obscure errors in the model formulation.
had been made of all sources and uses of funds. Additionally the handling of assets was extended to incorporate both tax and accounting conventions.

While the final transport tactics were being modelled the actual results for the first period covered by the model became available. Rockgas wished to incorporate these and a revised set of projections into the model. Several factors emerged from this exercise:

1) the model had been constructed for a "one-off" decision problem. Many aspects of the company's day to day operations had not been included. This made it difficult to reconcile the model inputs and outputs with budgeted figures;

2) because the model had been constructed in haste, no great effort had been paid to the use of a consistent notation for model variables. This made it difficult to alter the model;

3) the specifications and projections included in the model provided an extremely useful benchmark against which aspects of the organisation's performance could be evaluated. This was particularly evident when some projections were found to be significantly different from actual results, forcing a re-evaluation of the original projections, and through these an explanation of the differences.

It was decided by Rockgas that a new model should be constructed based on the old one, but more closely following the organisation's accounting structure and budgeting requirements. This new model is over eleven hundred statements long and took less than two weeks to construct and debug. It is fully compatible with Fletcher-Challenge's budgeting process, and has been used by Rockgas in its three year planning cycle.

4. This use of the model is analogous to that suggested in section 3.2.
9.1.6 Evaluation

The use of the model enabled:

1) a greater understanding to be reached of the financial profile of the project;

2) the realisation that the ultimate success of the project was not dependent upon the various implementation options (see figure 9.1). This meant that the project could be approved in broad outline, and more detailed analysis deferred to a later time;

3) projections to be quickly prepared over a variety of different scenarios;

4) progressive refinement of the evaluation of the transport alternatives.

Building the model highlighted:

1) the need for a consistent model notation;

2) the limitations of ad hoc decision models outside of their planned scope. The model specially constructed for the decision analysis was not suitable for more routine budgeting and planning applications;

3) the usefulness of specialised mechanisms to model the impact of proposed decisions.

It should be emphasised at this point that the model outputs acted as only one input to the decision process. There were many other non-quantifiable factors that were considered. It is a well-known limitation of this modelling approach that it cannot explicitly consider these other often very important factors.
This was the first use of Stratagem on a real problem in a "pressure" situation. Other modelling systems (such as IFPS and Simplan) would have been able to handle most of the analysis, except that their lack of a mechanism for modelling decision alternatives would have made the process extremely cumbersome. Although it is not possible to compare the relative performance of other systems in the areas in which they could have been used, the following points are worth noting:

1) It would probably not have been possible to successfully develop the model in the limited time period available by using a procedural modelling system. In such systems (e.g. Simplan, Plancode) it is necessary to specify all control structures, and to maintain the sequence of model statements in the order required for solution. These are implicit in non-procedural systems.

2) The incremental interpreter design philosophy was fully justified. Not only did it considerably reduce the number of steps and computer time required to build the model, but it also made model debugging and running (especially "what if?" analyses) relatively simple.

3) The elimination of line numbers also helped speed up model development. It was only necessary to remember what a particular item was called to check its value or definition, as opposed to the line where it occurred in the model. As statement ordering has no meaning in non-procedural systems, it is surprising that other such systems (e.g. CUFFS, IFPS) rely on line numbers. Two other major savings are that a special editor is not required, and that reports are freed from the artificial ordering imposed by line numbers.
4) The provision of a mechanism for automatically detecting and solving systems of simultaneous equations was essential. Despite the basic simplicity of the Rockgas model, several sets of these occurred naturally and it would have been time consuming to have had to isolate these (as in FPS and Simplan for example).

5) The use of decision structures greatly aided the analysis of the project. Without such structures it would have been necessary to write separate subroutines or structurally alter the main model to reflect each decision alternative.

6) It proved easy to learn how to use the system. The financial controller of Rockgas, with no previous experience and no formal tuition, learned how to effectively use the basic system (change definitions, print reports, etc) by watching the "chauffeur". This is probably due not only to the simple and consistent design philosophy imbedded in the system, but also to the fact that all interaction was conducted in his own vocabulary (which he in fact specified).

9.2 APPLICATION TWO: A REAL ESTATE MODEL

For the second application of the system, an entirely different modelling problem was selected. Whereas the model built for Rockgas was based on traditional simulation techniques and represented an actual problem to be solved under tight time constraints, the real estate model was built initially as the basis of a doctoral dissertation in the Department of Management Studies at the University of Auckland. It was originally constructed to investigate aspects of corporate real estate ownership under differing economic conditions. Thus this model differs markedly

5. For a complete description of the model see Snowden (1980).
from the Rockgas model not only because it is not based on a specific company but also because it is an advanced optimisation model.

The purposes for selecting such a model as an application were two-fold:

1) to illustrate that a correctly designed software system can considerably relieve the administrative burdens that are involved in the construction and running of advanced non-trivial models;

2) to show that a set of hitherto diverse but powerful techniques can, when integrated into a unified framework, usefully and easily extend the range of analyses possible.

A corollary to the first point is that had the real estate model initially been constructed using such a software system, it would not have been necessary to make many of the simplifying assumptions required to overcome the complexities arising from the need for manually preprocessing the model.

9.2.1 Model Background

The real estate model was constructed primarily to investigate the lease versus buy decision for corporate real estate. Additionally, the model allowed hypotheses relating to the impact of institutional constraints, inflation, and organisational goals to be tested within the lease versus buy analysis.

The structure and data used in the model was based on a sample of six New Zealand companies. The model had a planning horizon of five years. The primary decision variables in the model for each period were:
- the amount to invest in trading assets;
- the amount to invest in real estate;
- the amount of real estate to divest;
- the amount of finance to raise by
  1) equity issues,
  2) secured debenture issues,
  3) bank overdraft;
- the amount of bank overdraft to repay;
- the amount to pay out in dividends.

Thus the model had eight primary decisions spread over five time periods, making a total of forty decision variables.

The objective function used by the model was to maximise the present value of the firm to its shareholders. This was represented as:

\[
\text{the sum of} \begin{align*}
&\text{the present value of dividends paid} \\
&\text{the discounted terminal value of shareholders' funds (represented by share capital and retained earnings)} \\
\text{less} &\text{the present value of any new equity issued by the model (including a percentage issuing cost).}
\end{align*}
\]

The discount rate used was that giving a real rate of return of 7%, i.e. the rate is 7% above the prevailing inflation rate.
The model was run under three different scenario combinations:

1) inflation rate. Three inflation rates were considered, high (25%), expected (17%), and low (9%). The interest rates used in the model were dependent on the inflation rate;

2) the extent to which the nominal value of real estate investment responds to the inflation rate. Three possibilities were considered, where real estate assets increase by 33%, 66%, and 100% of the inflation rate;

3) external economic factors, reflected by the earning rate on trading assets. Three possibilities were considered, depressed (rate = 9%), normal (rate = 12.5%), and buoyant (rate = 20%) trading conditions.

The number of discrete possible futures described by these scenarios is $3 \times 3 \times 3 = 27$.

In addition to the three scenario combinations outlined above, the model was run under two different choice alternatives. The purpose of these was to investigate the effects of institutional bond constraints and managerial goals on corporate performance. Each choice consisted of two alternatives, making a total of $2 \times 2 = 4$ possibilities. Thus the total number of decision and scenario combinations considered was $27 \times 4 = 108$.

9.2.2 Original Implementation

The model was initially constructed using LPGen (section 5A), and the input for one of the combinations is shown in figure 5A.1. The analysis undertaken involved running the model under each of the possible combinations; a total of 108 model formulations and runs. The lines in the model which have to be changed for each scenario or decision are shown in figure 9.5.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Line No.</th>
<th>SCENARIO</th>
<th>DECISION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Inflation</td>
<td>Hedge</td>
</tr>
<tr>
<td>Objective</td>
<td>73-76</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Interest</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sdebt</td>
<td>101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maxint</td>
<td>114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sdebt</td>
<td>115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interest</td>
<td>120</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Creditors</td>
<td>121</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ngtrad</td>
<td>122</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Iprop</td>
<td>123</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Geota</td>
<td>124</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rntrad</td>
<td>131-135</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Rprop</td>
<td>136-140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dividend</td>
<td>-</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>18</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 9.5: Alterations Required to Switch Futures in Real Estate LPGen Model
If the model is changed in the most efficient order (i.e. by altering
the earnings scenario most and changing the inflation scenario only
twice) it is necessary to recalculate and alter a total of 234 model
lines, whereas if the least efficient order of changing is used it
is necessary to recalculate and alter 2197 lines. Thus there is
considerable manual effort required, and the scope for error is
large 6.

The analysis was accomplished by altering the model for each of
the possible combinations and then running it (a total of 108
model runs). The resultant LP models were of the order of 260
rows and 269 variables 7, with a matrix density of 0.75%, and each
taking 15 seconds of process time to solve (excluding setup and
printing time). Analysis consisted of a consideration of each
of the decision variables in each possible scenario. The
analysis did not extend to consideration of sensitivity, risk,
or of dual variables.

9.2.3 Stratagem Implementation

A Stratagem implementation of the model (see Appendix C) was
developed using the initial LPGen model as a starting point.
The model is equivalent to the initial model although it has
been changed as follows:

1) most constants that appeared in the model logic
   have been replaced by parameters;

2) all discount and interest rates, which had to be
   hand calculated in advance in the LPGen model,
   are calculated automatically by the model;

6. Without the aid of even a simple matrix generator such as
   LPGen the exhaustive methodology used would probably have
   been too complex to employ.

7. The LP size and solution time varied with the scenario
   used. Figures given are for the LP package Tempo run on
   a Burroughs B6700 computer.
3) the three scenarios and two decision options have been incorporated into the model, so that unlike the LPGen model, the Stratagem model is a complete statement of the problem;

4) subjective probabilities have been assigned to each future in each scenario;

5) reports have been defined for the presentation of results.

Thus what was previously represented as 108 different models has been able to be stated as a single problem, with a concomitant decrease in the manual effort required, and with accompanying benefits of better presentation of output and a full interactive system.

The Stratagem model was optimised over several scenarios and decisions, and the results were benchmarked with the corresponding results from the LPGen model. There were no significant differences in the results.

The LPs that Stratagem generated were considerably smaller than those generated by LPGen. This is because Stratagem removes all structural relationships from the model when it formulates an LP. Thus typically the objective function and the model structural relationships will reduce to a single row in the LP, with one additional row for each constraint in each time period, and one column for each decision variable in each time period. The real estate models because of a simultaneous relationship\(^8\), required an extra row and an extra column for each time period. Thus the resulting LP was 45 columns and 51 rows (66 rows when the managerial goals option was applied), with a matrix density of

\(^8\) The simultaneous relationship involved the inflationary effects on property (variables PROP and PROG). This was probably an error in the original formulation as the effect is to increase the property value beyond that suggested by the property hedge factor, and thus arbitrarily inflate the value of property against other assets.
37%. This is a considerable reduction in size from the LPGen model, and a resultant benefit is that the solution time is reduced by over 50%.

The reduction technique described above was developed for Stratagem so that implicit constraints included in the model definition would not be included in the LP (see section 5A.3.3). The real estate model had built in implicitly to its structure the requirement that the balance of the decision items (i.e. cash, property and overdraft), could not be less than zero. It was therefore necessary to include in the Stratagem model a non-negativity constraint for each of these items.

9.2.4 Summary of Stratagem Runs

A variety of different runs were made using the reconstructed model. The output from these runs is shown in Appendix C. The purpose of the runs was not to gain insights into the model itself, but to demonstrate that a better understanding could be gained by the use of an improved modelling technology.

9.2.4.1 Base Case

The model was run under the following conditions to provide a base case:

1) expected inflation (17%);
2) high hedge factor (100%);
3) low earnings (9%);
4) original bond indenture clauses;
5) managerial goals do apply.

This combination of scenarios and decisions represents one of the 108 possible.

9. This non-negativity requirement was explicit for the other decision items because of the assumption that trading assets could not be divested, and that finance raised through debentures had a repayment period longer than the model horizon. Thus neither of these two factors could become negative.
The run shows that the present value of the organisation (to the shareholders) is decreasing over time. This implies that a real rate of return of 7% is not being achieved. The risk analysis indicates that:

1) the objective function for this set of scenarios corresponds closely to the expected value given by all the scenarios;

2) the probability of a cash shortfall is 0.376. This is not insignificant, and if it occurs will require extra financing to cover. This will probably violate one of the operating constraints of the company. Thus there is some risk that, should the company follow this policy, it will find itself in an untenable position because of cash shortages.

3) there is a risk that the legally binding interest cover constraint will be broken (probability of between 0.164 and 0.250).

9.2.4.2 Low Inflation

The optimal solution from the base case was run under the low inflation scenario. This indicates that if the policy from the base case is followed and low inflation occurs, then the model is able to build up surplus cash, and the value of the firm to the shareholder increases.

The results also indicate that the overdraft and interest constraints are violated in periods three to five. These are, however, not significant, as the cash surplus more than covers the overdraft, so it can in fact be repaid at any time. This results in a corresponding decrease in the interest required, thus relieving the pressure on that constraint.

10. The output for the run is shown in Appendix C.
9.2.4.3 Post Horizon Effects (1)

The original scenarios used in the base case were restored and the horizon increased to ten periods. The model was rerun using the same objective function (i.e. maximise the value of the firm in period 5). The results of this run were the same as those for the base case. This shows that, in this particular case, post horizon constraints do not affect the solution.

9.2.4.4 Post Horizon Effects (2)

The objective function was altered to maximise the value of the firm in period ten, and the model rerun. The solution was, for the first five periods, the same as that for the base case, except that a different financing scheme was used. This probably represented an alternative optimal solution, made possible by having the same cost for both overdraft and debenture financing.

9.2.4.5 Lower Real Rate of Return

The results of the base case indicate that a real rate of return of 7% is too high (the present value is decreasing over time). This was lowered to 1% and the (five period) model rerun. The results show that this rate of return was just being achieved (the present value of the company does not change). Again the decision variables were unchanged, indicating that the model is not sensitive to changes in the required rate of return.

9.2.5 Review of Stratagem Implementation

The Stratagem implementation of the Real Estate Model had several advantages over the previous implementation. These are due in part to the flexibility of the software system and in part to the integration of techniques that is embodied in the modelling framework.
9.2.5.1 Model Specification

The simple tree structure incorporated into the framework made it possible to represent what were previously 108 distinct models as one model statement. The simple mechanism provided to select a set of scenario or policy options considerably reduced the time taken to run the model under any desired combination of these.

9.2.5.2 Model Output

The structure of the language made it considerably easier to understand the output from the model. Not only was it possible to view any sets of variables and constraints (or even expressions involving variables or constraints) on the output device, but also the report generator enabled the results to be presented in the form of standard financial statements. Thus a more meaningful presentation of results could be prepared faster than was possible using the old system.

9.2.5.3 Manual Errors

The automatic generation of the linear programme matrix coefficients meant that there was less room for arithmetic error. Instead of having to calculate the coefficients from their generating functions, it was sufficient to specify the functions themselves. This is clearly both faster and less prone to error.

An example of one type of "error" was discovered when discrepancies between the objective function values of the LPGen model and the Stratagem model were traced to round-off error in the calculation of the discount factors. For ease of data entry, these had been rounded to two significant figures in the LPGen model, whereas the Stratagem system uses eleven. The solution to the linear programme was in this case unchanged.
9.2.5.4 Model Analysis
The availability of the risk analysis made it possible to construct probability distributions of critical variables and relationships. This analysis, not previously possible, enabled the risk implications associated with a particular set of policy decisions to be evaluated.

The language features in the system made it easy to undertake other types of analysis, such as a quick investigation into the effects of the model horizon and the effects of changing the rate of return. Both of these analyses would have been very time consuming using a system such as LPGen.

9.2.5.5 Interactive
The interactive nature of the implementation means that results for each analysis can be achieved in a more timely fashion. Additionally it provides the opportunity to modify the solution sequence in mid-step. It is probable that the analysis done for the original research (Snowden (1980)) would not have had to be exhaustive over all 108 model combinations had such a modelling environment been available. For example, "worst", "expected" and "best" case optimisations may have been sufficient, with a "risk analysis" run on each of the decisions to indicate sensitivity.

9.2.5.6 Disadvantages
A possible disadvantage to the Stratagem approach was that traditional linear programming sensitivity analyses could not be carried out. This is a system implementation detail, in that it would be relatively simple to make such results available through Stratagem. The nature of the simulation-optimisation combination offers potentially more insights however than limited sensitivity analysis. Further, adopting the sanguine position that there is no optimal solution to a complex decision problem under conditions

11. Because interactive facilities were not available, the Stratagem version of the model was run in batch mode.
of uncertainty, it seems pointless to expend time analysing the scope of linear programming solutions. Instead, optimisation can be viewed as providing a good starting point, and a good solution can be arrived at by subsequent simulation.

9.3 SUMMARY AND CONCLUSIONS

The modelling system was applied successfully to two applications, each representing, in many respects, opposite ends of the modelling spectrum. The application for Rockgas, being based largely on traditional modelling techniques, was principally a test of the system implementation, i.e. modelling language, philosophy and features. The real estate model on the other hand was primarily a test of the modelling framework, i.e. the integration of a diverse set of techniques. The two models are summarised in figure 9.6.

The language proved to be sufficiently rich and powerful to implement easily both applications, yet sufficiently simple to be easy to learn and to use. Ease of use was greatly facilitated by the limited control structure, resulting from the non-procedural implementation, that allowed most dialogue to be carried out in the model's own vocabulary. The elimination of the traditional cumbersome line numbering approach by fully exploiting the nature of non-procedurality, further aided ease of use.

The real estate application demonstrated the practicality and utility of a modelling framework based on the integration of different solution techniques. The application showed that the combination of techniques can provide better insights and hence potentially better "solutions" than can the individual techniques in isolation.

The framework and the language are, from an implementation viewpoint, interdependent. The language not only overcomes the enormous complexities of the solution processes, but also enriches these processes. However the language is impotent without the underlying modelling framework, and the framework cannot be used without the language.
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Rockgas Model</th>
<th>Real Estate Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Decision Support</td>
<td>Investigatory</td>
</tr>
<tr>
<td>Size</td>
<td>150 Variables</td>
<td>81 Variables</td>
</tr>
<tr>
<td></td>
<td>296 Definitions</td>
<td>12 Constraints</td>
</tr>
<tr>
<td></td>
<td>27 Combinations</td>
<td>99 Definitions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>108 Combinations</td>
</tr>
<tr>
<td>Use</td>
<td>Interactive, tight time Constraints</td>
<td>Batch, no time Constraints</td>
</tr>
<tr>
<td>Principal Test</td>
<td>Modelling Language</td>
<td>Modelling Framework</td>
</tr>
<tr>
<td>Analysis:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Optimisation</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Scenarios</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Decisions</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Risk Analysis</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 9.6: Summary of Applications.
Chapter Ten: Summary and Conclusions.

The aim of this research was to narrow the gap between management science theory and practice, especially as applied to the use of formal models to support strategic decision making. Although management science techniques are widely known, they are less useful because they are largely inaccessible to the non-specialist.

This chapter gives a brief review of this thesis ordered into the areas of study which it encompasses.

10.1 STRATEGIC MANAGEMENT

In this research an organisation was viewed as an entity that operates in a complex, uncertain environment. Those organisational processes which were concerned with the long-term positioning of the organisation with respect to this environment were termed strategic.

For a formal model to be of assistance in strategic management it should span not only the organisation itself, but also the critical components of its environment and the mechanism through which these affect the organisation. A model should also be able to incorporate the uncertainties that characterise the environment. Since decisions must be made if the organisation is to operate successfully, the facility for explicitly modelling these should also be available.

The potential use of formal models in the strategic management process lies not only in their use as an evaluation aid for pre-specified decision alternatives (which is their more traditional role in strategic
planning), but also as a monitor of the organisation's perceptions of its environment. This formalised method of anticipatory problem recognition may be accomplished by a posterior comparison of model predictions with the situation that actually eventuated. Such an analysis may indicate deficiencies in the organisation's "world view". The application of models in this manner remains as an area for further research.

10.2 MANAGEMENT SCIENCE

The concept of an organisation interacting with a complex environment enables a model to be constructed as separate components:
- an internal model, describing the key processes within the organisation;
- an environmental model, describing the critical environmental factors;
- an interface model, describing the mechanisms through which the other two components interact.

A mathematical description of the processes that constitute these components was given in chapter four.

The Gear-Lockett framework was used as a basis for modelling uncertainties and decisions. The framework was modified to overcome many of the problems previously associated with it. The modifications included:
- simplifying the "tree" structure on which the framework is based. While this overcame the computational difficulties inherent in the approach, it also removed the ability to make contingent decisions. Further research is required to determine whether such decisions are useful, and if so, how to overcome the computational problems.
- allowing for synergy between decisions.
- removing the requirement for independence between stochastic trees.
- replacing the zero-one mixed integer programming solution technique by a combination of linear programming and simulation.
- developing a form of risk analysis so that the effects of uncertainty on decisions can be better understood.
10.3 DECISION SUPPORT SYSTEMS

If the modelling framework developed is to be of practical assistance in the decision making process, it must be accessible to the decision maker. This can only be achieved by means of a properly constructed computer system. A review was given of the nature of decision support systems, which are computer-based systems specially designed to support managerial decision making.

The implementation requirements of a decision support system for the modelling framework developed in this research were derived from a study of the decision process itself. A computer system called Stratagem was constructed based upon the integration of these requirements, the modelling framework, and concepts of software engineering.

The nature of Stratagem means that it is possible to construct models which are essentially "technique independent", i.e. models in which the solution technique is not explicit. In theory this enables models to be built which can be solved either by simulation, optimisation, or combinations of the two as desired. Further research is required to establish the utility of constructing solution independent models.

10.4 COMPUTER SCIENCE

The construction of Stratagem relied heavily on such computer science techniques as compiler construction, language design, virtual machine structure and dynamic storage allocation. Additionally, in order to implement some of the features, it was necessary to develop new techniques such as the multiple stack process in recursive factorisation (see Appendix A).

Extensions that are desirable for Stratagem include the provision for simple graphics and the ability to formulate zero-one mixed integer programming models. The methods for implementing the former of these are well understood, whereas those for the latter are not, although they will probably be based on recursive factorisation. Further research is required in this area.
A possible further feature for Stratagem is user-specified procedures. The structure of these could be based on Lambda Calculus (see for example Barendregt (1981)). At the simplest level these would be useful for respecifying data in common units and for applying commonly used calculations such as an adjustment for inflation. They would also, however, be able to handle more complex tasks such as special user-required depreciation routines. It is interesting to note that all of the present Stratagem functions (see section B6) could have been implemented in this manner. Further research is required to establish methods of implementation and the limits to the technique.

10.5 STRATAGEM

The two applications described in chapter nine provided preliminary evidence that the system is successful: it makes a set of management science tools accessible to the manager in an integrated, easy-to-use and relevant manner. That is not, of course, to suggest that it is the "ultimate" in modelling systems. A great deal more experience is required with the system to discover its short-comings and any areas in which it can be improved.
Appendix A: Computer Implementation Aspects of Stratagem.

Stratagem is a decision support system generator built to implement the modelling framework developed in this research. The system is implemented as a non-procedural language based on an incremental compiler/interpreter. The design of the system is based on the user requirements discussed in chapter seven.

This appendix discusses some of the more novel aspects of the computer implementation of Stratagem. The large size of the system precludes a detailed discussion of all its aspects. Instead a review of its characteristics is presented, followed by a discussion of the algorithms developed to convert a set of Fortran-like statements into a linear programming matrix. It is assumed throughout the discussion that the reader is familiar with computer terminology and concepts.

A1 SYSTEM ENVIRONMENT

Stratagem is a programme written in Burroughs Extended Algol for a Burroughs B6700 computer. The programme is over 8500 lines long, and consists of 154 procedures composed of 55000 syntactic items.

Stratagem is a recursive system, being based in part upon the techniques of recursive substitution and factorisation. The B6700 is a stack based machine, enabling it to handle recursion naturally through the hardware, and hence is ideally suited to the implementation of Stratagem. For example, the Rockgas model
described in chapter nine was constructed and run on a heavily loaded dual B6700 computer supporting a nationwide system of online transaction terminals. The construction and running of the model had no discernible impact on the functioning of the system.

A2 SYSTEM STRUCTURE

Stratagem is based on the following components:

- a "spread sheet". This is used to store the values of variables and constraints as they are calculated, and is implemented as an array. It corresponds to the model solution matrix (figure 4.1).

- a symbol table. This set of arrays contains the name, type and location of corresponding data structures (if any) of all identifiers used in the model.

- code storage. As each statement of model logic is read into the system it is compiled into virtual machine code which is stored in this array.

- symbol storage. The symbolic (textual) representation of the model logic is stored in a random access file.

- variable environment. This array links the variable (and constraint) items in the symbol table with corresponding entries in the spread sheet, the code storage, and the symbol storage.

- structure storage. The structure of all decisions, options, scenarios and futures is stored in this array. The actual code and symbolic representations of any definitions contained in the options and scenarios are stored in the code and symbolic storage areas.

- report storage. As each report is entered, it is compiled into a special interpretive intermediate code which is stored as a separate file. The corresponding symbolic representation is also stored in a file.
The components are shown in figure A1.

A set of compilers and virtual machines has been constructed for converting the user's input to a form that can be used by the computer.

- **statement compiler.** This is used to compile each statement of model logic into virtual machine code as it is entered. This is thus the principal compiler of the system. The generated code is stored in the code storage array.

