The planning and management of detailed building design

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/7061

Publisher: © Andrew John Newton

Please cite the published version.
This item is held in Loughborough University’s Institutional Repository (https://dspace.lboro.ac.uk/) and was harvested from the British Library’s EThOS service (http://www.ethos.bl.uk/). It is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
THE PLANNING AND MANAGEMENT OF DETAILED BUILDING DESIGN

By

Andrew John Newton

A Doctoral Thesis submitted in partial fulfilment of the award of Doctor of Philosophy of the Loughborough University of Technology.

August 1995.

© by Andrew John Newton 1995
The Planning and Management of Detailed Building Design

Historically, building design has been manageable without the help of special planning and management techniques, whereas in construction there have been clearer, more easily realisable benefits. As buildings become technically more complex and design teams more specialised and fragmented, the need to plan and co-ordinate the design process with greater accuracy is becoming increasingly important. Traditionally building design work has been planned in a perfunctory manner, often in the belief that this creative and iterative process cannot be analysed and planned in detail. This situation has been perpetuated by a lack of understanding of design information flow and dependency and the availability of suitable planning techniques.

ADePT (Analytical Design Planning Technique) has been developed in this research and permits a more sophisticated approach to the planning of building design work to be taken. This prototype methodology uses Design Structure Matrix Analysis (DSMA) to examine a Design Process Model (DPM) of the building design process. The synthesis of these two techniques produces a powerful but easily understood tool to assist in the planning and management of complex, multi-disciplinary building design problems.

Traditional design programming is time consuming and reliant on a planner's experience, with each task and link often being defined afresh at the beginning of each new project. The Design Process Model, constructed from data flow diagrams, eliminates much of this subjectivity by generically representing the tasks involved, and the information flowing in the design of any building in a consistent, re-usable manner.

The unsuitability of traditional planning tools also contributes to the development of unrealistic design programmes; design is an inherently iterative activity and techniques such as network analysis, are unable to represent this type of relationship. ADePT overcomes these failings by using DSMA to analyse the DPM to reveal how to most efficiently order inter-dependent tasks based purely on the optimal flow of design information. ADePT can also incorporate the impact of external influences such as construction programme, materials procurement or resourcing demands to be superimposed on this idealised design programme, allowing their influence on the optimal design task order to be assessed.
ACKNOWLEDGEMENTS

This Thesis would have not been possible without the help and support of many kind people. If I were to attempt to list everyone I would undoubtedly forget and offend someone, so if you have contributed to my work, however indirectly you have my heartfelt thanks. Having said all that it would be wrong not to mention the people who have had a major influence on this work.

Special thanks must go to Simon Austin and Andrew Baldwin, my joint supervisors, for their opinions, encouragement and good humour. Without their expertise and cajoling this work would never have been completed. I would also like to acknowledge the help and support of AMEC Design and Management, particularly the people who work at the Stratford office, who have always been willing to spare me five minutes of their valuable time whenever it was requested. A specific mention here must go to Mike Murray, my industrial supervisor, for his foresight in getting this project off the ground and for his advice and guidance over the three years.

On a personal note I would like to thank three special people: my parents for having the belief in me to complete a PhD; and last finally to Mandy, firstly for the thankless task of proof-reading this text, but most importantly for her continuing love and support.

Thank you all.
# CONTENTS

## Chapter One:

**INTRODUCTION**

1.1 BACKGROUND TO RESEARCH ........................................ 1
1.2 AIMS AND OBJECTIVES ............................................ 3
1.3 RESEARCH METHODOLOGY ........................................ 4
1.4 RESULTS OF RESEARCH ........................................... 5
   1.4.1 Research Output ........................................ 5
   1.4.2 Main Findings ........................................... 6
1.5 GUIDE TO THESIS ................................................ 8

## Chapter Two:

**A REVIEW OF CURRENT BUILDING DESIGN PROCESS PROBLEMS**

2.1 INTRODUCTION .................................................. 11
2.2 THE DESIGN PROCESS ........................................... 11
   2.2.1 Definitions ................................................ 11
   2.2.2 The Nature of the Design Process ..................... 12
   2.2.3 The Building Design Process ......................... 14
2.3 PROBLEMS IN BUILDING DESIGN ................................ 15
   2.3.1 The Causes ................................................. 15
   2.3.2 Client Sophistication .................................... 16
   2.3.3 Fast-Tracking ............................................. 18
   2.3.4 Building Complexity .................................... 19
   2.3.5 Information Management ................................. 21
   2.3.6 Planning of Design Work ................................. 23
   2.3.7 Possible Solutions ....................................... 24
2.4 WHY IS DESIGN PLANNED POORLY? ............................. 24
   2.4.1 The Basic Design Planning Procedure ................ 24
   2.4.2 Bar Charts ................................................ 25
   2.4.3 Network Analysis ......................................... 27
   2.4.4 The Applicability of Network Analysis to Represent and Plan Design ..................................... 28
   2.4.5 Improvements to Network Analysis .................... 32
2.5 CONCLUSIONS ................................................... 33
# Contents