- **report compiler.** When a report is specified, it is converted into a code which can later be interpreted to produce reports of the required layout. The report code is stored in a report storage file.

- **structure compiler.** The decision and scenario structures are converted into a series of linked lists, which are held in the structure storage array. The statement compiler is invoked to compile the augmented and replacement definitions (see Appendix B). These definitions are accessed through the linked list, and not the variable environment array.

- **statement virtual machines.** There are two virtual machines for executing the model logic. One of the machines solves the statements using recursive substitution (section 4.5.1), whereas the other machine transforms the model as represented by the virtual machine code statements into matrix form by recursive factorisation (section 6.4), and specialised routines are then called to "solve" these matrices. The virtual machines are discussed in detail in section A4.
Figure A1 Stratagem Data Structures.
report virtual machines. One report virtual machine uses the statement virtual machines to gather the data for a report and leave it in a disc file. The second report virtual machine converts this file into a form suitable for display.

These routines and data structures are linked by a set of ancillary processes.

- data management routines, which maintain the various data structures used by the system (code storage, symbol table etc).
- scanning routines, which collect and determine the type of the input items given by the user on a token by token basis.
- command user routines, which are a collection of special routines to handle individual types of input, such as variable declaration, saving a model, documenting a model, etc.

A2.1 Model Spread Sheet

The model spread sheet is used for storing the values of variables and constraint items. It is implemented as a one-dimensional array, with a mapping algorithm that logically divides the array into a matrix, where each row represents a variable or constraint and each column a time period. The one-dimensional structure was preferred to the more obvious two-dimensional array because in this case it is more efficient, being much quicker to reinitialise.

The elements of the array are B6700 words, each of 48 bits. The top bit in each word (bit 47) is not used in B6700 arithmetic, and can thus be used as a tag to indicate whether or not the element contains a legitimate model value. If the tag is zero, the element contains a proper value, otherwise the value must be calculated. The use of a tag in this manner is fundamental to the realisation of non-procedurality: it allows the system itself to determine if a variable needs to be solved, removing the onus from the user.
A2.2 Symbol Table

The symbol table is a set of arrays that contain information about all identifiers used in the model (i.e. all reserved words and items declared by the user, such as the names of variables and reports). Each time the scanner discovers an identifier in the input stream, it searches the symbol table to find the characteristics of the identifier.

Items are stored in the symbol table using the following technique:

- the first three and the last two characters of the identifier are stored.
- the length of the identifier is stored (up to a maximum of 16).
- a hash of the identifier is calculated based on the EBCDIC representation of each character. The hash is composed of the sum of the lower four bits of each byte, and is stored modulo 32.

In this manner each item can be stored as a bit pattern in one 48 bit word.

The advantage of this approach, apart from the speed of looking up items in the table, is that it allows the transposition of the middle (i.e. after the first three and before the last two) characters of an identifier. This is a very common keying error. The disadvantage of the approach is that it is based on a many-to-one mapping, so that different identifiers can have the same representation. Although such items would be flagged as they are declared, it is almost certainly preferable to have a one-to-one mapping for a symbol table, even though this will cause transpositions to become errors.
Associated with the symbol table arrays is a temporary work file. This is used to store a complete representation of each identifier, and also holds the description of the item as specified by the user. The file is accessed whenever the item's full name or description are required.

A2.3 Code Storage

The virtual machine code generated by the statement compiler is stored in a linear array. Space is allocated in the array using a technique similar to that given by Knuth (1968; p.435). The scheme has the advantage of eliminating the need for a separate garbage collection routine. The storage is handled as follows:

1) All array space is allocated in segments. A segment can be either "in use" or "empty", and has the following structure:

<table>
<thead>
<tr>
<th>USE</th>
<th>LINK</th>
<th>LENGTH</th>
<th>CODE SPACE</th>
<th>LINK ADDRESS</th>
<th>USE</th>
</tr>
</thead>
</table>

where:

- **USE**: is a flag indicating whether the segment is empty (and hence available for use), or not;
- **LINK**: is a pointer to the next segment in the chain, as discussed below;
- **LENGTH**: is the number of available bytes in the code space;
- **CODE SPACE**: is the space in the segment where code may be stored;
- **LINK ADDRESS**: is the position of the start of the segment, as represented by the LINK field.
2) Empty segments are chained together through the LINK fields. A value of zero indicates the end of the chain. The LINK field is not used when the segment is in use.

3) The array is initially set up as two segments, one (marked "in use") used for scratch storage into which statements can be compiled, and one empty segment spanning the remainder of the array.

Whenever space is required to store some code, the chain of empty segments is searched until one of sufficient length is located. The segment is removed from the chain and marked "in use". If the segment is significantly longer than required only a portion corresponding to the actual length required is taken, and a new, shorter, empty segment is constructed of the remainder.

When a segment is no longer required, it is marked empty and added to the chain of available segments. If the preceding or following segment is empty, the newly deallocated segment is added to this to make one longer segment. If both the preceding and following segments are empty they are joined to form one empty segment and added to the chain. This situation also requires that a link be removed from the chain to account for the elimination of one segment.

The advantage of this scheme is that, apart from eliminating the need for a special process which periodically scans through the segments and collects available ones (garbage collection), it also reduces the possibility of the array becoming checkerboarded. The approach means that there can never be two contiguous empty segments.
A2.4 Symbol Storage

The source statements from which the virtual machine code is generated by the statement compiler are saved in a temporary, random access work file with fixed record length. When more than one record is required to store a symbolic statement, the records are chained together using the last word in the record as a pointer field. Deallocated records are also stored as a chain for re-use so that the file does not grow too large. Whenever the symbolic representation of a definition is required, the appropriate records are read into the system.

The statements are saved on a token by token basis by the scanner as they are entered. This allows the elimination of unnecessary spaces, and thus results in compact and tidy representation of each statement.

A2.5 Variable Environment

The variable environment is a set of arrays which link the entry for each variable (or constraint) in the symbol table to its location in the spreadsheet, and also store the addresses of the corresponding statement code and symbolic. There is one entry in the array for each row of the spreadsheet (i.e. one entry for each constraint and variable in the model), and an additional entry for each definition that pertains to the variable or constraint. Whenever the item has to be solved, the array is searched to find the start address of the appropriate code to execute. The structure of the variable environment is shown in figure A2.

Although the present implementation of the variable environment is satisfactory, it could be improved. The reason for this is that the array must be searched whenever a variable or constraint is to be solved, and thus is accessed many times in a model run. This becomes particularly apparent when risk analysis is being done, and it is necessary to solve the model many times. This is
Figure A2: Structure of Variable Environment.
an expensive process, and it appears that most of the time is spent searching this array. Thus it is likely that significant benefits could be gained if a more efficient scheme, such as simple linked lists, were used.

A2.6 Structure Storage

Decisions, scenarios, options and futures are stored in any array. An array is split into two lists, one for decisions and one for scenarios. Each item in the list consists of two words, each separated into a variety of different fields.

Each decision in the list is represented as a list of options, and each option is represented as a list of items (variables or constraints) that are affected by that option. Each scenario is similarly represented as a list of futures, except that each future also has a probability definition associated with it. This is illustrated in figure A3.

The definitions that constitute an option or future are represented as list items which contain the variable name and definition type as well as the addresses of the corresponding code and symbolic statements. To find the details of a specified variable in the structures it is necessary to traverse all of the chains in the structure storage.

A third list of available items is maintained in the array. As list items are removed or replaced, the array space made available is added to this list. When space is required for new items, any deallocated space is used first.
Figure A3: Structure of Structure Storage.
A2.7 Report Storage

Each report is stored as two separate files. The first of these contains the report symbolic, as entered by the user. The second file contains the report code, which is a compiled version of the report symbolic. The code is not the same as that produced by the statement compiler.

When a report is to be displayed, the corresponding report code file is accessed. This describes the contents and the format of the report. An interpreter converts the code in this file into a temporary file, which contains the data to be printed as well as the code. A report listing routine then analyses this file and displays it in the specified format for the user.

The temporary, intermediate file was used so that comparison reports could be prepared. This feature was not however fully implemented. The file would be saved, and the data in it compared with those from subsequent model runs. The comparison would be presented to the user in the same format as that of the original report, but with the addition of the results from the later model and (optionally) percentage and absolute differences.

A3 STRUCTURE OF THE STATEMENT VIRTUAL MACHINES

All calculations in Stratagem are done by the statement virtual machines. As each model statement is presented to Stratagem it is compiled into virtual machine code by the statement compiler. This code is interpreted by the virtual machine when the expression is being solved.

There are two virtual machines for doing calculations in Stratagem. One is for calculating expressions using recursive substitution (see section 4.5.1), and the other is for factorising expressions using recursive factorisation (section 6.4). Both machines use the same interpretive code, but their
actions are different for many of the instructions. The machine for recursive substitution is in many respects typical of other stack machines. The recursive factorisation machine is markedly different, and is discussed in more detail in section A4.

A3.1 Virtual Machine Instruction Set

Both machines are designed around a stack process. This approach, unlike the more usual register-oriented von Neumann machine, lends itself to recursive programming. The virtual machine stack is used only for arithmetic; other functions, such as the storing of return addresses, are handled implicitly in the structure of Stratagem itself.

There are two instructions that load addresses onto the stack:

- **NAMC** - name call. This operation loads the row number (the "name") of the variable or constraint onto the top of the stack.

- **NAMI** - indexed name call. This operation loads the row number of an indexed variable (section B4.1) or constraint onto the top of the stack.

The row number is the number of the row in the spread sheet which is used to store the values of the variable or constraint item. When the row number has been loaded onto the stack, two further instructions are available that operate on this to calculate the address in the spread sheet of a row element, or the value of a particular row item.

- **LADR** - load address. This operator calculates the actual address of an item in the spread sheet. The address is the offset of the item from the base of the spread sheet, and is calculated from the row number at the top of stack and the current (model) time period. The time period is held in a special machine "register", and is maintained automatically by the system.
LVAL - load value. This operator calculates the actual address as above, and then proceeds to determine the value of the item, which it leaves at the top of stack. This operator is discussed in more detail in section A3.2.

There are two immediate instructions for loading model constants onto the stack.

LIT4 - literal call 4 bits. Load the four bits following the instruction onto the top of stack as an unsigned constant.

LT48 - literal call 48 bits. Load the following 48 bits (one B6700 word) onto the top of stack as a constant.

Two of the machine "registers" may also be loaded onto the stack and used in calculations.

LTIM - load the model time period as a constant onto the stack.

LHOR - load the model horizon as a constant onto the stack.

Arithmetic instructions operate on the top two stack items, and reduce the size of the stack by one.

ADD - add the two values at the top of stack, and leave the sum at the top of stack.

SUB - subtract the item at the top of stack from the item one below, and leave the result at the top of stack.
MULT - multiply the two items at the top of stack, leaving the product at the top of stack.

DIV - divide the second value in the stack by the top-of-stack value, and place the result at the top of stack.

An instruction to change the sign of a stack item corresponds to the "unary minus" operator.

CHSN - change the sign of the top of stack item.

The relational operators also work on the two top-of-stack values. They leave a value of true or false, represented by a decimal 1 and 0 respectively, at the top of stack depending upon the result of the operation. A tolerance level of $10^{-6}$ is used to account for round-off errors.

LEQ - less than or equal to. Returns true if the second stack item is less than or equal to that at the top of stack.

EQL - equal to. Returns true if the top two stack values are equal.

GEQ - greater than or equal to. Returns true if the second stack item is greater than or equal to the top stack value.

A special operator is provided for implementing conditional statements.

COND - evaluate a conditional statement. This operator causes the "IF clause" in a conditional expression to be evaluated. If the clause returns true, the operator (recursively) calls the "THEN expression", otherwise it calls the "ELSE expression".
The function intrinsics in the system, such as NPV and SUM, have been implemented in virtual machine code. This is necessary if the functions are to be available for factorisation. (i.e. simultaneous equation resolution and linear programme generation). Functions are invoked by a special operator, and another operator is used for evaluating function parameters.

**FNCT** - evaluate function. This operator invokes the code for the specified function, and leaves the result at the top of stack.

**LDP** - load the value of a function parameter. This operator evaluates the parameter expressions and leaves the result at the top of stack. All function parameters are passed by expression.

There are two operators for taking the results of calculations from the stack and storing them into the spreadsheet.

**STON** - store non-destructive. The value stored at the second stack position is placed into the spreadsheet at the address (calculated by the LADR instruction) on the top of stack. After the operation the address is removed, so that the value becomes the top of stack item.

**STOD** - store destructive. This is the same as the STON operator, except that both the address and the value are deleted from the stack.

All calculations are made by recursion, which can be invoked by the LVAL, COND, and LDP operators\(^1\). A special operator returns to the previous level of calculation.

---

1. An interesting effect of this is that it removes the need for explicit branching instructions.
RETN - return. This operator forces Stratagem to return to the previous level of calculation. It does not affect the operand stack, and is the operator that is used to terminate calculation sequences. Error conditions detected in other operations, such as an attempted divide by zero, also force a return to the previous level.

There are three operators available for stack maintenance.

DUPL - duplicate the top of stack item.
SWAP - interchange the top two stack items.
DEL - delete the top of stack item.

Two operators are used to display intermediate results. These are emitted when the PRINT modifier is used, and when the system is operating as a calculator.

DISP - display the name and value of the top of stack item at the user's terminal.

PRNT - print the name and value of the top of stack item onto the special print file (section 8.1.2) and also display it on the user's terminal.

A3.2 Loading the Value

The LVAL (load value) instruction is the most complex and important of all the operators. This operator determines if the value is available, and if not, how it should be calculated.

The operator first checks the "spread sheet" at the indicated address. All variable and constraint values are stored in the spread sheet, which is a one-dimensional array of B6700 48 bit words. The "top" bit in each word (bit 47) is not used in the representation of numeric items, and hence is available for use as a tag. The following values are used for the tag to represent different types of value.
Bit 47 = 0  The word contains a valid numeric item which the LVAL operator can use.

Bit 47 = 1  The status of the data item depends on bit 46.

Bit 46 = 0  The value has not been calculated (it is "undefined"), and the LVAL operator must determine the address of the statement that defines the value. If such a statement exists, bit 46 is set 1, and the virtual machine is recursively called to calculate the value. If no statement exists, a value of "undefined" is returned, or, in some situations, the user is requested to supply the value.

Bit 46 = 1  The value of the item is in the process of being calculated. When the LVAL operator encounters this tag value, it means that the statement that defines the value exists in a set of simultaneous relationships. These equations are solved by invoking the factorising virtual machine.

The factorising virtual machine uses additional tag values to indicate temporary storage addresses for linear expressions.

A3.3  Comment on Virtual Machine Instruction Set

With the benefit of experience, it would now be possible to eliminate some of the virtual machine instructions and incorporate their functions in other operations. As the interpretation of the virtual machine code is a relatively slow operation, a smaller, more powerful instruction set should increase the speed of execution.
The name call operators (NAMC and NAMI) are unnecessary since they are always immediately followed by a load instruction (LVAL or LADR), and every load instruction is preceded by a name call. Hence the name call function can be incorporated into the load instructions. As the sequence "name call, load" is probably one of the most used, its elimination could have significant benefits in terms of speed of execution.

The two store instructions (STON and STOD) are also unnecessary, as their operations can be equally well incorporated into the LVAL instruction. One function of LVAL is to determine the value of the item from its address. As it "knows" both the address and the value, there is no reason why it should not also act as a store instruction in those situations where the value has had to be calculated.

A4 FACTORISATION VIRTUAL MACHINE

In order to solve a model by linear programming or to solve a portion of the model as a set of simultaneous equations, it is necessary to convert the appropriate part of the model into matrix form. As no details of such an operation could be found, a new method had to be developed. A summary of this method, termed recursive factorisation, and its implementation, is given in this section. An example of the approach was given in section 6.4.

To form a matrix it is necessary to operate on a set of model statements (in this case in compiled form), to determine which variables are to become columns in the matrix, and to factorise the model into a set of linear combinations (rows) of these variables. When the matrix is generated, it can be solved using special routines. For simultaneous equations, LDR factorisation (see for example Blum (1972)) could be used, whereas a linear programming matrix could be solved by another computer programme, in this case Burroughs Tempo®.

A4.1 Structure of the Factorised Matrix

The result of the factorisation process is a matrix which contains a compressed version of the model statements. The structure of the matrix depends on its type:

1) Simultaneous Matrix. A simultaneous system is identified by the substitution virtual machine when it finds a variable with a value of "partially defined" (i.e. bit 47 = bit 46 = 1). The value for that variable must be determined simultaneously with at least one other variable value. Each such variable is allocated a column in the matrix. Each row in the matrix is formed by applying the factorising virtual machine to the definition of one of the simultaneous variables. For example, the statements:

\[
\begin{align*}
A &: 3 \times B + 3 - C/3; \\
B &: A + 6/(C \times 2) - 5; \\
C &: 3 \\
\end{align*}
\]

are converted into the following matrix:

\[
\begin{bmatrix}
A & B \\
1 & -3 & 2 \\
-1 & 1 & -4 \\
\end{bmatrix}
\]

2) Linear Programme Matrix. The columns of the matrix represent decision variables. These correspond to variables encountered in the model which do not have a definition, or which are defined to be "undefined". Whenever the factorisation algorithm discovers such a variable it is allocated a column.
The objective function is as entered by the user, but expressed in terms of decision variables. The constraint rows correspond directly to the model constraints, except that they are again expressed in terms of the decision variables. Stratagem constraints which do not have a definition are ignored. Any simultaneous relationships encountered are also incorporated directly into the linear programme. In certain situations this may involve adding extra columns or rows.

A4.2 Multiple Pass Recursive Factorisation

The first algorithm that was developed to formulate a linear programming matrix made multiple passes through the virtual machine code, each pass determining one of the matrix coefficients. Each "pass" consisted of executing the appropriate code using a revised set of arithmetic operators, the actions of which depended upon the type of the top two stack elements (see figure A4).

Each stack element was composed of two words, one for the coefficient and one for a tag which identified the operand. Each pass through the virtual machine code operated only upon the stack elements that corresponded to the particular matrix coefficient being determined (the "target"). All other stack items, with the exception of constants, were discarded from the stack as soon as practicable: the coefficients for these items would be determined in a separate pass.

The first pass was made to isolate the constant term of the objective function. This was accomplished by executing the objective function with a "target" of constant. During this pass, a list was constructed of any decision variables encountered. Subsequent passes took each decision variable in turn and executed the objective function with these as the target. In this manner the objective row of the matrix was constructed. The process was then repeated for each constraint in the model.
<table>
<thead>
<tr>
<th>OPERATOR</th>
<th>CONSTANT</th>
<th>TARGET</th>
<th>CONSTANT</th>
<th>TARGET</th>
<th>TARGET</th>
<th>UNKNOWN</th>
<th>UNKNOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>ADD</td>
<td>SWAP</td>
<td>DELETE</td>
<td>ADD</td>
<td>SWAP</td>
<td>DELETE</td>
<td></td>
</tr>
<tr>
<td>SUB</td>
<td>SUB</td>
<td>CHSN</td>
<td>DELETE</td>
<td>SUB</td>
<td>CHSN</td>
<td>DELETE</td>
<td></td>
</tr>
<tr>
<td>MULT</td>
<td>MULT</td>
<td>MULT</td>
<td>MULT</td>
<td>ERROR</td>
<td>ERROR</td>
<td>ERROR</td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td>DIV</td>
<td>ERROR</td>
<td>DIV</td>
<td>DIV (set tag = constant)</td>
<td>ERROR</td>
<td>ERROR</td>
<td></td>
</tr>
</tbody>
</table>

CONSTANT - constant (tag 0); TARGET - item being factorised; UNKNOWN - item to factorise in another pass. ERROR - non-linear operation; [Operation for first pass when TARGET is CONSTANT]. TOS - top of stack.

Figure A4: Multiple Pass Arithmetic Operations.
To illustrate the approach, consider the objective function

$$\text{Max } 3 * X + Y * (2 * Z) - W;$$

and the model statements:

$$Z : X / 2 - 1 ; $$
$$Y : 4 ; $$

These statements when entered are compiled into virtual machine code. Arithmetic expressions in this code are converted to a form of reverse Polish notation, so the statements become:

$$\text{Max } 3 X * Y 2 Z * * + W -$$

and $$X 2 / 1 - Z : $$
$$4 Y : $$

where ":" is a store operation.

The sequence of calculations and the different stack states are shown in figure A5 for the number of passes that must be made to completely factorise the objective function.

The resultant linear form of the objective function in terms of the decision variables is:

$$-8 - 7X - W$$

A4.3 Discussion of Multiple Pass Factorisation

The algorithm was able to factorise any (linear) system as required. However to do this required that the model (or portion of the model) by "solved" once for each of the factorised variable (i.e. columns in the final matrix). This proved extremely slow.
<table>
<thead>
<tr>
<th>Operation</th>
<th>First Pass Stack</th>
<th>X Pass Stack</th>
<th>W Pass Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>(3,C)</td>
<td>(3,C)</td>
<td>(3,C)</td>
</tr>
<tr>
<td>X 4</td>
<td>(3,C) (1,U)</td>
<td>(3,C) (1,X)</td>
<td>(3,C) (1,U)</td>
</tr>
<tr>
<td>*</td>
<td>(1,U)</td>
<td>(3,X) (4,C)</td>
<td>(1,U)</td>
</tr>
<tr>
<td>Y</td>
<td>(1,U) (4,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td></td>
<td>(A,Y)</td>
<td>(A,Y)</td>
<td>(A,Y)</td>
</tr>
<tr>
<td>:</td>
<td>(1,U) (4,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>Z</td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>X 2</td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>/</td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>1</td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td></td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>-</td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>Z</td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td></td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>*</td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td></td>
<td>(1,U) (4,C) (2,C)</td>
<td>(3,X) (4,C)</td>
<td>(1,U) (4,C)</td>
</tr>
<tr>
<td>+</td>
<td>(-8,C)</td>
<td>(1,U)</td>
<td>(-1,W)</td>
</tr>
<tr>
<td>W</td>
<td>(-8,C) (1,U)</td>
<td>(1,U)</td>
<td>(-1,W)</td>
</tr>
<tr>
<td></td>
<td>(-8,C)</td>
<td>(1,U)</td>
<td>(-1,W)</td>
</tr>
</tbody>
</table>

Where \((c,t)\) represents a single stack item of form \((\text{coefficient}, \text{tag})\)

- \(\text{tag} = U\): item is irrelevant in that context.
- \(\text{tag} = C\): item is a constant.
- \(\text{coefficient} = A\): item is an address.

Figure A5: Multiple Pass Factorisation Sequence.
An experiment with the initial implementation of the algorithm showed that it would take about twenty minutes of process time to formulate a moderate sized linear programme (this model, of 50 columns and 45 rows, is shown on Appendix C). Refinements to the implementation of the algorithm, such as saving intermediate results, resulted in a reduction of the process time requirements by a factor of ten. However even two minutes to formulate the linear programme was too expensive both in terms of user time (for interactive applications), and in computer time. An improved factorisation algorithm would be needed if optimisation was to be a viable modelling technique in this framework.

A4.4 Single Pass Factorisation

The multiple pass factorisation algorithm was not satisfactory because it was too slow. A single pass algorithm was therefore developed with the expectation that this would significantly reduce the time required.

The single pass algorithm uses a complex stack structure. It works by gathering and manipulating linear combinations of the variables. The virtual machine instructions treat these linear combinations as single stack entities. The result of the single pass is a single item on the stack. This item is the linear reduction of the model.

The essential difference between this algorithm and the multiple pass method lies in the objects manipulated on the stack : in the multiple-pass version these were essentially single numbers, whereas the single-pass version manipulates complete algebraic expressions. In effect the new version "does the algebra" on the stack.
A4.4.1 Stack Structure

The method is based on collecting and manipulating all the factors at one time on the stack. To accomplish this it was necessary to implement three parallel stacks for the virtual machine:

- coefficient stack, to hold the coefficients of the various factors as they are being manipulated;
- tag stack, to hold the tag that identifies the corresponding item in the coefficient stack;
- environment stack, to hold the details of each stack item. Since each item of the (coefficient and tag) stack can have a different size, it is necessary to store both its size and stack position.

The stack structure is illustrated in figure A6.

Because stack items are no longer simple, single elements, the operations of the virtual machine become correspondingly more complex. For example the SWAP instruction must move entire blocks of items from both the coefficient and tag stacks, and also update the environment stack. Despite this added level of complexity, most operations have the same function as in the simple virtual machine.

A4.4.2 Revised Operations

Whenever the stack is adjusted, it is necessary to alter all three of the actual stacks. For example, whenever a constant is placed on the stack (by means of either a LIT4 or LT48 operation), the value is placed on the coefficient stack, the corresponding tag in the tag stack is set to "constant", and an element is added to the top of the environment stack indicating the position and size of the new stack element (constant). The functioning of the
Figure A6: Single Pass Stack Structure.
operators has been altered so this can be done automatically.

For addition and subtraction (the ADD and SUB operators) each component in the top stack item is taken in turn, and the components in the second stack item are searched until one is found with the same tag. The coefficient of this item is then replaced by the sum (or difference) of the two coefficients. If no entry is found, the element is transferred from the top stack item to the second item. When all components of the top stack item have been added (or subtracted), the top item is deleted from the stack.

From this it can be seen that each stack item in fact represents a linear combination of the decision or simultaneous variables and their coefficients. At the end of the calculation sequence only one item will remain on the stack, and this corresponds to a row in the factorised matrix.

The multiplication and division operators (MULT and DIV) must check for linearity. The only valid form of multiplication is where a stack item is multiplied by a constant. Thus provided at least one of the two items at the top of stack has a size of 1 and a tag of constant, the multiplication can proceed. The coefficient of each element is multiplied by the coefficient of the constant, and the constant item is then removed from the stack. If one of the two top stack elements is not a constant, the operation will be non-linear, so an error is given.

Division is similar to multiplication, except that the item at the top of stack must be a constant, otherwise the operation will be non-linear. The coefficients of the second stack element are each divided by the coefficient of the constant, and the constant is removed from the top of stack. A special case in division is where both the stack items are identical. In this case the result of the division will be one, or, if they differ only in sign, negative one.
The relational instructions (GEQ, EQL, LEQ), which will appear only in the linear programming constraints, are treated as subtraction operators. This has the effect of transferring the right hand side of the expression to the left hand side. The type of the operator is preserved, and used as the row type in the final linear programme formulation.

The output operators (PRNT and DISP) serve no useful purpose in the factorising process, and hence are treated as no-operations (NOPs). The store operators (STOD and STON) acted initially as delete operators, but this was later altered as discussed in the next section.

To illustrate the process, consider again the previous example:

\[
\text{Max } 3 * X + Y * (2 * Z) - W;
\]

where

\[
Z : X / 2 - 1;
\]
\[
Y : 4
\]

In reverse Polish notation this becomes:

\[
\text{Max } 3 X * Y 2 Z * * + W -
\]
\[
\text{and } X 2 / 1 - Z :
\]
\[
4 Y :
\]

The process is shown in figure A7. The final form of the objective function expression is the item that remains on the top of stack at the end of the single pass. In the example, this has a value of:

\[
7 X - 8 - W
\]
<table>
<thead>
<tr>
<th>Operation</th>
<th>Environment Stack (position, length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>(1,1) (3,C)</td>
</tr>
<tr>
<td>X</td>
<td>(1,1) (3,1) (3,C) (1,X)</td>
</tr>
<tr>
<td>*</td>
<td>(1,1) (3,X)</td>
</tr>
<tr>
<td>Y 4</td>
<td>(1,1) (3,1) (3,X) (4,C)</td>
</tr>
<tr>
<td>Y</td>
<td>(1,1) (3,1) (3,X) (4,C) (A,Y)</td>
</tr>
<tr>
<td>:</td>
<td>(1,1) (3,1) (3,X) (4,C)</td>
</tr>
<tr>
<td>2</td>
<td>(1,1) (3,1) (3,X) (4,C) (2,C)</td>
</tr>
<tr>
<td>Z X</td>
<td>(1,1) (3,1) (3,X) (4,C) (2,C) (1,X)</td>
</tr>
<tr>
<td>2</td>
<td>(1,1) (3,1) (3,X) (4,C) (5,1) (2,C)</td>
</tr>
<tr>
<td>/</td>
<td>(1,1) (3,1) (3,X) (4,C) (2,C) (5,1)</td>
</tr>
<tr>
<td>1</td>
<td>(1,1) (3,1) (3,X) (4,C) (5,1) (1,C)</td>
</tr>
<tr>
<td>-</td>
<td>(1,1) (3,1) (3,X) (4,C) (5,1) (1,C)</td>
</tr>
<tr>
<td>Z</td>
<td>(1,1) (3,1) (3,X) (4,C) (2,C) (5,1)</td>
</tr>
<tr>
<td>:</td>
<td>(1,1) (3,1) (3,X) (4,C) (2,C) (5,1)</td>
</tr>
<tr>
<td>*</td>
<td>(1,1) (2,1) (3,2) (3,X) (4,C)</td>
</tr>
<tr>
<td>+</td>
<td>(1,2) (7,X) (-8,C)</td>
</tr>
<tr>
<td>W</td>
<td>(1,2) (3,1) (7,X) (-8,C) (1,X)</td>
</tr>
<tr>
<td>-</td>
<td>(1,2) (7,X) (-8,C) (-1,W)</td>
</tr>
</tbody>
</table>

**Notation:**

a) the environment stack is shown on the first line and takes the form (position, length).

b) the coefficient and tag stack components are shown on the second line as couples of form (coefficient, tag). A tag of "C" indicates a constant, and a tag of "A" indicates an address.

c) stack items containing more than one item are enclosed in braces.

**Figure A7: Single Pass Factorisation Sequence**
A4.4.3 Discussion of Single Pass Factorisation

The single pass algorithm proved superior to the multiple pass technique. The time taken to formulate the test linear programme was reduced from 120 seconds to just 40. However even this was much slower than was desirable, especially for a system designed to run interactively. The following refinements were made to improve the process:

1) The store operators were implemented in the factorising virtual machine. Whenever an item had been factorised into its components, it was stored in an array. A special tag value was implemented for the spread sheet, so that the position and size of the stored item could be retrieved easily by the LVAL operator. This meant that no item had to be calculated more than once when the matrix was being formulated.

2) The LVAL operator in the factorising virtual machine was altered so that when it calculated a value it first called up the simpler (and hence faster) substitution virtual machine. Thus items which could be solved directly were done on the faster machine, and only the factorising was done on the slower one.

The effect of these refinements was that the sample formulation took only about three seconds of process time. This is a four-hundred fold improvement over the first attempt, and is quite satisfactory both in terms of machine economy and interactive use.
Appendix B:
Stratagem
# Table of Contents

1 Concepts .................................................. 1
  1.1 Components ........................................ 2
  1.2 Use of Stratagem .................................... 3
  1.3 A Small Example .................................... 4

2 Syntax Diagrams .......................................... 7
  Example: ................................................ 7

3 System Input .............................................. 9
  Syntax: ................................................ 9
  Notes: ................................................ 10
  3.1 Comments ........................................... 11
  Example: ................................................ 11
  Notes: ................................................ 11
  3.2 Responding to Queries ............................... 12
  Syntax: ................................................ 12
  Notes: ................................................ 12

3.3 Modifiers ............................................... 13
<table>
<thead>
<tr>
<th>Stratagem</th>
<th>iii</th>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.10 TIMEWORD</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>3.11 HORIZONWORD</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>3.12 VARIABLES</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>3.13 CONSTRAINTS</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>4 DECLARATIONS</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>4.1 INDEX DECLARATION</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>4.2 VARIABLE DECLARATION</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>Stratagem</td>
<td>iv</td>
<td>Table of Contents</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----</td>
<td>-------------------</td>
</tr>
<tr>
<td>Syntax:</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>4.3 CONSTRAINT DECLARATION</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>4.4 SYNONYM SPECIFICATION</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>4.5 QUALIFICATION</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>5 DEFINITIONS</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>5.1 Definition Interrogation</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>5.2 VARIABLE PRIMARIES</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Specific Variable Primary:</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Stratagem</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Column Variable Primary:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Vector Variable Primary:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>General Variable Primary:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.3 CONSTRAINT PRIMARIES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific Constraint Primary:</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Column Constraint Primary:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Vector Constraint Primary:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>General Constraint Primary:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.4 VARIABLE AND CONSTRAINT DEFINITIONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Values</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Precedence of Definitions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entering and Changing a Definition</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Stratagem</td>
<td>vii</td>
<td>Table of Contents</td>
</tr>
<tr>
<td>-----------</td>
<td>-----</td>
<td>-------------------</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>59</td>
</tr>
<tr>
<td>6.1 DISCOUNT FUNCTION</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>6.2 NPV FUNCTION</td>
<td></td>
<td>61</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>61</td>
</tr>
<tr>
<td>6.3 SLDEPN FUNCTION</td>
<td></td>
<td>62</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>62</td>
</tr>
<tr>
<td>6.4 SLDEPNO FUNCTION</td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>6.5 SUM FUNCTION</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>One parameter:</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>Two Parameters:</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>Three Parameters:</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>7 INTERROGATION</td>
<td></td>
<td>66</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>66</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>8 Reports</td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>8.1 REPORT LAYOUT</td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>8.2 REPORT DEFINITION</td>
<td></td>
<td>71</td>
</tr>
<tr>
<td>Stratagem</td>
<td>Table of Contents</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>8.3 REPORT HEADER</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>8.4 REPORT PARAMETER LIST</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>8.5 REPORT BODY</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>Examples of &lt;DETAIL LINE&gt;s:</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>8.6 REPORT TAIL</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>8.7 REPORT INTERROGATION</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>Stratagem</td>
<td>ix</td>
<td>Table of Contents</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>8.8 Report Example</td>
<td></td>
<td>83</td>
</tr>
<tr>
<td>9 DECISIONS</td>
<td></td>
<td>85</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>85</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>86</td>
</tr>
<tr>
<td>9.1 OPTIONS</td>
<td></td>
<td>87</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>87</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>88</td>
</tr>
<tr>
<td>9.2 Decision and Option Examples:</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>9.3 Adding a Decision Option:</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>90</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>90</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>10 SCENARIOS</td>
<td></td>
<td>92</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>92</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>93</td>
</tr>
<tr>
<td>10.1 FUTURES</td>
<td></td>
<td>94</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td>94</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>95</td>
</tr>
<tr>
<td>10.2 Scenario and Future Examples.</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>10.3 Adding a Scenario Future.</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>Stratagem</td>
<td>Table of Contents</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>10.4 Respecifying a Probability</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>11 Augmented Definition</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>11.1 Changing an Augmented Definition.</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>12 Statements</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>12.1 STATEMENT INTERROGATION</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td>109</td>
<td></td>
</tr>
</tbody>
</table>
12.1 BYE STATEMENT
Syntax: 110
Example: 110

12.2 CHANGES STATEMENT
Syntax: 111
Example: 111

12.3 DOCUMENT STATEMENT
Syntax: 112
Example: 112

12.4 HORIZON STATEMENT
Syntax: 113
Example: 113
Notes: 113

12.5 IGNORE STATEMENT
Syntax: 115
Notes: 115
Example: 115

12.6 INITIALISE STATEMENT
Syntax: 116
Notes: 116
Examples: 116
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Syntax</th>
<th>Notes</th>
<th>Examples</th>
<th>Syntax</th>
<th>Notes</th>
<th>Examples</th>
<th>Syntax</th>
<th>Notes</th>
<th>Examples</th>
<th>Syntax</th>
<th>Notes</th>
<th>Examples</th>
<th>Syntax</th>
<th>Notes</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.8</td>
<td>MAXIMISE STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.9</td>
<td>MINIMISE STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.10</td>
<td>NAME STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.11</td>
<td>RISK STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.12</td>
<td>SAVE STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.13</td>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.14</td>
<td>USE STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table of Contents</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.15 VIEW STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 Reserved Words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 Appointments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 Provisions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Consequences</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 Delegation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 Preservation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stratagem
Stratagem is a user-friendly decision support system generator. It facilitates the construction and solution of corporate decision models. Features of the system include:

- **Non-procedural implementation.** The order in which the model logic is entered is irrelevant. The system automatically decides what it needs to calculate the answer to a specific request for information. This eliminates the need for line numbers and complex control logic within the model.

- **Incremental compiler/interpreter implementation.** As each model statement is entered it is automatically checked for correctness, and compiled into an intermediate code. This means that at all points in the development the model is free from syntax errors and ready to be used. Additionally, the compiler/interpreter makes the model run faster and use fewer resources than a system which interpretes the source code.

- **The elimination of unnecessary commands.** Once a model has been built a user can operate it using only a single command (to load the model). Commands such as RUN, EDIT, COMPILRE etc are not needed.

- **The ability to alter the vocabulary of the model.** The user is able to specify his own vocabulary, and to alter it as circumstances change. Thus the system adapts to the user, and not the other way round.

- **Advanced modelling features.** These include easy to use optimisation and risk analysis routines, as well as the automatic detection and solution of systems of linear simultaneous equations.
1.1 Components

A model is composed of:

- a set of time periods. The number of time periods in a model is specified by means of the HORIZON statement.
- a set of variables. All variables are declared before use by means of the VARIABLE statement.
- a set of variable definitions. These define the rules for calculating the value which is to be assigned to each variable in the model. The set of variable definitions defines the logic of the model.
- an optional set of constraints. A constraint is a logical variable that is principally used in optimisation. Each constraint must be declared before use by the CONSTRAINT statement.
- an optional set of constraint definitions. Each definition is composed of a logical relationship between variables, and may return a value of either TRUE or FALSE.
- an optional set of report definitions. These allow the model output to be presented in specialised formats. Each report definition consists of at least a name and a list of items (variables or constraints) which are to appear in the report. Reports are printed according to a standard format which may be modified by the use of special formatting directives if desired.
- an optional set of decisions. Each decision consists of a name and a set of options. Each option has a name and a set of "impacts". An impact represents the change that will occur in a model variable should that particular option be selected. Each option is assumed to come into effect at the end of the present period (period 0).
- an optional set of scenarios. A scenario, unlike a decision, represents events which are outside of the control of the user (such as the future interest or inflation rate). A scenario is similar to a decision in that it is composed of a set of "futures" which, like options, contain a set of "impacts". Additionally, a probability value may be associated with each future to indicate the likelihood of that future occurring.
1.2 Use of Stratagem

A Stratagem model is run by a series of interrogations. An interrogation may be made on any model item. There are two sorts of interrogation:

- the value of an item is interrogated by entering the item's name followed by a semicolon. The range of values which an item can assume is determined by the item's type. For example, the value of a variable is a number, the value of a constraint is TRUE or FALSE, and the value of a report is a display of the report.

- the structure of an item may be interrogated by entering the item's name followed by a question mark. The structure of a variable or constraint is the formula or relationship that defines its value, whereas the structure of a report is the list of variables and formatting directives that specify the report layout.

Variable constraint and report definitions are specified by entering the relationship that defines the value. If a definition already exists it is replaced by the new definition.

The possible impact of a decision or future state of the world may be modelled by means of the decision and scenario features. To alter the model so that it reflects the implementation of the decision or the occurrence of the future, it is necessary only to enter the keyword SELECT followed by the name of the option or future required. The model automatically adjusts so that it represents the situation that would occur should those events occur.

To print any item on the system printer (as opposed to the terminal), it is only necessary to enter the keyword PRINT before the input. In this way "hardcopy" listings of models, reports etc may be obtained.

To invoke the optimisation system, the keyword MAXIMISE (or MINIMISE) is entered followed by the objective function. This causes Stratagem to generate a linear programme of the model, call up an optimisation package to solve the model, and collect the results of the optimisation for further analysis.

To carry out risk analysis, the keyword RISK is entered followed by a list of the items for which the analysis is to be undertaken. This invokes a probabilistic type of analysis using each of the possible states of the model as defined by the combinations of the sets of futures. The key statistics of the resulting distributions are displayed.
1.3 A Small Example

In the following example all input supplied by the user is preceded by the prompt symbol used by Stratagem ("=>"). All other lines in the example are generated by Stratagem. The annotations down the right hand side of the page are for explanation only, and would not appear in a model.

```plaintext
=> HORIZON IS 2;
   3 PERIODS (0 - 2)
=> VARIABLE
   => SALES  "TOTAL SALES",
   => COSTS  "TOTAL COSTS",
   => PROFIT;
=> COSTS ARE 85% * SALES;
=> PROFIT IS SALES - COSTS;
=> SALES(0) ARE 100;
=> SALES ARE SALES(T-1) * 110%;
=> PROFIT;
   PERIOD  PROFIT
   0      15.00
   1      16.50
   2      18.15

=> SALES?
VARIABLE
   => SALES  TOTAL SALES
   => COSTS  "TOTAL COSTS"
=> PROFIT;
   PERIOD  PROFIT
   0      15.00
   1      16.75
   2      16.54

=> REPORT INCOME "INCOME STATEMENT";
   --- ENTER REPORT MODE ---
=> SALES;
   => "LESS: ", COSTS;
=> PROFIT;
```

set number of time periods to 3
(period 0 is the present period).

declare the model variables to use

a description of each variable
may be enclosed in quotation
marks.

specify the definitions that

make up the model logic.

initial data:

sales forecast.

interrogate the value of profit

system prints out the value for

each time period.

interrogate the definition for SALES.

system displays description ...

... and definitions.

enter revised sales forecast

system updates variables.

interrogate the value of PROFIT

define a report called INCOME.

system enters report mode.

report composed of SALES,

and COSTS;

and PROFIT.
```plaintext
= ENDREPORT;
---- REPORT COMPLETED ----
---- NEW DEFINITIONS ADDED ----
END OF REPORT MODE ----
= INCOME;

INCOME STATEMENT

| TOTAL SALES | 100.00 | 105.00 | 110.25 |
| LESS: TOTAL COSTS | 85.00 | 89.25 | 93.71 |
| PROFIT | 15.00 | 15.75 | 16.54 |

- Scenarios:
  - SALES GROWTH:
  - Low Growth:
    - Probability is 1/3,
    - Sales are Sales(T-1) * 105%
  - High Growth:
    - Probability is 1/3,
    - Sales are Sales(T-1) * 110%
  - No Growth:
    - Probability is 1/3,
    - Sales are Sales(T-1)

- End scenario:

- Select scenario:
  - NOGROWTH

- Select the future NOGROWTH:
  - NOGROWTH selected in SALES GROWTH;
  - Variables reinitialised from period 1

- Profit:
  - Period | Profit
  - 0     | 15.00
  - 1     | 15.00
  - 2     | 15.00

- Risk Profit(3):
  - Results of risk analysis:
  - Three possible future; three actual futures.
  - Only 3 futures for analysis
  - Expected value is 16.56
```

A Small Example
VARIANCE IS 2.48
STANDARD DEVIATION IS 1.58
RANGE IS FROM 15.00 TO 18.15
----- END OF RISK ANALYSIS ----
=> SELECT HIGROWTH
HIGROWTH SELECTED IN SALESGROWTH
REPLACES NOGROWTH
VARIABLES REINITIALISED FROM PERIOD 1
=> PRINT INCOME;
=> BYE;

select high growth scenario
print the income report for later reference and finish session.
2 SYNTAX DIAGRAMS

The structure of all input to Stratagem is described in the remainder of this manual in terms of syntax diagrams. All syntax diagrams used in this manual conform to the following conventions.

1) All items which are themselves represented by another syntax diagram are bracketed by the symbols "<" and ">". Items which are to be substituted directly; such as special symbols and keywords, appear in the diagrams with no brackets and (for keywords) in upper case. Descriptive items, such as that in the specification for a string, appear in lower case, and the user should specify something of his own for such items.

2) The item which a diagram is describing always appears on the left of the diagram, followed by a colon, and then the diagram itself.

3) The diagrams are based on a single "main route", which is a line starting at the colon and terminating with a vertical bar. When this route is too long to fit across the page, it is terminated with a ">" and started further down the page with a "<".

4) In many diagrams alternative routes are available. These are shown as branches off the main line. Branches beneath the line are travelled from left to right, and may only be entered from the left, whereas branches above the line are travelled from right to left and may only be entered from the right. The main route itself is always travelled from left to right.

5) Some branches may only be traversed a limited number of times. In such cases the maximum is indicated by the number enclosed in parentheses (i.e. "(" and ")")

Example:

The following set of diagrams describe a "jingle".

```
<---- + ----(3)---->

<JINGLE> : ----------------<ACTION PART>----------<TYPE>----- STAR ------1
```
<ACTION PART> : --------- SPARKLE ---------
    ----- TWINKLE ----

<TYPE> : --------------- LITTLE ---------------
    ----- DISTANT ----

This gives the following as the simplest form the jingle can take:

<ACTION>-----<TYPE>----- STAR ------

Substituting in for the <ACTION PART> means that either TWINKLE or SPARKLE may be used, and for <TYPE> either LITTLE or DISTANT can be used. This gives, for example, the following:

SPARKLE LITTLE STAR
TWINKLE LITTLE STAR
TWINKLE DISTANT STAR

However it is possible to repeat the <ACTION PART> up to three times by using the branch loop above it. This gives the following examples (amongst others) of a <JINGLE>:

TWINKLE, SPARKLE DISTANT STAR
SPARKLE, TWINKLE, TWINKLE, SPARKLE LITTLE STAR
TWINKLE, TWINKLE LITTLE STAR
3 **SYSTEM INPUT**

All input to the system is in "free format", with the following restrictions:

1) Input tokens (i.e. numbers, names, strings, etc) may not be split by the end of lines;

2) Spaces should not appear in the middle of any input token (with the possible exception of strings). Except where a space is used to separate two words, spaces have the usual literary significance: they separate items but don't mean anything;

3) Columns 73 to 80 of each input record are reserved for sequence numbers so that the individual input lines of a card or disc file may be numbered for reference or sequencing purposes;

4) A semicolon or, in some cases, a question mark indicates the end of a logical unit of input (such as a definition);

5) Comments may be included on any input line by preceding the comment with an exclamation mark.

In interactive mode, whenever the system is ready to receive an input from the user it displays a prompt of "=>". Up to ten input lines can be queued in anticipation of the prompt.

The structure of input to the system is given below. This does not include response to system generated queries (see Responding to Queries) or comments (see Comments).

**Syntax:**
<INPUT> : -------<MODIFIER LIST>-------<DECLARATION>-------
               |------<DEFINITION>------>
               |------<INTERROGATION>------>
               -------<STATEMENT>-------

>------------------------<BYE STATEMENT>------------------------

Notes:

1) The <BYE STATEMENT> terminates the programme and should only be entered at the end of a modelling session.

2) The <MODIFIER LIST> is used to direct output to the system printer or to cause a trace of the calculation sequence to be displayed.

3) The <DECLARATION> form of input is used to declare the variables, constraints, synonyms and indices used in the model.

4) The <DEFINITION> form of input is used to specify the definitions that describe model entities.

5) The <INTERROGATION> form of input is used to display the values or definitions of model items.

6) The <STATEMENT> form of input is used to control system actions and for specialised modelling requirements.
3.1 Comments

Comments may be included in a model for documentation purposes. The exclamation mark is used as a special character which tells the system to ignore the remainder of the input line. Thus any characters after the exclamation mark will be ignored by the system and may be treated as comment.

Example:

! THIS IS A COMMENT
! AND SO IS THIS.

Notes:

1) Characters which appear in a string are stored, but not inspected, by the system, so that an exclamation mark in a string does NOT indicate the start of a comment.

2) Comments entered into the terminal are printed onto the special print file (see PRINT modifier). In this way it is possible to annotate this output.

3) On some input devices, notably a card reader, the character codes are slightly different, so that the vertical bar must be used instead of the exclamation mark.
3.2 Responding to Queries

At certain times when solving a model, the system may discover that the definition required to calculate the value for a variable item has not been specified. In such cases the system will ask the user for the required value, and the user can:

a) enter an expression (NOT involving the item being requested) which is used to determine the item's value;

b) interrogate any constraint or variable value in the model by entering the keyword CALC followed by an expression.

Syntax:

```
<RESPONSE> : -----------------------<ARITHMETIC EXPRESSION>--------------
  |----- CALC --------------------<ARITHMETIC EXPRESSION>-----|
  |------<LOGICAL EXPRESSION>-----|
```

Notes:

1) No queries will be generated under the following conditions:
   - when operating in batch mode;
   - when generating a report;
   - when formulating a linear programme, where these items become decision variables;
   - when solving a set of simultaneous equations.

2) The CALC option causes the system to calculate and display the value of the expression. The system will then repeat the request for the value of the required variable.
3.3 Modifiers

There are two special modifiers which may be used before any statement or interrogation. These are:

PRINT — to cause the output to be directed to a special print file;

TRACE — to display the sequence of calculations.

The modifiers may be used alone or in combination with each other.

Syntax:

```
<MODIFIER LIST> : ------------------
                  |<---------<PRINT MODIFIER>-|
                  |<---------<TRACE MODIFIER>-|
```
3.4 PRINT MODIFIER

The PRINT modifier causes a copy of the results to be output to a special print file. In some cases where the output is likely to be of considerable length (such as report output or viewing the model definitions), the output to the terminal is suppressed. The print modifier is especially useful for printing reports.

Syntax:

<PRINT MODIFIER> : --------- PRINT ---------|

Notes:

1) When the PRINT modifier is used in conjunction with the USE statement, each record of input is displayed on the screen as well the print file.

2) When the PRINT modifier is used in specifying a variable or constraint definition, the print action is "compiled" into the definition, so that whenever the value is calculated it is also displayed.

Examples:

PRINT BALANCESHEET
forces the output from the report BALANCESHEET to be printed on the print file.

PRINT VIEW
forces the output from the VIEW statement to be diverted to the print file.

PRINT 1/3
causes the results of the expression to be displayed on the terminal and printed on the print file.
The TRACE modifier forces the name of each variable or constraint element to be displayed on the screen when it is being calculated. This is particularly useful in debugging models.

Syntax:

```
<TRACE MODIFIER> : ----- TRACE -------
```

Notes:

1) Care should be taken in using the TRACE modifier as it can result in a large quantity of output.

2) Use of the TRACE modifier in the MAXIMISE or MINIMISE statements results in a row by row display of the complete linear programming tableau.
### 3.6 CONSTANT

A constant is a sequence of digits with optional decimal part, exponent part, and percent sign, which represents a number. The exponent part represents the power of ten to which the number is raised. The maximum power to which a number can be raised is 30, and the minimum is -30. If the percent sign is present, the number is taken to represent a percentage, i.e., it is divided by 100.