## Chapter Three:
**RESEARCH OBJECTIVES AND METHODOLOGY**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 INTRODUCTION</td>
<td>34</td>
</tr>
<tr>
<td>3.2 RESEARCH OBJECTIVES</td>
<td>34</td>
</tr>
<tr>
<td>3.2.1 The Aim of the Research</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Phase I: Understanding the Building Design Process</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3 Phase II: Producing a Design Planning Methodology</td>
<td>36</td>
</tr>
<tr>
<td>3.3 RESEARCH METHODOLOGY</td>
<td>36</td>
</tr>
</tbody>
</table>

## Chapter Four:
**EXISTING MODELS OF THE DESIGN PROCESS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 INTRODUCTION</td>
<td>39</td>
</tr>
<tr>
<td>4.2 MODELS AND MODELLING</td>
<td>39</td>
</tr>
<tr>
<td>4.2.1 General Definitions</td>
<td>39</td>
</tr>
<tr>
<td>4.2.2 Requirements of a Model to Represent the Building Design Process</td>
<td>40</td>
</tr>
<tr>
<td>4.3 DESIGN MODELS FROM THE MANUFACTURING DOMAIN</td>
<td>42</td>
</tr>
<tr>
<td>4.3.1 Descriptive Models</td>
<td>42</td>
</tr>
<tr>
<td>4.3.2 Prescriptive Models</td>
<td>43</td>
</tr>
<tr>
<td>4.3.3 Consensus Models</td>
<td>44</td>
</tr>
<tr>
<td>4.3.4 Applicability to Represent the Building Design Process</td>
<td>45</td>
</tr>
<tr>
<td>4.4 DESIGN MODELS FROM THE CONSTRUCTION DOMAIN</td>
<td>46</td>
</tr>
<tr>
<td>4.4.1 Building Product Models</td>
<td>47</td>
</tr>
<tr>
<td>4.4.2 Wix's Model for Representing the Design Process</td>
<td>48</td>
</tr>
<tr>
<td>4.4.3 The RIBA Plan of Work</td>
<td>50</td>
</tr>
<tr>
<td>4.5 CONCLUSIONS</td>
<td>52</td>
</tr>
</tbody>
</table>

## Chapter Five:
**TECHNIQUES FOR MODELLING THE DESIGN PROCESS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 INTRODUCTION</td>
<td>54</td>
</tr>
<tr>
<td>5.2 STRUCTURED METHODOLOGIES &amp; TECHNIQUES</td>
<td>54</td>
</tr>
<tr>
<td>5.2.1 Entity Relationship Diagrams</td>
<td>55</td>
</tr>
<tr>
<td>5.2.2 Data Flow Diagrams</td>
<td>56</td>
</tr>
<tr>
<td>5.2.3 Structure Charts</td>
<td>58</td>
</tr>
<tr>
<td>5.3 OTHER STRUCTURED MODELLING TECHNIQUES</td>
<td>60</td>
</tr>
<tr>
<td>5.3.1 Role Activity Diagrams</td>
<td>60</td>
</tr>
<tr>
<td>5.3.2 Structured Analysis and Design Technique</td>
<td>62</td>
</tr>
</tbody>
</table>
# Contents

5.4 CONCLUSIONS ON THE MOST SUITABLE TECHNIQUE ........................................... 64
5.5 DATA FLOW DIAGRAMS IN DETAIL ................................................................. 65
  5.5.1 Application of DFDs to the Building Design Model ..................................... 65

Chapter Six:
THE DESIGN PROCESS MODEL

6.1 INTRODUCTION ................................................................................................... 70
6.2 PILOT STUDY ....................................................................................................... 70
  6.2.1 Model Formation ............................................................................................ 70
  6.2.2 Pilot Study Projects ....................................................................................... 70
  6.2.3 Pilot Study Observations ............................................................................... 72
  6.2.4 Design Process Model Proposals ................................................................... 73
  6.2.5 Modelling Approach ..................................................................................... 76
6.3 DEVELOPING THE DESIGN PROCESS MODEL .................................................. 78
  6.3.1 Data Acquisition for the Design Process Model .......................................... 78
  6.3.2 An Overview of the Extent of the Design Process Model ............................. 80
  6.3.3 Model Hierarchy .......................................................................................... 82
  6.3.4 The Data Flow Diagrams .............................................................................. 85
6.4 VERIFICATION AND VALIDATION OF THE DESIGN PROCESS MODEL ............... 95
  6.4.1 Definitions .................................................................................................... 95
  6.4.2 Verification .................................................................................................. 96
  6.4.3 Validation .................................................................................................... 98
6.5 APPLICATIONS OF THE DESIGN PROCESS MODEL ......................................... 99
  6.5.1 Developing Good Design Practice ................................................................ 99
  6.5.2 Design Planning .......................................................................................... 99
  6.5.3 Analysis of Change ...................................................................................... 100
  6.5.4 Design Process Analysis ............................................................................ 100
6.6 CONCLUSIONS .................................................................................................... 101