**Syntax:**

```
<CONSTANT> :
  [<--(8)--->
   ----<DIGIT>-----------------------]<---<DECIMAL PART>--! !<EXPOSENT PART>----|-- %|--

<DECIMAL PART> : ----- . ----<DIGIT>------------------------

<EXPOSENT PART> : ----- ñ ----<DIGIT>---------------------
```
Notes:

1) All constants must fall in the range
   $-10^{\text{39}} < \langle \text{CONSTANT} \rangle < 10^{\text{39}}$

Examples:

```
123
1.23
123%
123\text{e}-2
0.0123\text{e}2
1\text{e}6
1\text{e}-6
0.01\%
```
Equivalent to 1.23
Equivalent to 1.23
Equivalent to 1.23
Equivalent to $10^{\text{000000}}$
Equivalent to 0.000001
Equivalent to 0.0001
3.7 IDENTIFIERS

An identifier is the name by which a model item is referred. They are the names given to items that comprise the model and to directives and commands to the modelling system. These latter identifiers have been pre-defined, and are listed in the Reserved Word Table (see Reserved Words). Identifiers which refer to the items which constitute the model must be defined by the user. In effect these specify the model vocabulary.

An identifier may only refer to a single item in the model (i.e., it must be unique). The use of synonyms however enables any item or directive to be referred to by more than one identifier (see SYNONYM SPECIFICATION).

Syntax:

```
<---<DIGIT>--->
|<---<LETTER>-----(19)---|
```

`<IDENTIFIER> = ---------<LETTER>--------------------------1`

Notes:

1) An identifier is composed of at least one letter, optionally followed by a sequence of up to 19 letters and/or digits.

2) The order of any characters after the first three and before the last two is not significant.

   e.g. BALANCESHEET is equivalent to BALACNEHSEET.
3.8 **STRINGS**

A string is a sequence of any valid string characters, enclosed in quotation marks, with a maximum length of 70 characters (excluding the quotation marks, but including all blanks). Strings are used to give model items a more meaningful description than can be achieved in the item's name (an identifier), and are hence useful for documentation.

**Syntax:**

```
<STRING> : ---- " " " ---- Valid String Character ---- " " ----1
```

**Notes:**

1) Valid string characters are letters, digits, special characters (such as commas, periods, colons, hyphens etc) and the space character (blank). The quotation mark is not a valid string character.

**Examples:**

```
"THIS IS A STRING"
"AND SO IS THIS"
"!@#$%(}=-@#%*:;?/.. ARE VALID STRING CHARACTERS"
"0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```
3.9 **FILENAME**

The syntax of any FILENAME specified to the system must be correctly formed according to standard Burroughs large systems conventions (see Burroughs B6700 Input/Output Subsystem Reference Manual).

**Syntax:**

```plaintext
<FILENAME> : ----------<FILE TITLE>--------------------------
           | -----------<STRING>-------
           | <------ / ------(16)------
<FILE TITLE> : ----------<FILE ROOT PART>---------------------
               | <-----<DIGIT>-----
               | <------<LETTER>--------(16)-----
<FILE ROOT PART> : ------<LETTER>--------------------------
```

**Notes:**

1) If the string form of the FILENAME is used, the string must contain only a valid Burroughs file title, optionally followed by a period.
3.10 TIMEDWORD

The TIMEDWORD, represented by the identifiers TIME and T, is a special identifier used to reference the time periods in a model. It may be used in qualifications and in arithmetic expressions. Its value is taken to be the number of the period presently under consideration. It is not possible for the user explicitly to set the value of the timeword; instead, the system infers the value when required from the context of the user supplied input.

Syntax:

<TIMEDWORD> : ------------------------- TIME -----------------------
                                     |
                                     |
                                     --------- T ---------
The HORIZONWORD, represented by the identifier HORIZON, is used to reference the model horizon, as specified by the HORIZON statement. It may be used in arithmetic expressions, and returns as a value the number of the period which is designated to be the model horizon.

Syntax:

<HORIZONWORD> : ------------------ HORIZON ------------------
3.12 VARIABLES

Variables are the basic elements in any model, and are declared in variable declaration statements. Whenever a variable is declared, the modelling system assigns a row of its "spread sheet" to the variable, so that a copy of the variable exists in each time period of the model. If the variable is declared to be indexed, the system assigns a row of the spread sheet to each element of the variable. Variable primaries are used to reference the variable in specific time periods.

Variables may be assigned values for each time period. The rules for calculating and assigning these values are specified by means of variable definition statements.

Syntax:

```
<VARIABLE> ::= <IDENTIFIER>-------|
```

Examples:

```
INCOME
NPAT
EBIT
TOTAL_EXPENSES
INFLATION
```
3.13 CONSTRAINTS

Constraints are logical variables used for storing the result of comparisons between variables and expressions. Whenever a constraint is declared, the modelling system assigns a row of its "spread sheet" to the constraint, so that a copy of the constraint exists in each time period of the model. If the constraint is declared to be indexed, the system assigns a row of the spread sheet to each element of the constraint. Constraint primaries are used to reference the constraint in specific time periods.

Constraints may be assigned values for each time period. The rules for calculating and assigning these values are specified by means of constraint definition statements. A constraint can take on a value of TRUE, FALSE, or, in some case, UNDEFINED.

Syntax:

<CONSTRAINT> : <IDENTIFIER>

Examples:

DEBTLIMIT
DIVGROWTH
MINCASH
4 DECLARATIONS

Declarations are used to specify the type and name of model items. All index, variable, constraint and report items must be declared explicitly before they are used.

Syntax:

```
<DECLARATION> :  ----------------<INDEX DECLARATION>--------- ; ---- |
          |  <VARIABLE DECLARATION>--->
          |  <CONSTRAINT DECLARATION>--->
          |  <SYNONYM SPECIFICATION>-----
```

Notes:

1) The declaration for reports, decisions, options, scenarios and futures are implicit in their definitions.
4.1 INDEX DECLARATION

Frequently it is desired to specify in a model a set of variables (or constraints) whose definitions have largely the same structure. Instances of this may occur for example when modelling several divisions that form a company, or a range of different products. Such sets of variables may be established by declaring a variable to be indexed (see Variable Declaration). An index is represented as a set of indices, and is declared by means of an index declaration.

Syntax:

```
<INDEX DECLARATION> : ----- INDEX ----<INDEX SET>----- : ------>

|<------------------------ , ------------------------>|

>-----------<INDEX>------------------------------------------------------|

|<DESCRIPTION>||
```

```
<INDEX SET> : -----------<IDENTIFIER>-------------|

<INDEX> : -----------<IDENTIFIER>-------------|

<DESCRIPTION> : -----------<STRING>-------------|
```

Notes:
1) The <INDEX> and <INDEX SET> must be otherwise undeclared identifiers.
2) When the model is index-based (see HORIZON statement), the <INDEX> is used as a qualification.
3) The maximum number of indices per <INDEX SET> is 15.
Examples:

INDEX PRODUCTS: WIDGETP, WIDGETU, "PACKAGED WIDGETS", "UNPACKAGED WIDGETS", NUTS, BOLTS;
INDEX REGIONS: NORTH, SOUTH, EAST, WEST, CENTRAL;
INDEX GROUPS: TEXTILES, CHEMICALS, PROPERTY;
INDEX DIVISIONS: WHOLESALE, RETAIL;
4.2 VARIABLE DECLARATION

The variable declaration is used to indicate that a variable or list of variables is to become part of the model vocabulary. Variables must be declared to be part of the vocabulary by this statement before they can be used.

Syntax:

```plaintext
<VARIABLE DECLARATION> ::= 

----- VARIABLE ----------
\|-----------------------|
\|<BLOCK NAME>\|<INDEX PART>\|

\|-----------------------|
\|\|---<DESCRIPTION>\|---|

<INDEX PART> ::= ------ OF ------<INDEX>------- |

<BLOCK NAME> ::= ---------<STRING>--------- |
```

Notes:

1) Each variable is allocated a row of the model "spread sheet" when it is declared, and thus can assume a unique value in each time period of the model. The values are specified by means of a variable definition statement. If a variable is declared to be indexed (i.e., the <INDEX PART> of the variable declaration is used) it is allocated a row for each element of the index, and thus takes the form of a matrix.

2) The <BLOCK NAME> part of the variable definition is used to specify that the subsequent
variables belong to a particular block. These blocks are used only when documenting the model (see DOCUMENT statement). If no <BLOCK NAME> is specified, the variables belong to the previous block (if any).

3) The <DESCRIPTION> is an optional string that is used to give a more meaningful description of the variable than can be achieved by its name. The main use of the <DESCRIPTION> is in printing reports and in documenting the model.

Examples:

```plaintext
VARIABLE 
  CAREXP  "EXPENSE ITEMS",
  OFFICEEXP  "OFFICE EXPENSES",
  DEPNEXP  "DEPRECIATION EXPENSE";

VARIABLE OF DIVISIONS
  SALES,
  EXPENSES,
  INCOME  "NET INCOME";
```
4.3 CONSTRAINT DECLARATION

The constraint declaration is used to indicate that a constraint or list of constraints is to become part of the model vocabulary. Constraints must be declared to be part of the vocabulary by this statement before they can be used.

Syntax:

<CONSTRAINT DECLARATION>:

| ------ CONSTRAINT ----------------------------- |
| [--------<BLOCK NAME>----] [--------<INDEX PART>-----] |
| [-----------------<DESCRIPTION>----------------] |

Notes:

1) The notes on VARIABLE DECLARATION should be read in conjunction with this section.
4.4 SYNONYM SPECIFICATION

Synonyms are used to augment the existing model vocabulary. Any part of the vocabulary may be referenced by one or more synonyms.

Syntax:

<SYNONYM SPECIFICATION> :

┌───────────────┐
│              │
│<SYNONYM>     │
│------------<SYNONYM>--------------<MODEL ITEM>--------------┐
│                    │
│<MODEL ITEM> : ----------<IDENTIFIER>--------------│
│                    │

Notes

1) The <SYNONYM> must not have been previously used in the model.
2) The <MODEL ITEM> must be a part of the existing model vocabulary (e.g. a variable name or a reserved word).
3) When an item has synonyms associated with it, the synonyms and the item may be used interchangeably and are logically identical.
Examples:

SYNONYM REP = REPORT,
SYN = SYNONYM;
 SYN TC = TOTALCOSTS;
SYNONYM NPAT = NETPROFITAFTERTAX;
4.5 QUALIFICATION

A variable or constraint can represent a set of values, each value corresponding to a particular time period. To reference a specific time period the variable or constraint must be qualified. Qualification immediately follows the item's name.

There are three types of qualification:

- Specific Qualification, referring to a stated time period;
- Relative Qualification, referring to a time period relative to the actual time period;
- Indexed Qualification, used only when it has been specified that the model is not time based (see HORIZON statement).

Syntax:

<QUALIFICATION> : <SPECIFIC QUALIFICATION> <RELATIVE QUALIFICATION> <INDEXED QUALIFICATION>

<SPECIFIC QUALIFICATION> : <CONSTANT>

<RELATIVE QUALIFICATION> :
    <TIMELIST> 

<INDEXED QUALIFICATION> : <INDEX>
Examples:

Specific: $(20)$  $(0)$  $(2)$
Relative: $(T)$  $(T-1)$  $(T-4)$
Indexed: $(NORTH)$  $(SOUTH)$  $(BOLTS)$
5 DEFINITIONS

A definition is the means by which the structure of a model entity is specified. The form of the definition is determined by the type of the model item. For example, a definition for a variable consists of an arithmetic expression, and that for a report consists of a list of variables, constraints and formatting directives.

Syntax:

<DEFINITION> : <VARIABLE DEFINITION> ;
<CONSTRAINT DEFINITION> ;
<REPORT DEFINITION> ;
<DECISION DEFINITION> ;
<OPTION DEFINITION> ;
<SCENARIO DEFINITION> ;
<FUTURE DEFINITION> ;
<AUGMENTED DEFINITION RESPECIFICATION> ;
<PROBABILITY RESPECIFICATION> ;
5.1 Definition Interrogation

Definition interrogations are used to display the definition of an item such as a report, scenario or variable.

Syntax:

<DEFINITION INTERROGATION> : <VARABLE> ?

<CONSTRAINT> <REPORT NAME> <DECISION NAME> <OPTION NAME> <SCENARIO NAME> <FUTURE NAME> <STATEMENT INTERROGATION>

Notes:

1) When using the system interactively, care should be taken not to place the question mark in the first column of an input line. A question mark in column one has special meaning to the programme which drives the terminal (as opposed to Stratagem itself), and all such input is intercepted by this programme (see Burroughs Large System Cande Reference Manual).
5.2 VARIABLE PRIMARIES

A variable primary is a reference to a specific variable. There are several possible types of variable primary, depending upon whether the variable has been declared to be indexed or not.

Specific Variable Primary:

A specific variable primary refers to a single variable in a specific time period, and hence a single value. If the variable is indexed, an index must be specified.

Syntax:

```plaintext
<SPECIFIC VARIABLE PRIMARY> : ----------------------------------------

----------<INDEX>---

>-----<VARIABLE>-----<SPECIFIC QUALIFICATION>-----
```

Column Variable Primary:

A column variable primary refers to an indexed variable in a specific time period, and hence represents a column of values. For variables that have not been declared to be indexed, the column variable primary is equivalent to the specific variable primary (the column has a length of one).
Syntax:

COLUMN VARIABLE PRIMARY : <VARIABLE> <SPECIFIC QUALIFICATION>

Vector Variable Primary:

A vector variable primary refers to a variable with a specific index, and represents a row of values.

Syntax:

VECTOR VARIABLE PRIMARY : <INDEX> <VARIABLE>

General Variable Primary:

A general variable primary refers to a variable only, with no index or qualification specified. If the variable has been declared to have an index, the general variable primary represents a matrix of values, otherwise it represents a row.

Syntax:

GENERAL VARIABLE PRIMARY : <VARIABLE>

When evaluating a variable primary in an expression, the system must be able to resolve the primary into one representing a single value (i.e., a specific variable primary). The correct index and qualification to use are inferred from that of the subject of the expression (i.e., the left hand side).
Examples:

SALES
SALES(9)
RETAIL.SALES
WHOLESALE.SALES(3)

<GENERAL VARIABLE PRIMARY>
<COLUMN VARIABLE PRIMARY>
<vector variable primary>
<specific variable primary>
5.3 CONSTRAINT PRIMARIES

A constraint primary is a reference to a specific constraint. Constraint primaries are similar to variable primaries. A constraint primary can assume a value of TRUE, FALSE or UNDEFINED.

Specific Constraint Primary:

A specific constraint primary refers to a constraint in a specific time period, and hence a single value. If the constraint is indexed, an index must be specified.

Syntax:

\[<\text{SPECIFIC CONSTRAINT PRIMARY}> : \text{---------------------------}> \]
\[\text{-----<INDEX>----} \]
\[<\text{CONSTRAINT}>-----<\text{SPECIFIC QUALIFICATION}>----------]\n
Column Constraint Primary:

A column constraint primary refers to an indexed constraint in a specific time period, and hence represents a column of values. For constraints that have not been declared to be indexed, the column constraint primary is equivalent to the specific constraint primary (the column has a length of one).
A vector constraint refers to a constraint with a specific index, and represents a row of values. A vector constraint is specified by an expression of the form:

\[ \text{<VECTOR CONSTRAINT PRIMARY>} : \text{<INDEX>} \]

A general constraint refers to a constraint without an index or qualification. A general constraint is specified by an expression of the form:

\[ \text{<GENERAL CONSTRAINT PRIMARY>} \]

When evaluating a constraint in an expression, the system must be able to resolve the index and qualification to use. The correct syntax is:

\[ \text{<GENERAL CONSTRAINT PRIMARY>} : \text{<INDEX>} \]

This ensures that the constraint is applied correctly to the appropriate row of values.
Stratagem

Examples:

DIVGROWTH
ORDINARY.DIVGROWTH
DIVGROWTH(2)
PREFERENCE.DIVGROWTH(1)
5.4 VARIABLE AND CONSTRAINT DEFINITIONS

The rules that define many of the functions and uses of constraint definitions are largely the same as those for variable definitions. Therefore, unless otherwise stated, the following discussion will refer to both variable and constraint definitions.

Values

Definitions are used to define values for model entities. The range of values possible for a particular entity depends upon its type (i.e., variable, constraint).

A variable may assume a value of any real number, or it may be undefined. The latter case may occur in one of the following situations:

1) The variable has no definition;

2) The definition for the variable cannot be resolved (it may, for example, contain an illegal operation such as divide by zero, or be composed of items which are themselves undefined);

3) The variable is explicitly defined to be "UNDEFINED".

A constraint can take one of three values. It can be TRUE, indicating that the defining relationship holds; it can be FALSE, indicating that the relationship does not hold; or, like a variable, it can be undefined.

Precedence of Definitions

It is possible for several different definitions to refer to the same variable or constraint item. As an item may logically be defined only one way at any one time, a set of simple precedence rules are applied to ensure that a unique definition for an item is found. The precedence hierarchy is as follows:
1) specific definitions (i.e., single items which are defined in a specific time period);
2) column definitions (i.e., an indexed item which is defined in a specific time period);
3) vector definitions (i.e., an element of an indexed item which is defined for all time periods);
4) general definitions (i.e., a variable defined for all time periods);
5) replacement definitions (see Augmented Definitions).

Note that items 2 and 3 on the hierarchy only apply to indexed variables.

If an item is defined in more than one way, the definition which has the highest precedence is that which, if it exists, is used to calculate the value of the item.

Example 1:
Consider the (unindexed) variable TOTALCOSTS, and the following definitions:

a) TOTALCOSTS ARE 30;
   (general definition)
b) TOTALCOSTS(0) ARE 25;
   (specific definition)
c) TOTALCOSTS(3) ARE 35;
   (specific definition)

These result in the following set of values for TOTALCOSTS:

<table>
<thead>
<tr>
<th>Period</th>
<th>TOTALCOSTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
</tr>
</tbody>
</table>

Example 2:
Consider the indexed variable SALES with indices RETAIL and WHOLESALE, and the following definitions:

a) SALES ARE 100;
   (general definition)
b) SALES(1) ARE 110;
   (column definition)
c) RETAIL.SALES ARE 105;
   (vector definition)
d) WHOLESALE.SALES(2) ARE 120;
   (specific definition)
This gives the following table for SALES, where the letter indicates the definition which took precedence in the calculation.

<table>
<thead>
<tr>
<th></th>
<th>PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALES</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(c) 105</td>
</tr>
<tr>
<td>1</td>
<td>(b) 110</td>
</tr>
<tr>
<td>2</td>
<td>(c) 105</td>
</tr>
<tr>
<td>3</td>
<td>(d) 105</td>
</tr>
<tr>
<td>RETAIL</td>
<td></td>
</tr>
<tr>
<td>(a) 100</td>
<td></td>
</tr>
<tr>
<td>(b) 110</td>
<td></td>
</tr>
<tr>
<td>(d) 120</td>
<td></td>
</tr>
<tr>
<td>WHOLESALE</td>
<td>(a) 100</td>
</tr>
</tbody>
</table>

Entering and Changing a Definition

Definitions may be entered and modified at almost any time in the modelling process. To specify a definition it is entered according to certain rules of syntax (see sections on Variable Definitions and Constraint Definitions). If a definition is entered for an item for which a definition already exists, then the old (existing) definition is replaced by the one just entered. The effect of this may be not only to alter the item's definition and value, but also to change the value of other items in the model. If such is the case, the modelling system automatically reinitialises the variables to preserve the integrity of the model.

Definitions may be removed from the model by entering a null (or empty) definition. The system responds with a message telling whether the definition was removed, and if necessary reinitialises the variables.

An item may be defined to be UNDEFINED by a definition. In this case the item is treated as a decision variable, so that when using the item, the system will request the user to supply its value (see Responding to Queries). When the MAXIMISE or MINIMISE statements are used to generate a linear programme, such items become decision variables in the linear programme.
5.5 VARIABLE DEFINITIONS

Variable definitions are used to describe the model logic, and hence are the driving force of the model.

Syntax:

```
<VARIABLE DEFINITION> : ----------------<SPECIFIC VARIABLE DEFINITION>----------------
                      |----------------<COLUMN VARIABLE DEFINITION>----------------|
                      |----------------<VECTOR VARIABLE DEFINITION>----------------|
                      |----------------<GENERAL VARIABLE DEFINITION>----------------|

<SPECIFIC VARIABLE DEFINITION> :
    ---<SPECIFIC VARIABLE PRIMARY>---<ASSIGNMENT OPERATOR>-----
                      |----------------<ARITHMETIC EXPRESSION>----------------|
                      |----------------<CONDITIONAL ARITHMETIC EXPRESSION>-----|

<COLUMN VARIABLE DEFINITION> :
    ------<COLUMN VARIABLE PRIMARY>-----<VARIABLE DEFINITION PART>-----|

<VECTOR VARIABLE DEFINITION> :
    ------<VECTOR VARIABLE PRIMARY>-----<VARIABLE DEFINITION PART>-----|
```
<GENERAL VARIABLE DEFINITION>:
-----<GENERAL VARIABLE PRIMARY>-----<VARIABLE DEFINITION PART>-----

<VARIABLE DEFINITION PART>:
-------------------<ASSIGNMENT OPERATOR>-------------------

>-------------------<ARITHMETIC EXPRESSION>-------------------

|-------------------<CONDITIONAL ARITHMETIC EXPRESSION>-------------------|

|-------------------<NULL DEFINITION>-------------------|

---------- UNDEFINED ----------

<VNULL DEFINITION>:
---------------------------------------------------------------------

<ASSIGNMENT OPERATOR>:
-------------------

|-------------------|

| IS --->|

| ARE --->|

Notes:

1) No variable may be specifically defined in terms of itself, or in terms of a future value of any variable. Thus

SALES ARE 110% * SALES
SALES(2) ARE 110% * SALES(3)

are both invalid. This means that no variable on the left side of the <ASSIGNMENT OPERATOR> may appear on the right side, unless the time period on the right side is less than the time period on the left, or the INDEX on the right side is different from that on the left.

2) If no qualification is given on the left side of the <ASSIGNMENT OPERATOR> (i.e., the definition is neither a <SPECIFIC DEFINITION> nor a <COLUMN DEFINITION>), then the only specific qualification that may appear on the left side is one referring to period zero. For example:
COSTS ARE 80% ≠ SALES(3)
is not valid because COSTS in periods zero to two will be defined in terms of a future value.
Note however that although definitions of the form
SALES ARE 110% ≠ SALES(0)
are acceptable, an error will occur if the value is not specifically defined in period zero.

3) If no qualification or index is given in a <VARIABLE PRIMARY> on the right of the <ASSIGNMENT OPERATOR>, the qualification or index used on the left is assumed. Variables which are declared to be indexed must have an index specified if the <INDEX SET> for that variable is different from that of the variable on the left of the <ASSIGNMENT OPERATOR>.

4) If a definition is already present for a variable (i.e. the item on the left of the <ASSIGNMENT OPERATOR>), then the new definition replaces the existing definition.

5) The <NULL DEFINITION> causes the existing definition, if any, to be replaced.

6) The order of precedence for calculating the values for a variable is given in the PRECEDENCE OF DEFINITIONS section.

7) The UNDEFINED part of the definition causes the item to become a decision variable.

Examples:

WHOLESALE.SALES(0) ARE 59; 80; 32;  <SPECIFIC VARIABLE DEFINITION>
RETAIL.SALES ARE WHOLESALE.SALES(T-1) ≠ 2;  <VECTOR VARIABLE DEFINITION>
SALES(0) ARE 100;  <COLUMN VARIABLE DEFINITION>
SALES ARE 110% ≠ SALES(T-1);  <GENERAL VARIABLE DEFINITION>

TOTALCOSTS(3) ARE 92.5% ≠ (RETAIL.SALES + WHOLESALE.SALES);  <SPECIFIC DEFINITION>
TOTALCOSTS(2) ARE ;  <SPECIFIC DEFINITION>
TOTALCOSTS(0) ARE UNDEFINED;  <SPECIFIC DEFINITION>
TOTALCOSTS ARE TOTALCOSTS(T-1) ≠ 111%;  <GENERAL DEFINITION>
5.6 CONSTRAINT DEFINITIONS

Constraint definitions are used to describe relationships between sets of variables. The definitions are constructed as logical expressions, and hence return a value of either TRUE or FALSE (or, in some cases, UNDEFINED). Constraint definitions are handled in the same manner as variable definitions.

Syntax:

<CONSTRAINT DEFINITION> : <SPECIFIC CONSTRAINT DEFINITION>--------------
       |<COLUMN CONSTRAINT DEFINITION>------>
       |<VECTOR CONSTRAINT DEFINITION>------>
       |<GENERAL CONSTRAINT DEFINITION>-----

<SPECIFIC CONSTRAINT DEFINITION> :
       <SPECIFIC CONSTRAINT PRIMARY>---<ASSIGNMENT OPERATOR>------>
       |<LOGICAL EXPRESSION>-----------------------

<COLUMN CONSTRAINT DEFINITION> :
       <COLUMN CONSTRAINT PRIMARY>---<CONSTRAINT DEFINITION PART>----

<VECTOR CONSTRAINT DEFINITION> :
       <VECTOR CONSTRAINT PRIMARY>---<CONSTRAINT DEFINITION PART>----
<GENERAL CONSTRAINT DEFINITION>:

---<GENERAL CONSTRAINT PRIMARY>---<CONSTRAINT DEFINITION PART>---

<CONSTRAINT DEFINITION PART> :

------<ASSIGNMENT OPERATOR>---------

>-----------------<LOGICAL EXPRESSION>-----------------------

| <CONDITIONAL LOGICAL EXPRESSION>---

| <NULL DEFINITION>-------

| UNDEFINED ------------

Notes:

1) The notes under Variable Definitions should be read in conjunction with this section.
2) Constraints are binding only when optimising (see MAXIMISE and MINIMISE) statements.

Examples:

DEBTLIMIT = DEBT < 40% * TOTALASSETS;  
MINCASH(1) = CASH > CURRENTLIABILITIES/3;  
MINCASH = CASH > 0;  
DIVGROWTH = DIV > 105% * DIV(T-1)
5.7 ARITHMETIC EXPRESSION

An arithmetic expression is a rule for calculating a (real) number, and is more or less composed of a sequence of arithmetic primaries separated by arithmetic operators.

Syntax:

<ARITHMETIC EXPRESSION> :

\[
\begin{align*}
\text{<---<ARITHMETIC OPERATOR>---|} \\
\text{<---<UNARY OPERATOR>---|} \\
\text{<ARITHMETIC PRIMARY>---|} \\
\text{<VARIBALE PRIMARY>---|} \\
\text{<---<CONSTANT>---|} \\
\text{<---<ARITHMETIC EXPRESSION>---|} \\
\text{<---<FUNCTION PRIMARY>---|} \\
\text{<---<HORIZONWORD>---|} \\
\text{<---<TIMEWORD>---|} \\
\text{<UNARY OPERATOR> : \quad \text{<--- + ---|}} \\
\end{align*}
\]

ARITHMETIC EXPRESSION
<ARITHMETIC OPERATOR> : 

\[ \begin{align*} 
\text{+} & \quad \text{+} \\
\text{-} & \quad \text{-} \\
\text{*} & \quad \text{*} \\
\text{/} & \quad \text{/} 
\end{align*} \]

Notes:

1) The <ARITHMETIC OPERATORS> have conventional meaning as follows:

   + addition
   - subtraction
   * multiplication
   / division

   The sequence of operations described by an arithmetic expression is determined by the precedence of operators as follows:

   1) * -
   2) * /
   3) unary operator

   Operators with the highest precedence apply first. When operators have the same precedence, the sequence is determined by the order of their appearance (from left to right). Parentheses up to any depth may be used to alter the order of precedence, as in normal algebraic fashion. Items enclosed in parentheses are calculated first.

Examples:

\[
\begin{align*}
1 + 3 * 2 &= 7 \\
(1 + 3) * 2 &= 8 \\
-8 * 2 &= -16 \\
7 + 3 * 2 - 1 &= 12 \\
(7 + 3) * (2 - 1) &= 10 \\
(7 + 3) * 2 - 1 &= 19
\end{align*}
\]
5.8 LOGICAL EXPRESSION

A logical expression is a rule for calculating a (logical) value, and is composed of two arithmetic expressions separated by a relational operator. Logical expressions return a value of TRUE, FALSE, or UNDEFINED, and are typically used in definitions for constraints.

Syntax:

\[
\text{LOGICAL EXPRESSION} : \quad \quad \text{ARITHMETIC EXPRESSION} \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \q
Examples:

\[
\begin{align*}
3 &< 2 \quad \text{(Result) False} \\
3 &> 3 \quad \text{True} \\
3 &> 2 \quad \text{True} \\
(1 + 2) \times 3 &= 1 + 2 \times 3 \quad \text{False}
\end{align*}
\]
5.9 **CONDITIONAL EXPRESSIONS**

Conditional expressions are used when the exact definition for an item depends upon the state of some other item or items in the model. There are two types of conditional expression: arithmetic, which returns a real value, and logical, which returns a logical value.

**Syntax:**

```
<CONDITIONAL LOGICAL EXPRESSION> : ---<IF CLAUSE>--- THEN ------>
>-------------------<LOGICAL EXPRESSION>------------------- ELSE ------>
   |<CONDITIONAL LOGICAL EXPRESSION>|
   >-------------------<LOGICAL EXPRESSION>-------------------|
   |<CONDITIONAL LOGICAL EXPRESSION>|

<CONDITIONAL ARITHMETIC EXPRESSION> : ---<IF CLAUSE>--- THEN --->
>-------------------<ARITHMETIC EXPRESSION>------------------- ELSE --->
   |<CONDITIONAL ARITHMETIC EXPRESSION>|
   >-------------------<ARITHMETIC EXPRESSION>-------------------|
   |<CONDITIONAL ARITHMETIC EXPRESSION>|

<IF CLAUSE> : ------ IF ------<LOGICAL EXPRESSION>------|
    |<CONSTRAINT PRIMARY>------|
```
Notes:

1) In evaluating a conditional expression, the <IF CLAUSE> is evaluated first, and if it is TRUE, the "THEN expression" is evaluated and returned as the expression result, otherwise the "ELSE expression" is evaluated and returned. Note that every "IF" must have an accompanying "THEN" and "ELSE", and that conditional expressions can be nested to any depth.

Examples:

IF A < 4 THEN 12 ELSE 13
IF A < 4 THEN C > B ELSE C > D
IF A < 4 THEN
  IF B > 10 THEN 12 * (B + D)
  ELSE IF B > 5 THEN 12 * D
  ELSE 12 * B
ELSE D * B
IF TAXCONSTRAINT(T-1) THEN A = B
ELSE A < B

{arithmetic}
{logical}
{nested arithmetic}
{logical}
FUNCTION PRIMARY

There are some special functions available to aid model building. These may be used anywhere a `<VARIABLE PRIMARY>` can be used.

Syntax:

```
<FUNCTION PRIMARY> : <---<FUNCTION NAME>---<FUNCTION PARAMETER LIST>----->

<FUNCTION PARAMETER LIST> :

    |<---------------- , ------------|

    |<----------------<FUNCTION PARAMETER>------------->|

<FUNCTION PARAMETER> : <---------<ARITHMETIC EXPRESSION>-------->
6.1 **DISCOUNT FUNCTION**

The discount function returns the value of an item discounted to the present period. It has three parameters:

- **Parameter 1**: The item which is to be discounted;
- **Parameter 2**: The discount rate, expressed as a percentage;
- **Parameter 3**: The time period from which the item is to be discounted.

The discounting is done with the following formula:

Let \( T \) be the current period,
\( H \) be the period from which to discount.

Then \( \text{DISCOUNT}(A, R, H) \) gives:

- if \( H < T \) error
- \( H = T \) \( A(H) \)
- \( H > T \) \( \frac{A(H)}{(1 + R(T)) \times (1 + R(T+1)) \times \ldots \times (1 + R(H))} \)

where \( X(j) \) is the value of \( X \) in period \( j \)

**Example:**

\[ \text{DISCOUNT(PROFIT, 10\%, T+3)} \]
Discount the value of \( \text{PROFIT} \) in the third period from now at 10\% per period.

\[ \text{DISCOUNT(CAPEX, INFLATION, HORIZON)} \]
Discount the value of \( \text{CAPEX} \) at the model horizon at the \( \text{INFLATION} \) rate.
6.2 NPV FUNCTION

The NPV function calculates the present value of an item. There are three parameters:

Parameter 1: The item for which to calculate the present value;
Parameter 2: The discount rate, expressed as a percentage;
Parameter 3: The time period from which to calculate the present value.

The calculation is done as follows:

Let T be the current period,
    H be the period from which to calculate the present value.

Then NPV(A, R, H) gives

if H < T    error
    H = T    A(H)

H > T    Sum from t = T to H of DISCOUNT(A, R, t)

where A(H) is the value of A in period H.

Examples:

NPV(CASHFLOW, INFLATION+2%, HORIZON)
    present value of CASHFLOW discounted at
    a rate 2% higher than the inflation rate
    from the model horizon.
6.3 **SLDEPN FUNCTION**

This function returns the straight line depreciation calculated for a series of capital expenditure items. The item is not depreciated in its year of purchase. The function has two parameters:

- **Parameter 1:** The amount expended on the item;
- **Parameter 2:** The number of periods over which the item is to be depreciated. The item has no residual value.

The calculation is done as follows:

Let \( T \) be the current period, then \( \text{SLDEPN}(A, L) \) returns:

\[
\text{sum from } t = 0 \text{ to } T-1 \text{ of} \\
\quad \text{if } L(t) + t > T \text{ then } A(t) / L(t) \\
\quad \text{else } 0. \\
\text{where the inequality reads STRICTLY GREATER THAN.}
\]

**Examples:**

\[
\text{SLDEPN(CARCAPEX, 4)} \\
\text{Straight line depreciation of cars (as represented by their capital expenditure) over a life time of four years.}
\]
6.4 SLOEPNO FUNCTION

This function calculates the straight line depreciation of capital expenditure items in the same the SLOEPN function, except that the depreciation is calculated for the current period. For a discussion of the parameters refer to the SLOEPN function.

The calculation is done as follows:

Let \( T \) be the current period, then \( \text{SLOEPNO}(A, L) \) returns:

\[
\text{sum from } t = 0 \text{ to } T \text{ of } \begin{cases} 
A(t) / L(t) & \text{if } L(t) + t < T \\
0 & \text{otherwise}
\end{cases}
\]

where the inequality reads STRICTLY LESS THAN.

Examples:

\[
\text{SLOEPNO(CARCAPEX, 4)}
\]

Straight line depreciation of cars over four years including the current period.
6.5 SUM FUNCTION

The SUM function is used to sum an expression over time. The number of parameters determines the range of the summation:

One parameter:

Example: \( \text{SUM}(A) \)

Returns: If \( T = 0 \) then error
otherwise the sum from \( t = 1 \) to \( T \) of \( A(t) \)
where \( T \) is the current period.

Two Parameters:

Example: \( \text{SUM}(A, H) \)

Returns: If \( H(T) = 0 \) then error
otherwise the sum from \( t = 1 \) to
minimum(\( H(T) \), \( \text{HORIZON} \)) of \( A(t) \)
where \( T \) is the current period.

Three Parameters:

Example: \( \text{SUM}(A, I, H) \)

Returns: If \( H(T) < I(T) \) then error
otherwise the sum from \( t = \text{maximum}(0, H(T)) \)
to \( \text{minimum}(H(T), \text{HORIZON}) \) of \( A(t) \)
where \( T \) is the current period.
Examples:

\[
\text{SUM(PROFIT, TIME)}
\]
sum of PROFIT from period 1 to the present period. This is equivalent to \text{SUM(PROFIT)}.

\[
\text{SUM(DIVIDENDS, HORIZON)}
\]
sum of DIVIDENDS from period 1 to the model horizon.

\[
\text{SUM(INTEREST+TAX, T-2, T-1)}
\]
sum of the previous two periods interest and tax.
7 INTERROGATION

Interrogation is the means by which the user enquires as to the value or definition of a model item. When running a model, input will usually take the form of an interrogation.

Syntax:

<INTERROGATION> : ----------<VALUE INTERROGATION>--------------
| | ----<DEFINITION INTERROGATION>---- |

<VALUE INTERROGATION> : --------------<VARIABLE>-------------- | |
| | ------<CONSTRAINT>------ | |
| | ------<REPORT NAME>------ | |
| | ------<DECISION NAME>------ | |
| | ------<OPTION NAME>------ | |
| | ------<SCENARIO NAME>------ | |
| | ------<FUTURE NAME>------ | |
| | ------<INTERROGATION LIST>--- | |
<INTERROGATION LIST> :  <ARITHMETIC EXPRESSION> ; --- | <LOGICAL EXPRESSION> | <VARIABLE PRIMARY> | <CONSTRAINT PRIMARY> |

Notes:

1) The <INTERROGATION LIST> allows a list of expressions to be entered and the results to be displayed in tabular form on the screen.

2) The <VARIABLE> and <CONSTRAINT> form of value interrogation cause the value of the items to be calculated (if necessary) and displayed.

3) The <REPORT NAME> form of value interrogation causes the specified report to be prepared and displayed.

4) The scenario, future, decision and name forms of the value interrogation display the status of the item (i.e. selected or ignored).

5) The <DEFINITION INTERROGATION> causes the definition of the item to be displayed (see Definition Interrogation section).
A report specifies how data are to be output to the terminal or the printer. In its simplest form, a report is merely a list of variables and constraints. The basic layout is the same for all reports, although optional formatting controls are available if required.
8.1 REPORT LAYOUT

All reports are produced on a "virtual page" with the following layout:

<table>
<thead>
<tr>
<th>Item Description</th>
<th>Field 1</th>
<th>...</th>
<th>Field n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column m</td>
<td>Column 1</td>
<td></td>
<td>Column m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

The report heading contains a user-specified heading for the report, the time and date when printed, and the page number. If any options or scenarios have been selected when the report is prepared, their descriptions are also displayed in the report heading.

The item description contains the name or description of the variable or constraint whose value is being displayed on the corresponding report line.

Each field in the report represents a particular time period in the model, or if indices are being used instead of time periods, a particular index. The fields each contain a list of values, each of which corresponds to the item description on the same line of the report. These
values may be arranged into columns in the usual accounting manner.

The description width is the number of characters that will be printed to make up an item description. This has a default value of 14. The fieldwidth depends upon the number of columns per field (default is 2) and the width of each column (default is 11 characters).

Typically the virtual page onto which the report is printed is too big to be displayed on either a terminal or a line printer (whose widths are usually only 80 and 132 characters respectively). This problem is overcome by splitting the virtual pages into actual pages, each one of which can be accommodated horizontally on the target display device. The split is done where possible at a field boundary, and each actual page has its own heading and item descriptions. Note that there is no guarantee that an actual page will fit vertically onto the output page (i.e. there may be more lines on an actual page than on a physical page of the output device).
8.2 REPORT DEFINITION

A report definition consists of three parts: a report header, a report body, and a report tail. The report header specifies the name of the report, and can contain details that control the layout of the report on the output device. The report body is a list of variables or constraints which are to be output as the report, and the report tail signifies the end of the report definition.

Syntax:

```plaintext
<REPORT DEFINITION> : <REPORT HEADER>----------------<REPORT TAIL>--
                  |                                      |
                  |                                      |
                  |<REPORT BODY>--
```

-71-
8.3 REPORT HEADER

The <REPORT HEADER> specifies the name by which the report is to be referred, and (optionally) a heading for the report and any modifications that are to be made to the virtual page layout.

Syntax:

<REPORT HEADER> : REPORT ---<REPORT NAME>-------------------------->
                     |                                          |
                     |                                          |
                     |<REPORT HEADING>---

>------------------------------------------------------------------:
                                                                                                                                 |
                     |<REPORT PARAMETER LIST>--

<REPORT NAME> : ----<IDENTIFIER>-------
<REPORT HEADING> : ----<STRING>------

Examples:

REPORT PROFIT;
REPORT BALANCE "BALANCE SHEET";
REPORT INCOME (COLUMNS = 1);
Notes:

1) The <REPORT NAME> is any identifier which has not previously been declared, or the name of a previously defined report. In the latter case, the new report specification overwrites the old specification.

2) The <REPORT HEADING> is a string which contains the main heading for the report. This is printed at the top of the displayed report.

3) The <REPORT PARAMETER LIST> is used to modify the default layout of the report. It is discussed in the next section.
8.4 REPORT PARAMETER LIST

The use of a report parameter list allows the user to override the system default specifications for the report layout. The items which determine the report layout are shown below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Specification</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characters per line</td>
<td>PAGEWIDTH</td>
<td>No greater than 132.</td>
</tr>
<tr>
<td></td>
<td>80 for terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>132 for line printer</td>
<td></td>
</tr>
<tr>
<td>Characters in description</td>
<td>DESCRIPTION</td>
<td>No greater than 50.</td>
</tr>
<tr>
<td>Number of fields per page</td>
<td>FIELDS</td>
<td>Cannot be set greater than 10.</td>
</tr>
<tr>
<td>of display</td>
<td></td>
<td>May be overridden by the system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if incompatible with other parameters</td>
</tr>
<tr>
<td>Columns per field</td>
<td>COLUMNS</td>
<td>No greater than 7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Characters per column</td>
<td>WIDTH</td>
<td>No greater than 20.</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Decimal places</td>
<td>DECIMAL</td>
<td>No greater than 6.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Result scaling</td>
<td>SCALE</td>
<td>No less than -10 and no greater than 10.</td>
</tr>
<tr>
<td>by power of 10</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Fields in the report</td>
<td>TIME</td>
<td>Any range up to the model horizon.</td>
</tr>
<tr>
<td></td>
<td>ALL</td>
<td></td>
</tr>
</tbody>
</table>
Syntax:

\[ \text{<REPORT PARAMETER LIST>} : \]

\[
\begin{align*}
\text{<REPORT PARAMETER LIST> :} & \quad \text{<CONSTANT>} \quad \text{<INDEX>} \\
\text{<TIMEDWORD> :} & \quad \text{<CONSTANT>} \\
\text{<DESCRIPTION> :} & \quad \text{<INDEX>} \\
\text{<FIELDS> :} & \quad \text{<INDEX>} \\
\text{<DECIMAL> :} & \\
\text{<WIDTH> :} & \\
\text{<COLUMN> :} & \\
\text{<PAGEWIDTH> :} & \\
\end{align*}
\]

Example:

\[
\begin{align*}
\text{WIDTH} & = 15, \quad \text{DECIMAL} = 4 \\
\text{DESCRIPTION} & = 20 \\
\text{T} & = 3 \\
\text{SCALE} & = -3, \quad \text{TIME} = 3-5
\end{align*}
\]
Notes:

1) When the report contains items that are indexed, copies of the report are produced for all indices within the <INDEX SET>. The INDEX form of the parameter may be used to specify a report for a single index value.
8.5 REPORT BODY

The <REPORT BODY> contains a list of items which are to be printed in the report, along with optional formatting directives. Each list item is separated from other items by a semicolon, and refers to a line of output in the report.

Syntax:

```
<REPORT BODY> : <HEADING LINE> <EMPTY LINE> <FIELD LINE> <DETAIL LINE> ;

<HEADING LINE> : HEADING <HEADING TEXT>

<EMPTY LINE> : LINE <SKIP COUNT>

<FIELD LINE> : FIELD
```
Notes:

1) The `<HEADING LINE>` causes a heading to be printed. The line consists of the text given in the `<HEADING TEXT>`, and is automatically centred on the output page. For example: HEADING "RESULTS FOR DIVISION B";
HEADING "PERFORMANCE RATIOS";

2) The `<EMPTY LINE>` causes blank lines to be printed on the output device. The number of blank lines is given by the `<SKIP COUNT>`.* If no `<SKIP COUNT>` is specified, one blank line is printed. For example: LINE;
LINE 3;

3) The `<FIELD LINE>` causes a heading to be printed on each field in the report. If the model is time-based, the heading consists of the period number for this column. If the model is indexed-based, it consists of the name of the index to which the column pertains. For example: FIELD;

4) The `<DETAIL LINE>` represents a single data line in the report.

5) The `<INDENT>` part will cause the description for this line to be indented three spaces. The `<LINE DESCRIPTION>` is placed in the description field of the report. It is truncated if necessary.

6) The `<VARIABLE>`, `<CONSTRAINT>`, or `<SUMCOL>` are the items whose value is to be printed out in the designated `<COLUMN>`. The description of the variable or constraint (see Variable Description) is printed in the description field after the `<LINE DESCRIPTION>` if there is sufficient room, otherwise the name of the item is shown. If no `<COLUMN>` is specified, column one is assumed.

7) The `<SUMCOL>` item places the sum of the column specified in brackets into the designated `<COLUMN>` (column one if none is specified). At this stage the sum for the column is reset to zero.

8) The `<FORMAT>` has the following effects:

- suppress the printing of the item's description and name in the description line;
- treat the item as a percentage. Its value is multiplied by 100 and it is printed out with two decimal places, with no scaling.
% print the value with no decimal places or scaling

9) The <COLUMN> signifies the column in the field in which to place the value. There are seven columns in each field, with the leftmost column numbered 1 and the rightmost numbered 7 in the usual accounting manner. If no column is specified, column 1 is assumed. When more than one column is used in any one line, only the <LINE DESCRIPTION> and description associated with the item in the left most column is used in the item description field of the page.

10) The <UNDERLINE> option causes the value in the column to be underlined.

Examples of <DETAIL LINE>s:

```plaintext
SALES;
"LESS: ", OTHERCOSTS 2;
ROI %;
SUMCOL(2) 1 UNDERLINE;
```

Print value of SALES in column 1.
Print ROI as a percentage in col 1.
Print the sum of all items in column 2 in column 1 underlined.
8.6 REPORT TAIL

The end of the report is marked by the single keyword ENDREPORT. This signifies that the end of the report definition has been reached. If no errors have been detected the report definition is saved.

Syntax:

<REPORT TAIL> : ------ ENDREPORT ------ }

Example:

ENDREPORT

Notes:

1) If the <REPORT END> is omitted, all subsequent input will be treated as if it were part of the report.
8.7 REPORT INTERROGATION

A report may be displayed by entering its name. The title and page layout may also be altered from that specified in the report by including a <REPORT PARAMETER LIST>. If it is desired to print a report on the printer, the PRINT modifier should be used.

Syntax:

<REPORT INTERROGATION> : ---<REPORT NAME>------------------------->
    |---<REPORT HEADING>---|
    >-------------------------|
    |---<REPORT PARAMETER LIST>---|

Examples:

PROFIT
PROFIT "MODIFIED INFLATION REVENUE"
PROFIT "WORST CASE PROFIT" (COLUMNS = 1)

Notes:

1) The <REPORT HEADING>, if specified, is used instead of that in the <REPORT HEADER>.
2) Any parameters specified in the <REPORT PARAMETER LIST> override those specified in the <REPORT HEADER>. 
8.8 Report Example

REPORT PROFITREPORT "PROFIT REPORT"
  (COLUMNS = 2, DECIMAL = 0);
  "FOR YEAR ENDED 30TH JUNE", YEAR#;
  LINE;
  SALES;
  "LESS: " LABOUR #;
  " " OTHERCOSTS #, "TOTAL COSTS" SUMCOL(2) UNDERLINE;
  NPBT;
  "LESS " TAX;
  LINE;
  NPAT UNDERLINE;
  ENDREPORT;

The input PROFITREPORT results in the following output:

<table>
<thead>
<tr>
<th>Model name</th>
<th>PROFIT REPORT</th>
<th>Time &amp; Date</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR YEAR ENDING 30TH JUNE</td>
<td>xxxxx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SALES</td>
<td>xxxxx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LESS: LABOUR</td>
<td>xxxxx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LESS OTHER COSTS</td>
<td>xxxxx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NET PROFIT BEFORE TAX</td>
<td>xxxxx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LESS TAX</td>
<td>xxxxx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NET PROFIT AFTER TAX</td>
<td>xxxxx</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Input of the form

```
PROFIREPORT (COLUMNS = 1);
```

results in the following

<table>
<thead>
<tr>
<th>model name</th>
<th>PROFIT REPORT</th>
<th>time &amp; date</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR YEAR ENDING 30TH JUNE</td>
<td>XXXX</td>
<td>XXXX</td>
<td></td>
</tr>
<tr>
<td>SALES</td>
<td>XXXX</td>
<td>XXXX</td>
<td></td>
</tr>
<tr>
<td>TOTAL COSTS</td>
<td>XXXX</td>
<td>XXXX</td>
<td></td>
</tr>
<tr>
<td>NET PROFIT BEFORE TAX</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LESS TAX</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NET PROFIT AFTER TAX</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9 DECISIONS

A decision represents a set of mutually exclusive courses of actions (options) which are available to the decision maker. Each option in the decision represents the impact on the remainder of the model that will result should that option be taken. The impact is expressed as a set of definitions which define new values for the affected variables.

Syntax:

<DECISION DEFINITION> :

    ---<DECISION HEAD>------------------------<DECISION TAIL>---

    ---<DECISION BODY>---

<DECISION HEAD> : --- DECISION ---<DECISION NAME>----------------- ; ---

    ---<DESCRIPTION>---

<DECISION NAME> : ---<IDENTIFIER>-------

<DECISION BODY> : ---------<DECISION OPTION>--- ; ---------

<DECISION TAIL> : --- ENDDECISION ------
 Notes:

1) It is assumed that the decision is made after period zero but before period one.

2) It is assumed that decisions are made before the scenario futures occurs (see Scenarios).

3) There is no practical limit on the number of decisions used in a model, nor on the number of options in each decision.

4) Decisions, once specified, will be ignored by the system if the "ignore" statement is used.

5) The <DECISION NAME> must be unique.

6) The <DESCRIPTION> in the <DECISION HEAD> is used only for documentation.

7) When a decision is being specified, the system enters a "structure definition mode". While in this mode all input is assumed to pertain to the decision, and the only valid form of input is that describing an option. The mode is terminated only when the <DECISION TAIL> is entered, at which point a message to this effect is displayed on the terminal.
9.1 OPTIONS

An option represents a specific course of action within a decision. All options in the same
decision are mutually exclusive, although combinations of options from different decisions are
quite legitimate. Each option is composed of a set of variable definitions which specify the
impact of the variables if that option is specified.

Syntax:

<DECISION OPTION> :

---<OPTION HEAD>-------------------<OPTION TAIL>--- ; ---|

|<OPTION BODY>|---

<OPTION HEAD> : --- OPTION ---<OPTION NAME>------------------- ; ---|

|<DESCRIPTION>|---

<OPTION NAME> : ---<IDENTIFIER>-------|

|<AUGMENTED DEFINITION>|---

<OPTION BODY> : -------<AUGMENTED DEFINITION>--- ; -------|

<OPTION TAIL> : ------- END OPTION -------|
Notes:

1) The <OPTION NAME> must be unique.

2) The <DESCRIPTION> in the <OPTION HEAD> is printed as part of the heading of any report printed when that option is selected (see Select statement).

3) The <OPTION TAIL> terminates the specification of the option.
9.2 Decision and Option Examples:

```
DECISION EXPANSION "LEVEL OF EXPANSION";
  OPTION BIGPLANT "BUILD BIG PLANT";
    INCREASE CAPACITY BY 1500;
    INCREASE CAPEX BY 2500;
  END OPTION;
  OPTION SMALLPLANT "BUILD SMALL PLANT";
    INCREASE CAPACITY BY 500;
    INCREASE CAPEX BY 1500;
  END OPTION;
  OPTION NOPLANT "NO EXPANSION";
END DECISION;
```

```xml
<DECISION HEAD>
  <OPTION HEAD>
    <IMPACT DEFINITION>
    <OPTION TAIL>
  <OPTION HEAD>
    <IMPACT DEFINITION>
    <OPTION TAIL>
  <OPTION HEAD>
    <IMPACT DEFINITION>
    <OPTION TAIL>
<DECISION TAIL>
```
9.3 Adding a Decision Option:

An option may be added to a previously defined decision.

Syntax:

```
<OPTION DEFINITION> :
  ---<DECISION OPTION HEAD>------------------------<OPTION TAIL>---|
  |<OPTION BODY>---|

<DECISION OPTION HEAD> : --- OPTION ---<OPTION NAME>--- IN ------>
                       >---<DECISION NAME>------------------------ ; ---|
                                    |<DESCRIPTION>---|
```

Notes:

1) The <DECISION NAME> refers to the decision to which the option is to be added.
2) The <OPTION NAME> must not have been previously used.
3) The system enters "structure definition mode" when an option is being added, so that all variable or constraint definitions entered become part of the option. The mode is terminated when the <OPTION TAIL> is entered, at which point a message is displayed to this effect.
Example:

```plaintext
OPTION MEDIUMPLANT IN EXPANSION
    "BUILD MEDIUM PLANT":
    INCREASE CAPACITY BY 1000;
    INCREASE CAPEX BY 2000;
ENDOPTION;

<DECISION OPTION ...>
    <IMPACT DEFINITION>
    <IMPACT DEFINITION>
    <OPTION TAIL>
```
10 SCENARIOS

A scenario is a set of possible future states of the world (futures), only one of which may occur. Each future in the Scenario represents the impact on the remainder of the model that will result should that state of the world occur. A scenario differs from a decision in that the decision maker has no control in an actual situation over which future will occur, and hence each future in Stratagem has a probability of occurrence associated with it.

Syntax:

<SCENARIO DEFINITION> :

    ---<SCENARIO HEAD>--------------------------<SCENARIO TAIL> ---
    
    |                   ---<SCENARIO BODY>--- |

<SCENARIO HEAD> : --- SCENARIO ---<SCENARIO NAME>-------------------;--- |
                    |<DESCRIPTION>--- |

<SCENARIO NAME> : -----<IDENTIFIER>----- |

<SCENARIO BODY> : -----<SCENARIO FUTURE>--- ; ----- |

<SCENARIO TAIL> : ---- ENDSCENARIO ---- |
Notes:

1) The scenario is assumed to commence after period zero but before period one.
2) Scenarios are assumed to occur after any decisions have been taken.
3) There is no practical limit on the number of scenarios used in a model, nor on the number of futures in each scenario.
4) Scenarios, once specified, will be ignored by the system if the "ignore" statement is used.
5) The <SCENARIO NAME> must be unique.
6) The <DESCRIPTION> in the <SCENARIO HEAD> is used only for documentation.
7) When a scenario is being specified, the system enters "structure definition mode". While in this mode all input is assumed to pertain to the scenario, and the only valid form of input is that describing a future. The mode is terminated only when the <SCENARIO TAIL> is entered, at which point a message to the effect is displayed on the terminal.
10.1 FUTURES

A future represents one possible state of the world within a scenario. All futures in the same scenario are mutually exclusive, although combinations of futures from different scenarios are quite legitimate. Each future is composed of a set of variable definitions which specify the impact of the variables should that future occur.

**Syntax:**

<SCENARIO FUTURE> : ---<FUTURE HEAD>------------------------<FUTURE TAIL>---

| ---<FUTURE BODY>--- |

<FUTURE HEAD> : --- FUTURE ---<FUTURE NAME>------------------- ; --- |

| ---<DESCRIPTION>--- |

<FUTURE NAME> : ------<IDENTIFIER>------ |

| ------------------ |

<FUTURE BODY> : ---<AUGMENTED DEFINITION>--- ; ------------------- |

| ---<PROBABILITY DEFINITION>--- |

<FUTURE TAIL> : ------ ENDOFUTURE ------ |

<PROBABILITY DEFINITION> :

--- PROBABILITY ---<VARIABLE DEFINITION PART>--- |

| ---<NULL DEFINITION>--- |
Notes:

1) The `<FUTURE NAME>` must be unique.

2) The `<DESCRIPTION>` in the `<FUTURE HEAD>` forms part of the heading of any report printed when that future is selected.

3) The `<FUTURE TAIL>` terminates the specification of the option.

4) The `<PROBABILITY DEFINITION>` is used to indicate the probability of that future occurring instead of any other future in the scenario. Any general `<VARIABLE PRIMARY>`s that are contained in the `<ARITHMETIC EXPRESSION>` that defines the probability value are assumed to refer to the period one value of the variable. When risk analysis is being used, the sum of the probabilities of each each future must be one for each scenario (see Risk statement).
10.2 Scenario and Future Examples.

SCENARIO INTERESTRATES "INTEREST RATE FORECASTS";
  FUTURE LOWINTEREST "LOW INTEREST RATES";
    PROBABILITY IS 30%;
    INTERESTRATE(1) IS 17%, 16%, 15%, 15%, 13%;
  ENDFUTURE;
  FUTURE EXPINTEREST "EXPECTED INTEREST RATES";
    PROBABILITY IS 70%;
    INTERESTRATE(1) IS 17%, 17.5%, 17.5%,
                      17%, 16%;
  ENDFUTURE;
ENDSCENARIO;

SCENARIO INFLATIONRATES "INFLATION RATE FORECASTS";
  FUTURE LOWINFLATION "LOW INFLATION RATE";
    PROBABILITY IS 0.25;
    INFLATION(1) IS 16.5%, 15%, 15%, 14%, 14%;
  ENDFUTURE;
  FUTURE EXPINFLATION "EXPECTED INFLATION RATE";
    PROBABILITY IS 0.75;
    INFLATION(1) IS 17%, 17.5%, 17%, 16%, 15%;
  ENDFUTURE;
ENDSCENARIO;
10.3 Adding a Scenario Future.

A future may be added to a previously defined scenario.

Syntax:

\[
\text{<FUTURE DEFINITION>} : \quad \text{---<SCENARIO FUTURE HEAD>-------------<FUTURE TAIL>---I}
\]

\[
\text{---<FUTURE BODY>---}
\]

\[
\text{<SCENARIO FUTURE HEAD>} : \quad \text{--- FUTURE ---<FUTURE NAME>--- IN -------} \\
\text{>---<FUTURE TAIL>------------------------ i; ---I}
\]

\[
\text{---<DESCRIPTION>---}
\]

Notes:

1) The \text{<SCENARIO NAME>} refers to the scenario to which the future is to be added.
2) The \text{<FUTURE NAME>} must not have been used previously.
3) The system enters "structure definition mode" when adding the future, so that all variable or constraint definitions entered become part of the new future. The mode is terminated when the \text{<FUTURE TAIL>} is entered, and a message to this effect is displayed on the terminal.
4) It may be necessary to respecify the probability of the other futures in the scenario so that they all sum to one.
Example:

FUTURE HIGHINTEREST IN INTEREST RATES
   "HIGH INTEREST RATES";
   PROBABILITY IS 10%;
   INTERESTRATE(1) IS 18%, 18.5%, 20%, 22%, 21%;
ENDFUTURE;

! <SCENARIO FUTURE ...
   ** HEAD>
   ** BODY>
! <FUTURE TAIL>
10.4 Respecifying a Probability

The probability of a future occurring may be respecified. This might be necessary when new information becomes available, or when a new future has been added to the scenario.

Syntax:

<PROBABILITY RESPECIFICATION> :

--- PROBABILITY --- OF ---<FUTURE NAME>---<VARIABLE DEFINITION PART>---

Notes:

1) The <FUTURE NAME> refers to the future whose probability is to be respecified.
2) When a <NULL DEFINITION> is used in the variable definition part, the existing probability definition is removed.

Examples:

PROBABILITY OF HIGHINTEREST IS 5%.
PROBABILITY OF LOWINTEREST IS IF INFLATION < 16.5% THEN 60% ELSE 30%;
PROBABILITY OF EXPINTEREST IS IF INFLATION < 16.5% THEN 35% ELSE 65%;
11 AUGMENTED DEFINITION

An augmented definition is one which is used in either an option or a future. The definition is used when the option or future is selected. There are two forms of augmented definition:

1) Replacement Definition: The augmented definition is a complete definition in its own right, and completely redefines the variable or constraint.

2) Impact Definition: The augmented definition increases or decreases the value of the variable.

Syntax:

<AUGMENTED DEFINITION> : ---------<REPLACEMENT DEFINITION>---------

|---------<IMPACT DEFINITION>---------|

<REPLACEMENT DEFINITION> : ---------<VARIABLE DEFINITION>---------

|-------<CONSTRAINT DEFINITION>-------|

<IMPACT DEFINITION> : ---------<SPECIFIC IMPACT DEFINITION>---------

|-------<GENERAL IMPACT DEFINITION>-------|

<SPECIFIC IMPACT DEFINITION> :

--------- INCREASE ---------<SPECIFIC VARIABLE PRIMARY>--- BY ---------

|------- DECREASE -------|

|<---------------------- , ----------------------|

|----------<ARITHMETIC EXPRESSION>----------|
<GENERAL IMPACT DEFINITION>:

-------- INCREASE --------< VARIABLE PRIMARY>---------- BY -------->

!------ DECREASE ------!

!------------------< ARITHMETIC EXPRESSION>------------------!

|<CONDITIONAL ARITHMETIC EXPRESSION>| |

----------------< NULL DEFINITION>----------------

Notes:

1) Because futures and options take effect between periods zero and one, it is not possible to redefine variable or constraint values in period zero.

2) When <REPLACEMENT DEFINITION>s are specified for the same variable (or constraint), in more than one option or future, care should be taken to select only those combinations of options and futures which contain a single definition of the value. When a variable or constraint is redefined more than once in a selected combination, a conflict occurs in choosing which definition to use, and an error message is generated.

3) <REPLACEMENT DEFINITION>s in selected options and futures are used instead of any definitions with the same precedence specified elsewhere in the model. The precedence of the replacement definitions is:

5.1) specific replacement definitions;
5.2) column replacement definitions;
5.3) vector replacement definitions;
5.4) general replacement definitions.

The definition with the highest precedence determines the value for the variable or constraint in the usual manner.

5) All <IMPACT DEFINITIONS> pertaining to the current period that are found in the selected options and futures are used. Thus it is quite feasible to mix both specific and general
<IMPACT DEFINITIONS>. The impacts specified by these definitions are calculated, and are added to the value as determined by the variable definition (or replacement definition) to give the actual value for the variable.

Examples:

INCREASE SALES BY 10% * SALES(T-1);  ! <GENERAL IMPACT DEFINITION>
DECREASE TAX(2) BY 12000;  ! <SPECIFIC IMPACT DEFINITION>
SALES(1) ARE 2000, 3000, 4000;  ! <REPLACEMENT DEFINITION>

Consider a model with the following definition:
SALES ARE 100;
and a selected option which contains the following augmented definitions:
INCREASE SALES BY 50;
DECREASE SALES(2) BY 30;

This gives the following values for SALES:

PERIOD  SALES
0        100
1        150
2        120
3        150
4        150

...
11.1 Changing an Augmented Definition.

Augmented definitions are those definitions which appear in futures or options. These may be altered or removed, or new definitions may be added, for a specific future or scenario.

Syntax:

<AUGMENTED DEFINITION RESPECIFICATION>:

----------<SPECIFIC AUGMENTED DEFINITION RESPECIFICATION>----------1

<!--------<NON-SPECIFIC AUGMENTED DEFINITION RESPECIFICATION>--------1

<SPECIFIC AUGMENTED DEFINITION RESPECIFICATION>:

----------<SPECIFIC REPLACEMENT DEFINITION RESPECIFICATION>----------1

!-------<SPECIFIC IMPACT DEFINITION RESPECIFICATION>-------1

<SPECIFIC REPLACEMENT DEFINITION RESPECIFICATION>:

<!--------<SPECIFIC VARIABLE PRIMARY>---<STRUCTURE PART>---<ASSIGNMENT OPERATOR>--->

<--------------------------,--------------------------1

>--------------------------<ARITHMETIC EXPRESSION>--------------------------1

!--------<CONDITIONAL ARITHMETIC EXPRESSION>---1
<SPECIFIC IMPACT DEFINITION RESPECIFICATION> :
     ----- INCREASE ------<SPECIFIC VARIABLE PRIMARY>---<STRUCTURE PART>---->
     ——— DECREASE ———

     [<-------------------------, ------------------------>]

     >------ BY ----------------<ARITHMETIC EXPRESSION>--------------------->
     ———<CONDITIONAL ARITHMETIC EXPRESSION>——

<NON-SPECIFIC AUGMENTED DEFINITION RESPECIFICATION> :
     ----------------------<REPLACEMENT DEFINITION RESPECIFICATION>-------------------
     ———<IMPACT DEFINITION RESPECIFICATION>——

<REPLACEMENT DEFINITION RESPECIFICATION> :
     ----------------<REPLACEMENT VARIABLE DEFINITION RESPECIFICATION>----------
     ———<REPLACEMENT CONSTRAINT DEFINITION RESPECIFICATION>——

<REPLACEMENT VARIABLE DEFINITION RESPECIFICATION> :
     ------<VARIABLE PRIMARY>---<STRUCTURE PART>---<ASSIGNMENT OPERATOR>------->

     >----------------<ARITHMETIC EXPRESSION>-------------------------->
     [---<CONDITIONAL ARITHMETIC EXPRESSION>---]
     [----------------<NULL DEFINITION>------------------>
     [---------------- UNDEFINED ---------------------------]
Stratagem

<REPLACEMENT CONSTRAINT DEFINITION RESPECIFICATION> :

---<CONSTRAINT PRIMARY>---<STRUCTURE PART>---<ASSIGNMENT OPERATOR>-----

>----------------------------------------<LOGICAL EXPRESSION>-------------------

| ---<CONDITIONAL LOGICAL EXPRESSION>--->
| -----------------------------------
| ---<NULL DEFINITION>----------------
| -----------------------------------
| UNDEFINED ------------------------
| -----------------------------------

<IMPACT DEFINITION RESPECIFICATION> :

------ INCREASE ------<VARIABLE PRIMARY>---<STRUCTURE PART>--- BY ------

\[ --- DECREASE \]

>--------------------------------<ARITHMETIC EXPRESSION>---------------------

| ---<CONDITIONAL ARITHMETIC EXPRESSION>--->
| --------------------------------------
| ---<NULL DEFINITION>----------------
| --------------------------------------
| UNDEFINED --------------------------
| --------------------------------------

<STRUCTURE PART> : ------ IN ------<FUTURE NAME>---------

\[ ---<OPTION NAME>\]

Notes:

1) If the definition, as identified by the variable or constraint primary, is already present in the option or future, it is replaced by the respecified definition. If the definition is not present in the option or future, the new definition is added to the option or future. If the <NULL DEFINITION> form is used, the existing definition for the variable or constraint is removed from the option or scenario.
Examples:

INCREASE CAPEX IN MEDIUM PLANT BY 1200;
INTEREST(4) IN HIGH INTEREST IS 19%, 19.5%;
12 STATEMENTS

A statement is a special directive to Stratagem. There are a variety of different statements, and the appropriate section should be consulted for further information.

Syntax:

```
<STATEMENT> : -------------------<BYE STATEMENT>------------------- ; -------
|------<CHANGES STATEMENT>------>
|------<DOCUMENT STATEMENT>------>
|------<HORIZON STATEMENT>------>
|------<IGNORE STATEMENT>------>
|------<INITIALISE STATEMENT>------>
|------<MAXIMISE STATEMENT>------>
|------<MINIMISE STATEMENT>------>
|------<NAME STATEMENT>------>
|------<RISK STATEMENT>------>
|------<SAVE STATEMENT>------>
|------<SELECT STATEMENT>------>
|------<USE STATEMENT>------>
|------<VIEW STATEMENT>------>
```
12.1 **STATEMENT INTERROGATION**

Apart from the exceptions noted below, the result of a statement interrogation is a short display of the purpose of the statement.

**Syntax:**

```
<STATEMENT INTERROGATION> : ------------------ CHANGES ------------------

----- DOCUMENT ----->
----- HORIZON ----->
----- IGNORE ----->
----- INITIALISE ----->
----- MAXIMISE ----->
----- MINIMISE ----->
----- NAME ----->
----- RISK ----->
----- SAVE ----->
----- SELECT ----->
----- USE ----->
----- VIEW ----->
```
Notes:

1) Interrogation on the following statements has a different effect, and the appropriate section in the manual should be referred to.

   CHANGES  ... display changes in the model
   HORIZON  ... display the model horizon
   NAME     ... display the model name

2) The BYE statement may not be interrogated.
12.2 **BYE STATEMENT**

The **BYE** statement is used to terminate a modelling session so that control of the terminal is returned to the MCS (usually SYSTEM/CANDE) if interactive, or to the originating batch job. Before terminating, all changes that have been recorded by the **CHANGES** statement are automatically printed.

**Syntax:**

```
<BYE STATEMENT> : -------------- BYE ---------------]
```

**Example:**

```
enter:       BYE
response:  System prints out any changes recorded,
            then displays

            --- END OF STRATAGEM ---
            control is then returned to the MCS.
```
12.3 CHANGES STATEMENT.

The CHANGES statement is used to tell the system to record any changes made to the variable and constraint definitions. The purpose of this statement is to allow the documentation of the model to be updated easily and accurately.

The use of the CHANGES statement causes a system switch to be toggled and its resulting value displayed. Changes to the model will only be recorded when the option is set. The option should be set for the first time only after all the existing model definitions have been read into the system by means of the USE statement.

The recorded changes may be viewed at any time by interrogating the CHANGES statement. When the BYE statement is entered, the recorded changes are automatically printed out.

Syntax:

<CHANGES STATEMENT> : -------------- CHANGES ---------------

Examples:

enter: CHANGES;
response: CHANGES WILL BE RECORDED

enter: CHANGES?
response: a display of all variable and constraint definitions changed.

enter: PRINT CHANGES?
response: a printout of all variable and constraint definitions changed.
12.4 DOCUMENT STATEMENT

The DOCUMENT statement is used to prepare output suitable for use in subsequent documentation of the model. The output consists of:

1) a list of all variables, their descriptions and definitions, organised according to their <BLOCK NAME> (see VARIABLE DECLARATIONS);
2) a list of all constraints, their descriptions and definitions, organised according to their <BLOCK NAME> (see CONSTRAINT DECLARATIONS);
3) a list of all reports and their structure;
4) a list of all decisions and their structure;
5) a list of all scenarios and their structure;
6) a list of all synonyms.

The documentation is always printed on the output print file and cannot be shown on the terminal. Thus the PRINT modifier has no effect on this statement.

Syntax:

<DOCUMENT STATEMENT> : ----------------- DOCUMENT ----------------- |

Example:

enter: DOCUMENT
response: a printed list of the complete model as described above.
12.5 **HORIZON STATEMENT**

The HORIZON statement is used to set the model horizon. By default the horizon is set to the maximum number of periods permitted in the model (currently 20).

In certain applications it may be desired not to have a time-based model. In such situations the horizon can either be set to zero, or to an `<INDEX SET>`. The latter indicates that the columns are to be represented by the indices contained in the `<INDEX SET>`, so that variables and constraints are qualified using these indices (see QUALIFICATION).

**Syntax:**

```plaintext
<HORIZON STATEMENT>:
    ---- HORIZON ----<ASSIGNMENT OPERATOR>--------<CONSTANT>-------\n    |\n    ----<INDEX SET>----
```

**Examples:**

```
HORIZON IS 10;
HORIZON IS REGIONS;
HORIZON : 3;
```

**Notes:**

1) If the horizon is to be other than time-based, this must be indicated by using the `<INDEX SET>` form of specification before any definitions have been entered.

2) An interrogation on the HORIZON statement will cause the type and range of the horizon to be displayed.
Example: Enter: HORIZON?
Response: 4 PERIODS (0 - 3)
12.6 **IGNORE STATEMENT**

The `IGNORE` statement is used to tell the system to ignore a particular decision or scenario. Any option or future that is selected in the decision or scenario will no longer have any effect on the model. The `SELECT` statement is used to reverse the effect of the `IGNORE` statement.

**Syntax:**

```
<IGNORE STATEMENT> : ------ IGNORE ---------<DECISION NAME>---------|
                   |                             |
                   |------<SCENARIO NAME>------|
```

**Notes:**

1) If the scenario or decision, which is being ignored has a selected future or option, the variables are reinitialised from period one.

2) When the `SELECT` statement is used to restore a scenario or decision, any option or scenario which was selected when the item was ignored is reinstated.

**Example:**

Enter:  IGNORE INTEREST RATES;
Response: TREE WILL BE IGNORED
12.7 **INITIALISE STATEMENT**

The initialise statement forces the system to reinitialise all variables from the period specified. If no period is given, period zero is assumed.

**Syntax:**

```
<INITIALISE STATEMENT> : -------- INITIALISE -------------------------
                           \-------------------------
                           \-----<CONSTANT>-----
```

**Notes:**

1) In some situations, such as when a variable is redefined or a new future is selected, the system will reinitialise the variables to preserve the integrity of the model.

**Examples:**

Enter: INITIALISE;
Response: VARIABLES REINITIALISED FROM PERIOD 0

Enter: INITIALISE 5;
Response: VARIABLES REINITIALISED FROM PERIOD 5
12.8 MAXIMIZE STATEMENT

The MAXIMIZE statement is used to invoke the linear programming subsystem. This causes the <ARITHMETIC EXPRESSION> in the statement to become the objective function of a linear programme of the decision variables, and the mathematical programming system TEMPO is initiated. Upon completion of TEMPO, the results are loaded into the model for further analysis.

Syntax:

<MAXIMIZE STATEMENT> : ------- MAXIMIZE -------<ARITHMETIC EXPRESSION>-------

Notes:

1) The <ARITHMETIC EXPRESSION> must not involve any vectors or variables with relative qualifications (i.e., it must be composed of <SPECIFIC VARIABLE PRIMARY>s).

2) Any variables encountered without definitions become decision variables. If no such variables exist, the solution to the linear programme is trivial. The model ought to contain at least one decision variable.

3) The model must include some defined constraints; otherwise the linear programme will be unbounded.

4) The linear programme generated has the following structure:
   a) one decision variable (i.e., column) is generated for each variable encountered which does not have a definition;
   b) the objective function usually includes the whole of the model logic, as specified by the variable definitions;
   c) one row is generated for each defined constraint in each time period (except period zero) up to the model horizon;
   d) an extra row and column is generated for each simultaneous relationship encountered.

5) Both the <ARITHMETIC EXPRESSION> and the relevant variable definitions must reduce to linear form. Any non-linearities will terminate the formulation process with an error message.
indicating where the non-linearity was encountered.

6) Whenever an attempt is made to generate a linear programme, the results of the previous linear programme (if any) are discarded.

7) The work matrix is not automatically reinitialised when the linear programme is generated.

8) The use of the TRACE modifier with this statement causes a row by row display of the linear programme.

9) The model statements must be reducible to linear form (for example a decision variable should not be multiplied or divided by another decision variable). If any non-linearities are detected, an error is displayed and the formulation process terminates.

Examples:

MAXIMISE NPV(DIVIDENDS/SHARECAPITAL, DISCOUNT, HORIZON)
MAXIMISE OBJECTIVE(5)
MAXIMISE 100 + 3 * A(1) - 2 * B(1) / C(1)
12.9 MINIMISE STATEMENT

The MINIMISE statement has the same effect as the MAXIMISE statement, except that the resulting linear programme is minimised. For further details refer to the MAXIMISE statement.

Syntax:

<MINIMISE STATEMENT> : ------ MINIMISE ------<ARITHMETIC EXPRESSION>------
12.10 NAME STATEMENT

The NAME statement associates a name with the model. This name is used in report headings, and also as documentation.

Syntax:

\(<\text{NAME STATEMENT}\> : \text{------ NAME ------<ASSIGNMENT OPERATOR>------<STRING>------}\)

Notes:

1) The model name may be specified or redefined throughout the modelling process.
2) An interrogation on the NAME statement will display the current name of the model.

Examples:

\begin{verbatim}
NAME IS "XYZ PLANNING MODEL"
NAME "INVESTMENT MODEL"
NAME IS "EXPANSION STRATEGIES"
\end{verbatim}

Enter: NAME?
Response: EXPANSION STRATEGIES
12.11 RISK STATEMENT

The risk statement provides the capability for running the model under all possible combinations of futures so that probability distributions are built for selected items.

Syntax:

```
<RISK STATEMENT> : --- RISK ------------<ARITHMETIC EXPRESSION>------
|     | <LOGICAL EXPRESSION>--->
|     | <CONSTRAINT PRIMARY>-----
```

Notes:

1) The distribution is constructed as follows:
   a) The "first" combination of futures is selected from all those scenarios which have not been ignored.
   b) The probabilities are checked in each future to ensure they sum to 1.
   c) The model is solved for the items specified (i.e., the list of items in the RISK Statement). The results are accumulated to calculate the statistics of the distribution.
   d) The next combination of futures is selected, and the process is repeated from step (b).
   e) If all combinations have been analysed, the scenarios are restored to their starting values, and the results are displayed.

2) For real valued items the expected value, range, variance and standard deviation are displayed. For logical items the probability of the conditional succeeding and failing is displayed.

3) Because the model must be solved many times to build the distributions, the risk analysis can be time consuming and expensive.

4) Only specific variable primaries and specific constraint primaries can be used in the risk
analysis expressions.

Examples:

RISK CASH(1), CASH(2), CASH(3), CASH(3) < 0;
RISK COSTS(5) > 85% * REVENUE(5),
    ASSETCONSTRAINT(3),
    EBIT(1), EBIT(5);
12.12 SAVE STATEMENT

The SAVE statement causes all variable and constraint definitions to be saved or displayed. SAVE is a system-supplied synonym for VIEW. Refer to the VIEW statement for a full description.

Syntax:

```
<SAVE STATEMENT> : ----- SAVE -----------------------------------------------
                  | ----- AS --<FILENAME>------
```
12.13 SELECT STATEMENT

The SELECT statement is used for selecting a set of options and futures to be used in the model. Additionally, it reverses the effect of the IGNORE statement when used with scenarios and decisions.

Syntax

<SELECT STATEMENT> : ______ SELECT _____________________________
     | ______<OPTION>----->
     | ______<FUTURE>----->
     | ______<SCENARIO>----->
     | ______<DECISION>-----

Notes:

1) When no items are specified in the SELECT statement, a display of all scenarios, decisions, futures and options, and their present state, is generated.

2) Whenever a set of items is selected, the variables are reinitialised from period one.

3) An option or future will remain selected until another option or future is selected from the same decision or scenario.
Examples:

Enter: SELECT LOWINFLATION, EXPINTEREST; RESPONSE: LOWINFLATION SELECTED IN INFLATIONRATES EXPINTEREST SELECTED IN INTERESTRATES REPLACES LOWINTEREST — NOTE THAT INTERESTRATES IS IGNORED VARIABLES REINITIALISED FROM PERIOD 1

Enter: SELECT; Response: SCENARIO INTERESTRATES (IGNORED) FUTURE LOWINTEREST FUTURE EXPINTEREST (SELECTED) FUTURE HIGHINTEREST ENDSCE NARIO SCENARIO INFLATIONRATES FUTURE LOWINFLATION (SELECTED) FUTURE EXPINFLATION ENDSCE NARIO DECISION EXPANSION OPTION BIGPLANT OPTION SMALLPLANT OPTION MEDIUMPLANT ENDDECISION

Enter: SELECT INTEREST RATES; Response: INTERESTRATES IS SELECTED SELECTED BRANCH IS EXPINTEREST VARIABLES REINITIALISED FROM PERIOD 1
12.14 USE STATEMENT

The USE statement causes input to the model to be read from an existing disc file. The entire disc file specified is read and processed by the model. At the end of the file, input reverts to the original device (i.e., terminal, card reader, etc).

Syntax:

<USE STATEMENT> : ----- USE --------<FILENAME>---------

Notes:

1) USE statements can be nested up to four deep (that is, a USED file can contain further USE statements).

2) If the PRINT modifier is used in conjunction with a USE statement, all input is displayed on the screen as it is read. Note that output to the print file is suppressed.

3) The remainder of the input record on which the USE statement occurs is treated as comment.

4) The format of the disc file is the same as that for standard input.

Examples:

USE SAVEDMODEL
USE REPORT/FILE
USE "STRATAGEM/DATA."
12.15 **VIEW STATEMENT**

The **VIEW** statement causes all the variable and constraint definitions to be displayed or printed.

**Syntax:**

```
<VIEW STATEMENT> : ----- VIEW ---------------------------------------
                  | ---- AS ---<FILENAME>-----
```

**Notes:**

1) **SAVE** is a system-supplied synonym for **VIEW**.
2) There are three possible media on which the definitions can be viewed:
   a) the user's screen.
      Example: VIEW
   b) the output print file (if the **PRINT** modifier is used).
      Example: PRINT VIEW
   c) a disc file if the **AS** **<FILENAME>** part is used. This also causes a copy to be printed on
      the output print file for later reference.
      Example: SAVE AS SAVEDMODEL
3) The file created by the **AS** **<FILENAME>** part is in a suitable format to be read by subsequent
   model runs using the **USE** statement.
4) The **<FILENAME>** must not be an already existing file.
## 13 RESERVED WORDS

<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARE</td>
<td>49</td>
</tr>
<tr>
<td>BY</td>
<td>100</td>
</tr>
<tr>
<td>BYE</td>
<td>110</td>
</tr>
<tr>
<td>CALC</td>
<td>12</td>
</tr>
<tr>
<td>CHANGES</td>
<td>111</td>
</tr>
<tr>
<td>COLUMNS</td>
<td>74</td>
</tr>
<tr>
<td>CONSTRAINT</td>
<td>32</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>74</td>
</tr>
<tr>
<td>DECISION</td>
<td>85</td>
</tr>
<tr>
<td>DECREASE</td>
<td>100</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>74</td>
</tr>
<tr>
<td>DISCOUNT</td>
<td>60</td>
</tr>
<tr>
<td>DOCUMENT</td>
<td>112</td>
</tr>
<tr>
<td>ELSE</td>
<td>57</td>
</tr>
<tr>
<td>ENDECISION</td>
<td>85</td>
</tr>
<tr>
<td>ENDFUTURE</td>
<td>94</td>
</tr>
<tr>
<td>ENDOPTION</td>
<td>87</td>
</tr>
<tr>
<td>ENDOREPORT</td>
<td>81</td>
</tr>
<tr>
<td>ENDSCEENARIO</td>
<td>92</td>
</tr>
<tr>
<td>FIELD</td>
<td>77</td>
</tr>
<tr>
<td>FIELD</td>
<td>74</td>
</tr>
<tr>
<td>FUTURE</td>
<td>94</td>
</tr>
<tr>
<td>HEADING</td>
<td>77</td>
</tr>
<tr>
<td>HORIZON</td>
<td>24, 113</td>
</tr>
<tr>
<td>IF</td>
<td>57</td>
</tr>
<tr>
<td>IGNORE</td>
<td>115</td>
</tr>
<tr>
<td>IN</td>
<td>90, 105</td>
</tr>
<tr>
<td>INCREASE</td>
<td>100</td>
</tr>
<tr>
<td>INDEX</td>
<td>28</td>
</tr>
<tr>
<td>INITIALISE</td>
<td>116</td>
</tr>
<tr>
<td>IS</td>
<td>49</td>
</tr>
<tr>
<td>LINE</td>
<td>77</td>
</tr>
<tr>
<td>MAXIMISE</td>
<td>117</td>
</tr>
<tr>
<td>MINIMISE</td>
<td>119</td>
</tr>
<tr>
<td>NAME</td>
<td>120</td>
</tr>
<tr>
<td>NPV</td>
<td>61</td>
</tr>
</tbody>
</table>
### Reserved Words

<table>
<thead>
<tr>
<th>Word</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>OF</td>
<td>30, 99</td>
</tr>
<tr>
<td>OPTION</td>
<td>87</td>
</tr>
<tr>
<td>PAGEWIDTH</td>
<td>74</td>
</tr>
<tr>
<td>PRINT</td>
<td>14</td>
</tr>
<tr>
<td>PROBABILITY</td>
<td>94</td>
</tr>
<tr>
<td>REPORT</td>
<td>72</td>
</tr>
<tr>
<td>RISK</td>
<td>121</td>
</tr>
<tr>
<td>SAVE</td>
<td>123</td>
</tr>
<tr>
<td>SCALE</td>
<td>74</td>
</tr>
<tr>
<td>SCENARIO</td>
<td>92</td>
</tr>
<tr>
<td>SELECT</td>
<td>124</td>
</tr>
<tr>
<td>SLDEPN</td>
<td>62</td>
</tr>
<tr>
<td>SLDEPNO</td>
<td>63</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>33</td>
</tr>
<tr>
<td>SUM</td>
<td>64</td>
</tr>
<tr>
<td>SUMCOL</td>
<td>77</td>
</tr>
<tr>
<td>T</td>
<td>23</td>
</tr>
<tr>
<td>TAB</td>
<td>77</td>
</tr>
<tr>
<td>THEN</td>
<td>57</td>
</tr>
<tr>
<td>TIME</td>
<td>23</td>
</tr>
<tr>
<td>TRACE</td>
<td>15</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>49, 52, 104, 105</td>
</tr>
<tr>
<td>UNDERLINE</td>
<td>77</td>
</tr>
<tr>
<td>USE</td>
<td>126</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>30</td>
</tr>
<tr>
<td>VIEW</td>
<td>127</td>
</tr>
<tr>
<td>WIDTH</td>
<td>74</td>
</tr>
</tbody>
</table>

The following words are reserved for future expansion:

- AS
- CLEAR
- CODE
- COMPARE
- DBDEPN
- IRR
- PLOT
- WITH
### INDEX

<p>| A Small Example                        | 4 |
| Adding a Decision Option              | 4 |
| Adding a Scenario Future              | 90 |
| ARE                                    | 97 |
| ARITHMETIC EXPRESSION                 | 51 |
| CONDITIONAL                           | 51 |
| ARITHMETIC OPERATOR                   | 49 |
| ARITHMETIC PRIMARY                    | 53 |
| ASSIGNMENT OPERATOR                   | 53 |
| Augmented Definition                  | 42 |
| Respecification                       | 42 |
| AUGMENTED DEFINITION RESPECIFICATION  | 42 |
| BLOCK NAME                            | 100 |
| BY                                      | 110 |
| BYE                                     | 30 |
| BYE STATEMENT                          | 100 |
| CALCULATE                              | 110 |
| CHANGES                                | 12 |
| CHANGES STATEMENT                      | 111 |
| Changing a Definition                  | 47 |
| Changing an Augmented Definition       | 103 |
| COLUMN                                 | 78 |
| COLUMN CONSTRAINT DEFINITION           | 51 |
| COLUMN CONSTRAINT PRIMARY              | 42 |
| COLUMN VARIABLE DEFINITION             | 48 |
| COLUMN VARIABLE PRIMARY               | 39 |
| COLUMNS                                | 74 |
| Comments                               | 11 |
| Components                             | 2 |
| Concepts                               | 1 |
| CONDITIONAL EXPRESSION                 | 57 |
| CONSTANT                               | 16 |
| CONSTRAINT                             | 32 |
| CONSTRAINT DECLARATION                 | 45 |
| CONSTRAINT DEFINITION                  | 32 |
| COLUMN                                 | 51 |
| DECIMAL                                | 74 |
| DECISION                               | 85 |
| DECISION and Option Examples           | 85 |
| DECISION BODY                          | 85 |
| DECISION DEFINITION                    | 85 |
| DECISION HEAD                          | 85 |
| DECISION NAME                          | 85 |
| DECISION OPTION                        | 87 |
| DECISION OPTION HEAD                   | 90 |
| DECISION TAIL                          | 85 |
| DECISIONS                              | 85 |
| DECLARATIONS                           | 27 |
| DECREASE                               | 100 |
| Definition                             | 100 |
| AUGMENTED                               | 47 |
| DECISION                               | 47 |
| Entering                               | 47 |
| FUTURE                                 | 97 |
| GENERAL IMPACT                         | 101 |
| IMPACT                                 | 100 |
| NULL                                   | 49 |
| OPTION                                 | 90 |
| Precedence                             | 45 |
| PROBABILITY                            | 94 |
| REPLACEMENT                            | 100 |
| REPORT | 71 | FUNCTION PARAMETER | 59 |
| SCENARIO | 92 | FUNCTION PARAMETER LIST | 59 |
| SPECIFIC IMPACT | 100 | FUNCTION PRIMARY | 59 |
| Definition Interrogation | 38 | FUTURE | 94 |
| DEFINITIONS | 37 | Example | 96 |
| Depreciation | 62-63 | FUTURE BODY | 94 |
| DESCRIPTION | 28 | FUTURE DEFINITION | 97 |
| 74 | FUTURE HEAD | 94 |
| DETAIL LINE | 78 | FUTURE NAME | 94 |
| DISCOUNT | 60 | FUTURE TAIL | 94 |
| DISCOUNT FUNCTION | 60 | FUTURES | 94 |
| DOCUMENT | 112 | GENERAL CONSTRAINT DEFINITION | 51 |
| DOCUMENT STATEMENT | 112 | GENERAL CONSTRAINT PRIMARY | 43 |
| ELSE | 57 | GENERAL IMPACT DEFINITION | 101 |
| EMPTY LINE | 77 | GENERAL VARIABLE DEFINITION | 48 |
| ENDECISION | 85 | GENERAL VARIABLE PRIMARY | 40 |
| ENDFUTURE | 94 | HEADING | 77 |
| ENDOPTION | 87 | HEADING LINE | 77 |
| ENREPORT | 81 | HEADING TEXT | 78 |
| ENSCENARIO | 92 | HORIZON | 113 |
| Entering a Definition | 47 | HORIZON STATEMENT | 113 |
| Example | 96 | HORIZONWORD | 24 |
| FUTURE | 96 | IDENTIFIERS | 18 |
| SCENARIO | 96 | IF | 57 |
| EXPRESSION | 53 | IF CLAUSE | 57 |
| ARITHMETIC | 57 | IGNORE | 115 |
| ARITHMETIC CONDITIONAL | 57 | IGNORE STATEMENT | 115 |
| CONDITIONAL LOGICAL | 55 | IMPACT DEFINITION | 100 |
| LOGICAL | 55 | IMPACT DEFINITION RESPECIFICATION | 105 |
| FIELD | 77 | IN | 100 |
| FIELD LINE | 77 | INCREASE | 100 |
| FIELDS | 74 | INDENT | 78 |
| FILENAME | 21 | INDEX | 28 |
| FORMAT | 78 | INDEX DECLARATION | 28 |
| FUNCTION | 60 | INDEX PART | 30 |
| DISCOUNT | 61 | INDEX SET | 28 |
| NPV | 62 | INDEXED QUALIFICATION | 35 |
| SLDEPN | 62 | INITIALISE | 116 |
| SLEDNPO | 63 | INITIALISE STATEMENT | 116 |
| SUM | 64 | | |</p>
<table>
<thead>
<tr>
<th>INTERROGATION</th>
<th>REPORT</th>
<th>STATEMENT</th>
<th>INTERROGATION LIST</th>
<th>IS</th>
<th>LINE</th>
<th>LINE DESCRIPTION</th>
<th>LOGICAL EXPRESSION</th>
<th>CONDITIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>82</td>
<td>108</td>
<td>66</td>
<td>49</td>
<td>77</td>
<td>78</td>
<td>55</td>
<td>57</td>
</tr>
<tr>
<td>MAXIMISE</td>
<td>MAXIMISE STATEMENT</td>
<td>MINIMISE STATEMENT</td>
<td>MODEL ITEM</td>
<td>Modifiers</td>
<td>NAME</td>
<td>NAME STATEMENT</td>
<td>NON-SPECIFIC AUGMENTED DEFINITION</td>
<td>RESPECIFICATION</td>
</tr>
<tr>
<td>117</td>
<td>117</td>
<td>119</td>
<td>13</td>
<td>13</td>
<td>120</td>
<td>120</td>
<td>104</td>
<td>61</td>
</tr>
<tr>
<td>QUALIFICATION</td>
<td>INDEXED</td>
<td>RELATIVE</td>
<td>SPECIFIC</td>
<td>RELATIONAL OPERATOR</td>
<td>RELATIVE QUALIFICATION</td>
<td>REPLACEMENT CONSTRAINT DEFINITION</td>
<td>RESPECIFICATION</td>
<td>REPLACEMENT DEFINITION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>99</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>105</td>
<td>100</td>
</tr>
<tr>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>71</td>
<td>83</td>
<td>72</td>
<td>72</td>
<td>82</td>
<td>69</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>74-75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>123</td>
<td>74</td>
<td>92</td>
<td>92</td>
<td>96</td>
<td>92</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>74</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>PAGewidth</td>
<td>Present Value</td>
<td>PRINT</td>
<td>PRINT MODIFIER</td>
<td>PROBABILITY</td>
<td>45</td>
<td>101</td>
<td>61</td>
<td>14</td>
</tr>
</tbody>
</table>
Appendix C:
A Stratagem Example.

In this appendix an annotated listing is given of the Stratagem implementation of the real estate model discussed in chapter nine. In the listing, all user input is preceded by a prompt character which can take one of two forms:

"=>" for input entered directly through the screen (or, as in this case, the card reader);

"O>" for input read from an existing disc file by means of the "USE" statement (see section B12.4).

Any lines in the listing which do not start with one of these prompts has been generated by Stratagem. A complete description of Stratagem is contained in the Stratagem User Manual (Appendix B).
Start of Stratagem. System initialises variables.

Read input from disc file "REALSTATE" and display each line.

Specify model name.

Declare variables used in model. For convenience the variables have been arranged by "function". The quoted string after each variable is a description that is used when displaying the variable in a report.
Declare the model parameters. This is a specialized use of the type "variable". The system itself does not distinguish amongst types of variables; such differentiation is defined by the user.

Declare the model constraints. These are the same as variables except that they may only take on a value of TRUE or FALSE.

Specify initial data. For this application the current balance sheet data is used.
Specify the model logic by means of Fortran-like arithmetic expressions. The subscript "(T-1)" refers to the value of the item in the previous period.
Specify the objective function.

The value of the objective function in period zero is meaningless, so it is set to zero.

Define the parameter values.

Specify the constraint definitions. These take the form of a simple logical expression.

Define likely futures for inflation and interest. In this case three possibilities have been identified. Subjective probabilities are assigned to each of these.
Strategies indicates that the scenario has been incorporated in the model.

Define likely values for the property-hedge factor. In this particular case, the probability of any particular hedge factor occurring is conditional upon the rate of inflation.

Define likely earning rates. These are a surrogate measure for the prevailing economic climate.

Define the possible alternatives for the bond structure. As these are controlled directly by the organisation they are specified as a decision instead of a scenario.

Define the possible alternatives for the application of managerial goals. Note that the option "GOALS" adds three new constraints to the model.
Specify the format for the Balance Sheet report.

Strategies indicates that the report is available for use.

Specify the layout of a special report to print the values of key property variables.

Specify a report to display the constraint values.
Set the model horizon to period five. This means that, with the inclusion of period zero, there are six periods.

Select a combination of futures and decision options for analysis. Stratagem ensures model consistency by reinitialising the variables.

Instruct the system to maximise the objective function in period five. Stratagem responds by displaying the objective function in terms of the actual decision variables.

<table>
<thead>
<tr>
<th>Objective Function Is</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DVW1</td>
<td>0.4093</td>
</tr>
<tr>
<td>DVW2</td>
<td>0.2415</td>
</tr>
<tr>
<td>DVW3</td>
<td>0.1435</td>
</tr>
<tr>
<td>DVW4</td>
<td>0.0692</td>
</tr>
<tr>
<td>DVW5</td>
<td>-0.1301</td>
</tr>
<tr>
<td>DVW6</td>
<td>-0.5690</td>
</tr>
<tr>
<td>H_UCAP(1)</td>
<td>-0.4908</td>
</tr>
<tr>
<td>H_UCAP(2)</td>
<td>-0.3049</td>
</tr>
<tr>
<td>H_UCAP(3)</td>
<td>-0.2999</td>
</tr>
<tr>
<td>H_UCAP(4)</td>
<td>-0.1756</td>
</tr>
<tr>
<td>H_UCAP(5)</td>
<td>-0.7560</td>
</tr>
<tr>
<td>IPUP(1)</td>
<td>0.3650</td>
</tr>
<tr>
<td>IPUP(2)</td>
<td>0.3650</td>
</tr>
<tr>
<td>IPUP(3)</td>
<td>0.3659</td>
</tr>
<tr>
<td>IPUP(4)</td>
<td>0.3659</td>
</tr>
<tr>
<td>IPUP(5)</td>
<td>0.1241</td>
</tr>
<tr>
<td>IPUP(6)</td>
<td>0.1414</td>
</tr>
<tr>
<td>IPUP(7)</td>
<td>0.1341</td>
</tr>
<tr>
<td>IPUP(8)</td>
<td>0.1893</td>
</tr>
<tr>
<td>IPUP(9)</td>
<td>0.2927</td>
</tr>
<tr>
<td>IPUP(10)</td>
<td>0.2927</td>
</tr>
<tr>
<td>IPUP(11)</td>
<td>0.7627</td>
</tr>
<tr>
<td>IPUP(12)</td>
<td>0.0927</td>
</tr>
<tr>
<td>IPUP(13)</td>
<td>0.1873</td>
</tr>
<tr>
<td>IPUP(14)</td>
<td>0.0623</td>
</tr>
<tr>
<td>IPUP(15)</td>
<td>0.2195</td>
</tr>
<tr>
<td>IPUP(16)</td>
<td>0.2195</td>
</tr>
<tr>
<td>SPUP(1)</td>
<td>0.2195</td>
</tr>
<tr>
<td>SPUP(2)</td>
<td>0.2195</td>
</tr>
<tr>
<td>SPUP(3)</td>
<td>0.0708</td>
</tr>
<tr>
<td>SPUP(4)</td>
<td>0.0905</td>
</tr>
<tr>
<td>SPUP(5)</td>
<td>0.0726</td>
</tr>
<tr>
<td>SPUP(6)</td>
<td>0.1463</td>
</tr>
<tr>
<td>SPUP(7)</td>
<td>0.1463</td>
</tr>
<tr>
<td>SPUP(8)</td>
<td>0.0740</td>
</tr>
<tr>
<td>SPUP(9)</td>
<td>0.0837</td>
</tr>
<tr>
<td>SPUP(10)</td>
<td>0.0637</td>
</tr>
<tr>
<td>SPUP(11)</td>
<td>0.0610</td>
</tr>
<tr>
<td>SPUP(12)</td>
<td>0.0732</td>
</tr>
<tr>
<td>SPUP(13)</td>
<td>0.0732</td>
</tr>
</tbody>
</table>
## Balance Sheet

### Period 0

<table>
<thead>
<tr>
<th>Assets</th>
<th>Period 0</th>
<th>Period 1</th>
<th>Period 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Trading Assets</td>
<td>770,000</td>
<td>900,900</td>
<td>1,054,053</td>
</tr>
<tr>
<td>Real Estate</td>
<td>230,000</td>
<td>450,450</td>
<td>527,077</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>0,000</td>
<td>0,000</td>
<td>0,000</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>1,000,000</td>
<td>1,351,350</td>
<td>1,581,079</td>
</tr>
</tbody>
</table>

### Represented By:

#### Shareholders Funds

<table>
<thead>
<tr>
<th></th>
<th>166,000</th>
<th>338,221</th>
<th>375,225</th>
</tr>
</thead>
<tbody>
<tr>
<td>Share Capital</td>
<td>371,000</td>
<td>467,920</td>
<td>580,465</td>
</tr>
<tr>
<td>Cumulative Retained Earnings</td>
<td>537,000</td>
<td>806,141</td>
<td>956,090</td>
</tr>
<tr>
<td>Current Liabilities:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provision for Tax</td>
<td>9,000</td>
<td>27,911</td>
<td>32,539</td>
</tr>
<tr>
<td>Provision for Dividend</td>
<td>16,000</td>
<td>11,286</td>
<td>13,305</td>
</tr>
<tr>
<td>Overdraft</td>
<td>69,000</td>
<td>44,694</td>
<td>44,944</td>
</tr>
<tr>
<td>Survey Creditors</td>
<td>178,000</td>
<td>208,260</td>
<td>243,664</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>272,000</td>
<td>291,051</td>
<td>334,042</td>
</tr>
</tbody>
</table>

### Period 3

<table>
<thead>
<tr>
<th>Assets</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Trading Assets</td>
<td>1,233,242</td>
<td>1,442,893</td>
<td>1,689,185</td>
</tr>
<tr>
<td>Real Estate</td>
<td>616,621</td>
<td>1,427,447</td>
<td>846,092</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>0,000</td>
<td>0,000</td>
<td>0,000</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>1,849,863</td>
<td>2,156,340</td>
<td>2,532,277</td>
</tr>
</tbody>
</table>

### Represented By:

#### Shareholders Funds
When the complete linear programme has been generated, Stratagem initiates Burroughs TEMPO to solve the problem. The user is instructed to wait until the solution is available, at which point it is loaded into Stratagem, the solution is displayed, and control returns to the user. The user instructs Stratagem to display the value of some critical variables.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>0</td>
<td>0.60</td>
<td>UNDEFINED</td>
<td>UNDEFINED</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>47.59</td>
<td>0.00</td>
<td>126.13</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>44.46</td>
<td>0.00</td>
<td>147.57</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>41.20</td>
<td>0.00</td>
<td>172.65</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>38.62</td>
<td>0.00</td>
<td>202.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>236.35</td>
<td>77.19</td>
</tr>
</tbody>
</table>

When the complete linear programme has been generated, Stratagem initiates Burroughs TEMPO to solve the problem. The user is instructed to wait until the solution is available, at which point it is loaded into Stratagem, the solution is displayed, and control returns to the user. The user instructs Stratagem to display the value of some critical variables.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>0</td>
<td>0.60</td>
<td>UNDEFINED</td>
<td>UNDEFINED</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>44.46</td>
<td>0.00</td>
<td>24.51</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>41.20</td>
<td>0.00</td>
<td>78.83</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38.62</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>126.13</td>
<td>77.19</td>
</tr>
</tbody>
</table>

When the complete linear programme has been generated, Stratagem initiates Burroughs TEMPO to solve the problem. The user is instructed to wait until the solution is available, at which point it is loaded into Stratagem, the solution is displayed, and control returns to the user. The user instructs Stratagem to display the value of some critical variables.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>0</td>
<td>0.60</td>
<td>UNDEFINED</td>
<td>UNDEFINED</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>47.59</td>
<td>0.00</td>
<td>126.13</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>44.46</td>
<td>0.00</td>
<td>147.57</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>41.20</td>
<td>0.00</td>
<td>172.65</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>38.62</td>
<td>0.00</td>
<td>202.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>236.35</td>
<td>77.19</td>
</tr>
</tbody>
</table>

Display the balance sheet report.
<table>
<thead>
<tr>
<th>現代課題</th>
<th>状態方程式</th>
<th>時系列データ</th>
<th>結果分析</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHADE CAPITAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CURRENT LIABILITIES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL LIABILITIES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERDRAFT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH IN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH OUT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Do risk analysis for each of the variables and constraints specified in list.

Stratagem displays the results of the risk analysis. It first gives the possible and actual number of combinations of futures which can be generated by the specified scenarios. This is followed by a summary of the distribution generated for each item in the list.

Results for OBJ(1):  
EXPECTED VALUE IS 474.3929  
VARIANCE IS 196.0191  
STANDARD DEVIATION IS 44.8190  
RANGE IS FROM -317.6291 TO 337.9223

Results for OBJ(2):  
EXPECTED VALUE IS 462.3909  
VARIANCE IS 206.0212  
STANDARD DEVIATION IS 45.3901  
RANGE IS FROM -312.7159 TO 334.3901

Results for OBJ(3):  
EXPECTED VALUE IS 410.0206  
VARIANCE IS 37.113061  
STANDARD DEVIATION IS 61.2701  
RANGE IS FROM 246.7405 TO 543.9198

Results for OBJ(4):  
EXPECTED VALUE IS 378.0801  
VARIANCE IS 546.0900  
STANDARD DEVIATION IS 73.8036  
RANGE IS FROM 111.2242 TO 533.6316

Results for OBJ(5):  
EXPECTED VALUE IS 347.0806
VARIANCE IS 699.1,656
STANDARD DEVIATION IS 83.6283
RANGE IS FROM 131,8603 TO 516,7210

RESULTS FOR NPAT(1):
- EXPECTED VALUE IS 95.5129
  VARIANCE IS 141.6129
  STANDARD DEVIATION IS 11.9041
  RANGE IS FROM 44,0124 TO 107,2490

RESULTS FOR NPAT(2):
- EXPECTED VALUE IS 78.2447
  VARIANCE IS 652.4096
  STANDARD DEVIATION IS 24.941
  RANGE IS FROM 49,0371 TO 135.9904

RESULTS FOR NPAT(3):
- EXPECTED VALUE IS 43.3270
  VARIANCE IS 161.7294
  STANDARD DEVIATION IS 12.6981
  RANGE IS FROM 51.7167 TO 173,0119

RESULTS FOR NPAT(4):
- EXPECTED VALUE IS 111.8144
  VARIANCE IS 141.6041
  STANDARD DEVIATION IS 40,2368
  RANGE IS FROM 53,2397 TO 220,647

RESULTS FOR NPAT(5):
- EXPECTED VALUE IS 134.5308
  VARIANCE IS 261.3729
  STANDARD DEVIATION IS 52,5408
  RANGE IS FROM 95,0164 TO 281,8735

RESULTS FOR CASH(1):
- EXPECTED VALUE IS 13.0830
  VARIANCE IS 166,1532
  STANDARD DEVIATION IS 40,7864
  RANGE IS FROM 54.5899 TO 126.5696

RESULTS FOR CASH(2):
- EXPECTED VALUE IS 16.2960
  VARIANCE IS 653.7346
  STANDARD DEVIATION IS 80.8537
  RANGE IS FROM -127.5128 TO 236,8410

RESULTS FOR CASH(3):
- EXPECTED VALUE IS 17,2615
  VARIANCE IS 1,860.8769
  STANDARD DEVIATION IS 130.1746
  RANGE IS FROM -227.5461 TO 360.8243

RESULTS FOR CASH(4):
- EXPECTED VALUE IS 15.2730
The results for logical expressions and constraints are given as the probability of the success and failure of that expression. In this case there is a probability of 0.376 that the constraint XCASNO will be broken (i.e. there will be insufficient cash).
PROBABILITY OF CONDITION SUCCEEDING IS 0.7500  
VARIANCE IS 0.11075  
STANDARD DEVIATION IS 0.4330  

RESULTS FOR XINT(4):  
PROBABILITY OF CONDITION FAILING IS 0.2500  
PROBABILITY OF CONDITION SUCCEEDING IS 0.7500  
VARIANCE IS 0.11075  
STANDARD DEVIATION IS 0.4330  

RESULTS FOR XINT(5):  
PROBABILITY OF CONDITION FAILING IS 0.1640  
PROBABILITY OF CONDITION SUCCEEDING IS 0.8360  
VARIANCE IS 0.11711  
STANDARD DEVIATION IS 0.44303  

--- END OF PLOT ANALYSIS. ---

VARIABLES DELETED FROM PERIOD 1  
=SPROP, PROP, LPRO, DPRO  
VARIABLES DELETED FROM PERIOD 1  
=SPROP, PROP, LPRO, DPRO, LTRAD, NTRAD, DTRAD, NTRAD  

<table>
<thead>
<tr>
<th>TIME</th>
<th>OHL</th>
<th>CASH</th>
<th>EXIT</th>
<th>IHAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>UNDEFINED</td>
<td>UNDEFINED</td>
</tr>
<tr>
<td>1</td>
<td>496.39</td>
<td>46.71</td>
<td>116.62</td>
<td>48.75</td>
</tr>
<tr>
<td>2</td>
<td>467.95</td>
<td>49.02</td>
<td>126.08</td>
<td>51.98</td>
</tr>
<tr>
<td>3</td>
<td>436.60</td>
<td>180.58</td>
<td>154.11</td>
<td>55.93</td>
</tr>
<tr>
<td>4</td>
<td>463.13</td>
<td>209.36</td>
<td>196.42</td>
<td>59.66</td>
</tr>
<tr>
<td>5</td>
<td>375.90</td>
<td>190.68</td>
<td>162.18</td>
<td>63.41</td>
</tr>
</tbody>
</table>

Different futures are selected to test the linear programming solution under alternative environmental assumptions, in this case low inflation instead of expected inflation. The value of some critical variables are displayed.
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>232.00</td>
<td>241.08</td>
<td>247.90</td>
<td>246.48</td>
<td>242.05</td>
<td>216.39</td>
<td>20.00</td>
</tr>
</tbody>
</table>

---

### BALANCE SHEET

<table>
<thead>
<tr>
<th></th>
<th>PERIOD 1</th>
<th>PERIOD 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROSS TRADING ASSETS</td>
<td>770,000</td>
<td>839,300</td>
</tr>
<tr>
<td>REAL ESTATE</td>
<td>230,000</td>
<td>410,950</td>
</tr>
<tr>
<td>CASH IN HAND</td>
<td>0.00</td>
<td>46,710</td>
</tr>
<tr>
<td></td>
<td>1,000,000</td>
<td>1,264,860</td>
</tr>
</tbody>
</table>

---

### SHAREHOLDERS' EQUITY

<table>
<thead>
<tr>
<th></th>
<th>PERIOD 1</th>
<th>PERIOD 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARE CAPITAL</td>
<td>166,000</td>
<td>188,225</td>
</tr>
<tr>
<td>EQUITY</td>
<td>537,000</td>
<td>706,128</td>
</tr>
<tr>
<td>CURRENT Liabilities:</td>
<td>427,907</td>
<td>485,768</td>
</tr>
<tr>
<td></td>
<td>660,943</td>
<td></td>
</tr>
<tr>
<td>PROVISION FOR TAX</td>
<td>9,000</td>
<td>29,770</td>
</tr>
<tr>
<td>OVERDRAFT</td>
<td>16,000</td>
<td>13,305</td>
</tr>
<tr>
<td>PAYABLES</td>
<td>178,000</td>
<td>211,422</td>
</tr>
<tr>
<td></td>
<td>290,430</td>
<td></td>
</tr>
</tbody>
</table>

---

### BALANCE SHEET

<table>
<thead>
<tr>
<th></th>
<th>PERIOD 3</th>
<th>PERIOD 4</th>
<th>PERIOD 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROSS TRADING ASSETS</td>
<td>997,172</td>
<td>1,086,918</td>
<td>1,104,750</td>
</tr>
<tr>
<td>REAL ESTATE</td>
<td>463,678</td>
<td>489,953</td>
<td>515,498</td>
</tr>
</tbody>
</table>
CASH IN HAND

159,518
1629,358
1800,188
309,680
2009,919

REPRESENTED BY:

SHAREHOLDERS FUNDS

SHARE CAPITAL
CORPORATE CAPITAL EARNINGS
643,766
961,806
600,830
1069,471
225,131
555,847
1180,778

CURRENT LIABILITIES:

PROVISION FOR TAX
31,977
34,170
36,314

PROVISION FOR DIVIDEND
15,829
18,675

OVERTIME
12,124
12,132

SUNDAY CREDITORS
230,518
401,645
251,262
427,431
273,870
455,507

TOTAL LIABILITIES
131,000
111,000
194,296
310,286
91,000
282,634
373,634

1620,368
1806,188
2009,919

REAL ESTATE HOMEL

MODEL CONSTRAINTS

MAINTAIN ORIGINAL PUNG STRUCTURE
MANAGERIAL GOALS TO APPLY
LOW INFLATION (90)
HIGH INFLATION (100)
LOW RETURN ON TRADING ASSETS (90)

STRUCTURAL REQUIREMENTS

LOW RETURN ON TRADING ASSETS (90)

MANAGERIAL GOALS

INCREASING DIVIDEND REQUIREMENT
MAXIMUM DIVIDEND REQUIREMENT
NON-NEGATIVITY
CASH
REAL ESTATE NON-NEGATIVITY
OVERDRAFT NON-NEGATIVITY

=>1<! LOOK AT POST HORIZON EFFECTS
=>SELECT EXCEPT:
EXPENSES SELECTED IN INFLATION RATES
REPLACES LOW INF
VARIABLES DEINITIALIZED FROM PERIOD 1
=HORIZON IS 100:
11 PERIODS (= 10)
=>MAXIMIZE OBJECTIVE

A display of the "constraint report" shows that if inflation is low, the overdraft and dividend requirements may not always be met under the present policy.

To test for post-horizon effects the horizon is extended to period ten, and the linear programme rerun (under expected inflation). Rerunning the model ensures that constraints will not be broken after period five.
The results suggest that there will be no undesirable post-horizon effects for this particular policy. Note, however, that the present value of the organisation, as represented by the objective function (variable OBJ), is dropping over time, indicating that the desired rate of return (7%) is not being achieved.
<table>
<thead>
<tr>
<th>ASSETS</th>
<th>PERIOD 0</th>
<th>PERIOD 1</th>
<th>PERIOD 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROSS TRADING ASSETS</td>
<td>770,000</td>
<td>906,400</td>
<td>1,054,053</td>
</tr>
<tr>
<td>REAL ESTATE</td>
<td>230,000</td>
<td>450,450</td>
<td>577,977</td>
</tr>
<tr>
<td>CASH IN HAND</td>
<td>0,000</td>
<td>6,000</td>
<td>0,000</td>
</tr>
<tr>
<td></td>
<td>1,000,000</td>
<td>1,351,350</td>
<td>1,581,000</td>
</tr>
</tbody>
</table>

| PRESENTED BY:                  |          |          |          |
| SHARPEHOLDERS FUNDS            |          |          |          |
| SHARE CAPITAL                  | 166,000  | 338,221  | 375,225  |
| CUMULATIVE RETAINED EARNINGS   | 371,000  | 467,020  | 530,865  |
| CURRENT LIABILITIES            |          | 806,141  | 956,090  |
| PROVISION FOR TAX              | 9,000    | 27,811   | 33,629   |
| PROVISION FOR DIVIDENDS         | 16,000   | 11,986   | 13,305   |
| OVERDRAFT                      | 69,000   | 90,090   | 105,405  |
| SELF CREDITORS                 | 178,000  | 272,000  | 243,644  |
| ADVANCEビルフォリリABILITIES     |          | 337,447  | 394,993  |
| OLD PRESENTATION STOCK          | 191,000  | 171,000  | 151,000  |
| NEW PRESENTATION STOCK          | 0,000    | 191,000  | 78,996   |
|                                 | 1,000,000| 1,351,350| 1,581,079|

<table>
<thead>
<tr>
<th>BALANCE SHEET</th>
<th>PERIOD 3</th>
<th>PERIOD 4</th>
<th>PERIOD 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSETS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROSS TRADING ASSETS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REAL ESTATE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH IN HAND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRESENTED BY:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHARPEHOLDERS FUNDS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Period 6</td>
<td>Period 7</td>
<td>Period 8</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td><strong>ASSETS</strong></td>
<td><strong>ASSETS</strong></td>
<td><strong>ASSETS</strong></td>
<td></td>
</tr>
<tr>
<td>Gross Trading Assets</td>
<td>2092,987</td>
<td>2450,552</td>
<td>3190,543</td>
</tr>
<tr>
<td>Real Estate</td>
<td>1016,979</td>
<td>1226,276</td>
<td>0,000</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>3109,960</td>
<td>3675,027</td>
<td>4061,308</td>
</tr>
</tbody>
</table>

**PRESENTED BY:**

**SHAREHOLDERS' FUNDS**

<table>
<thead>
<tr>
<th>Period 9</th>
<th>Period 10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SHARE CAPITAL</strong></td>
<td><strong>SHARE CAPITAL</strong></td>
</tr>
<tr>
<td><strong>CUMULATIVE RETAINED EARNINGS</strong></td>
<td><strong>CUMULATIVE RETAINED EARNINGS</strong></td>
</tr>
<tr>
<td>Provision for Tax</td>
<td>60,212</td>
</tr>
<tr>
<td>Provision for Dividends</td>
<td>28,889</td>
</tr>
<tr>
<td>Share Reserve</td>
<td>50,54</td>
</tr>
<tr>
<td>Others</td>
<td>441,339</td>
</tr>
<tr>
<td><strong>TOTAL LIABILITIES</strong></td>
<td><strong>TOTAL LIABILITIES</strong></td>
</tr>
<tr>
<td>Old Preference Stock</td>
<td>71,000</td>
</tr>
<tr>
<td>New Preference Stock</td>
<td>254,371</td>
</tr>
<tr>
<td><strong>BALANCE SHEET</strong></td>
<td><strong>BALANCE SHEET</strong></td>
</tr>
<tr>
<td>3109,960</td>
<td>3675,027</td>
</tr>
</tbody>
</table>
Investigate the effect of changing the planning horizon by maximising the objective function in period ten.
The results show that there is no significant change in the decision variables, suggesting that the solution is not sensitive to changes in the planning horizon.
<table>
<thead>
<tr>
<th>YEAR</th>
<th>PROP</th>
<th>LPROP</th>
<th>HPROP</th>
<th>SPROP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>230.00</td>
<td>230.00</td>
<td>230.00</td>
<td>230.00</td>
</tr>
<tr>
<td>2</td>
<td>650.45</td>
<td>650.45</td>
<td>650.45</td>
<td>650.45</td>
</tr>
<tr>
<td>3</td>
<td>1301.76</td>
<td>1301.76</td>
<td>1301.76</td>
<td>1301.76</td>
</tr>
<tr>
<td>4</td>
<td>724.15</td>
<td>724.15</td>
<td>724.15</td>
<td>724.15</td>
</tr>
<tr>
<td>5</td>
<td>786.51</td>
<td>786.51</td>
<td>786.51</td>
<td>786.51</td>
</tr>
<tr>
<td>6</td>
<td>849.96</td>
<td>849.96</td>
<td>849.96</td>
<td>849.96</td>
</tr>
<tr>
<td>7</td>
<td>913.41</td>
<td>913.41</td>
<td>913.41</td>
<td>913.41</td>
</tr>
<tr>
<td>8</td>
<td>976.87</td>
<td>976.87</td>
<td>976.87</td>
<td>976.87</td>
</tr>
<tr>
<td>9</td>
<td>1040.43</td>
<td>1040.43</td>
<td>1040.43</td>
<td>1040.43</td>
</tr>
<tr>
<td>10</td>
<td>1104.00</td>
<td>1104.00</td>
<td>1104.00</td>
<td>1104.00</td>
</tr>
</tbody>
</table>

**Balance Sheet**

**Page 1**

**Maintain original bond structure**

**Managerial goals to apply**

**Expected inflation (15%)**

**High hedge (160%)**

**Low return on trading assets (9%)**

**Assets**

- Gross trading assets
- Real estate
- Cash in hand

**Balance Sheet**

**Page 2**
### Period 3

<table>
<thead>
<tr>
<th>Assets</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Trading Assets</td>
<td>1233.242</td>
<td>1442.893</td>
<td>1688.185</td>
</tr>
<tr>
<td>Real Estate</td>
<td>616.631</td>
<td>721.647</td>
<td>844.092</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>1849.863</td>
<td>2164.390</td>
<td>2532.277</td>
</tr>
</tbody>
</table>

**Represented by:**

<table>
<thead>
<tr>
<th>Shareholders Funds</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Share Capital</td>
<td>418.046</td>
<td>1130.636</td>
<td>525.131</td>
</tr>
<tr>
<td>Cumulative Retained Earnings</td>
<td>712.590</td>
<td>846.319</td>
<td>1045.825</td>
</tr>
</tbody>
</table>

**Current Liabilities:**

| Provision for Tax       | 38.070   | 44.642   | 52.114   |
| Provision for Dividend  | 35.629   | 16.475   | 21.933   |
| Owe Dep</li> | 91.107   | 104.299  | 130.120  |
| Trade Creditors         | 285.087  | 390.093  | 594.483  |

**Total Liabilities:**

| Old Preference Stock    | 131.000  | 111.000  | 91.000   |
| New Preference Stock    | 198.134  | 178.321  | 275.839  |
|                        | 319.134  | 289.321  | 366.839  |

| Balance Sheet           | 1849.863 | 2164.340 | 2532.277 |

### Period 6

<table>
<thead>
<tr>
<th>Assets</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Trading Assets</td>
<td>1975.176</td>
<td>2310.956</td>
<td>2703.819</td>
</tr>
<tr>
<td>Real Estate</td>
<td>987.580</td>
<td>1108.478</td>
<td>1351.910</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>-3.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>2462.765</td>
<td>3466.435</td>
<td>4055.729</td>
</tr>
</tbody>
</table>

**Represented by:**

<table>
<thead>
<tr>
<th>Shareholders Funds</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Share Capital</td>
<td>591.827</td>
<td>1647.346</td>
<td>1787.069</td>
</tr>
<tr>
<td>Cumulative Retained Earnings</td>
<td>1255.560</td>
<td>1500.567</td>
<td>2169.829</td>
</tr>
</tbody>
</table>

**Current Liabilities:**

| Provision for Tax       | 60.974   | 71.339   | 83.667   |
| Provision for Dividend  | 25.863   | 30.378   | 35.847   |

### Period 7

### Period 8
<table>
<thead>
<tr>
<th>OVERDEBT</th>
<th>130,120</th>
<th>130,120</th>
<th>130,120</th>
<th>130,120</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUDEY CREDITORS</td>
<td>456,539</td>
<td>534,721</td>
<td>766,057</td>
<td>625,039</td>
</tr>
<tr>
<td>TFP LIABILITIES:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLD DEBTURENT STOCK</td>
<td>71,000</td>
<td>51,000</td>
<td>31,000</td>
<td></td>
</tr>
<tr>
<td>NEW DEBTURENT STOCK</td>
<td>370,863</td>
<td>419,584</td>
<td>604,216</td>
<td>635,216</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BALANCE SHEET</td>
<td>2962,165</td>
<td>3666,435</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASSETS</td>
<td>PERIOD 9</td>
<td>PERIOD 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH IN HAND</td>
<td>4745,202</td>
<td>5551,887</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESTATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROSS TRADE ASSETS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3163,468</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1581,734</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4745,202</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHAREHOLDERS FUNDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHARE CAPITAL</td>
<td>863,832</td>
<td>985,523</td>
<td>3449,074</td>
<td></td>
</tr>
<tr>
<td>CUMULATIVE RETAINED EARNINGS</td>
<td>2121,914</td>
<td>7513,551</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREDIT LIABILITIES:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROVISION FOR TAX</td>
<td>97,656</td>
<td>114,258</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROVISION FOR DIVIDEND</td>
<td>41,000</td>
<td>61,087</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERDEBT</td>
<td>130,120</td>
<td>370,126</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUEY CREDITORS</td>
<td>711,255</td>
<td>855,615</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TFP LIABILITIES:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLD DEBTURENT STOCK</td>
<td>11,000</td>
<td>-9,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEW DEBTURENT STOCK</td>
<td>747,585</td>
<td>677,826</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>663,826</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5551,887</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test model sensitivity to changes in the required rate of return. Change its value from 7% to 1%.
The table below represents a decision variable analysis with 66 rows and 45 decision variables. The optimization status is optimal, and the objective function value is 512.28069111. The table includes columns for different time periods (T0, T1, T2, T3, T4, T5) with values for each variable.

Again, there is no significant difference in the value of the decision variables. Note that the present value of the organization (OBJ) is almost static over time, indicating that a rate of return of 1% is just being achieved.
### Balance Sheet

#### Period 0

<table>
<thead>
<tr>
<th>Assets</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Trading Assets</td>
<td>770,000</td>
</tr>
<tr>
<td>Real Estate</td>
<td>230,000</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>

#### Period 1

<table>
<thead>
<tr>
<th>Assets</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Trading Assets</td>
<td>900,900</td>
</tr>
<tr>
<td>Real Estate</td>
<td>450,450</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>1,351,350</td>
</tr>
</tbody>
</table>

#### Period 2

<table>
<thead>
<tr>
<th>Assets</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Trading Assets</td>
<td>1,054,053</td>
</tr>
<tr>
<td>Real Estate</td>
<td>527,077</td>
</tr>
<tr>
<td>Cash in Hand</td>
<td>1,581,080</td>
</tr>
</tbody>
</table>

#### Represented By:

<table>
<thead>
<tr>
<th>Shareholders Funds</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Share Capital</td>
<td>156,000</td>
</tr>
<tr>
<td>Cumulative Retained Earnings</td>
<td>371,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current Liabilities:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision for Tax</td>
<td>9,000</td>
</tr>
<tr>
<td>Provision for Dividend</td>
<td>16,000</td>
</tr>
</tbody>
</table>

#### Maintained Original Bond Structure

- MANAGEMENT GOALS TO APPLY
- EXPECTED INFLATION (17%)
- HIGH INCOME (100%)
- LOW RETURN ON TRADING ASSFS (99%)

#### Notes:

- 11:24 AM, 08 MAR 1982
- PAGE 1
## OVERDRAFT

**TOTAL LIABILITIES:**

<table>
<thead>
<tr>
<th></th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLD DEFERRED STOCK</td>
<td>178,000</td>
<td>171,000</td>
<td>151,000</td>
</tr>
<tr>
<td>NEW DEFERRED STOCK</td>
<td>69,000</td>
<td>76,467</td>
<td>98,312</td>
</tr>
<tr>
<td></td>
<td>248,000</td>
<td>247,467</td>
<td>249,312</td>
</tr>
</tbody>
</table>

## ASSETS

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH IN HAND</td>
<td>100,000</td>
<td>100,000</td>
<td>100,000</td>
</tr>
<tr>
<td>REAL ESTATE</td>
<td>1,462,493</td>
<td>721,447</td>
<td>844,092</td>
</tr>
<tr>
<td>GROSS TRADING ASSETS</td>
<td>1,564,493</td>
<td>821,447</td>
<td>944,092</td>
</tr>
<tr>
<td></td>
<td>2,026,493</td>
<td>942,894</td>
<td>1,088,185</td>
</tr>
</tbody>
</table>

## SHAREHOLDERS FUNDS

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARE CAPITAL</td>
<td>418,046</td>
<td>467,641</td>
<td>526,131</td>
</tr>
<tr>
<td>CURRENT LIABILITIES:</td>
<td>113,636</td>
<td>133,960</td>
<td>157,956</td>
</tr>
<tr>
<td>PROVISION FOR TAX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRIORITY FOR DIVIDEND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERDRAFT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLD DEFERRED STOCK</td>
<td>131,000</td>
<td>429,076</td>
<td>390,256</td>
</tr>
<tr>
<td>NEW DEFERRED STOCK</td>
<td>150,151</td>
<td>111,000</td>
<td>390,256</td>
</tr>
<tr>
<td></td>
<td>281,151</td>
<td>540,076</td>
<td>780,512</td>
</tr>
</tbody>
</table>

--- END OF STATEMENT ---

Terminate model session.
Bibliography.


Cowie, G.C., 1977, An Investigation into the Uses of Multiple Decision Trees as a Modelling Technique, Unpublished Research Essay, Department of Management Studies, University of Auckland.


Ross, M., (editor), 1973, OR'72, North Holland.