Chapter Seven:
DEVELOPMENT OF A PROTOTYPE DESIGN PLANNING METHODOLOGY

7.1 INTRODUCTION ................................................................................................... 103
7.2 DESIGN STRUCTURE MATRIX ANALYSIS ...................................................... 103
  7.2.1 History ........................................................................................................ 103
  7.2.2 Representing the Design Process ................................................................ 104
  7.2.3 Task Order Within a Matrix ...................................................................... 105
  7.2.4 Partitioning the Matrix .............................................................................. 106
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.5 Tearing the Matrix</td>
<td>107</td>
</tr>
<tr>
<td>7.2.6 Product Design</td>
<td>108</td>
</tr>
<tr>
<td>7.2.7 Building Design</td>
<td>108</td>
</tr>
<tr>
<td>7.3 PROTOTYPE DESIGN PLANNING METHODOLOGY</td>
<td>109</td>
</tr>
<tr>
<td>7.3.1 Summary</td>
<td>109</td>
</tr>
<tr>
<td>7.3.2 Step One: Analysing the DPM</td>
<td>111</td>
</tr>
<tr>
<td>7.3.3 Step Two: Design Structure Matrix Analysis</td>
<td>114</td>
</tr>
<tr>
<td>7.3.4 Step Three: Producing the Logic Network</td>
<td>117</td>
</tr>
<tr>
<td>7.3.5 Step Four: The Design Programme</td>
<td>118</td>
</tr>
<tr>
<td>7.3.6 Conclusions on the Prototype Methodology</td>
<td>118</td>
</tr>
<tr>
<td>7.4 CONCLUSIONS</td>
<td>119</td>
</tr>
</tbody>
</table>

### Chapter 8:

#### REFINEMENT OF THE DESIGN PLANNING METHODOLOGY

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 INTRODUCTION</td>
<td>121</td>
</tr>
<tr>
<td>8.2 SUMMARY OF ADePT</td>
<td>121</td>
</tr>
<tr>
<td>8.2.1 The Procedure</td>
<td>121</td>
</tr>
<tr>
<td>8.2.2 The Design Example</td>
<td>124</td>
</tr>
<tr>
<td>8.3 IMPROVEMENTS TO DESIGN STRUCTURE MATRIX ANALYSIS</td>
<td>126</td>
</tr>
<tr>
<td>8.3.1 Link Between the DPM and DSMA</td>
<td>126</td>
</tr>
<tr>
<td>8.3.2 DSMA: The Problem</td>
<td>127</td>
</tr>
<tr>
<td>8.3.3 Classifying Information Dependencies</td>
<td>130</td>
</tr>
<tr>
<td>8.3.4 Examples of varying information Classifications</td>
<td>132</td>
</tr>
<tr>
<td>8.3.5 Applying Information Classifications</td>
<td>135</td>
</tr>
<tr>
<td>8.3.6 Robustness of Classified Information Dependencies</td>
<td>138</td>
</tr>
<tr>
<td>8.4 IMPROVEMENTS TO DESIGN PROGRAMMES</td>
<td>139</td>
</tr>
<tr>
<td>8.4.1 Design Programmes: The Problem</td>
<td>139</td>
</tr>
<tr>
<td>8.4.2 Development of Logic Networks</td>
<td>140</td>
</tr>
<tr>
<td>8.4.3 Linking the Logic Network to a Project Management Package</td>
<td>143</td>
</tr>
<tr>
<td>8.5 INCORPORATING INTERNAL AND EXTERNAL INFLUENCES</td>
<td>144</td>
</tr>
<tr>
<td>8.5.1 Internal and External Influences on Design: The Problem</td>
<td>144</td>
</tr>
<tr>
<td>8.5.2 Internal Influences on the Design Process</td>
<td>146</td>
</tr>
<tr>
<td>8.5.3 External Influences on the Design Process</td>
<td>146</td>
</tr>
<tr>
<td>8.5.4 Using DSMA to Study the Impact of Construction Influences.</td>
<td>147</td>
</tr>
<tr>
<td>8.6 OTHER APPLICATIONS OF ADePT</td>
<td>149</td>
</tr>
<tr>
<td>8.6.1 Studying the Effect of Change</td>
<td>149</td>
</tr>
<tr>
<td>8.6.2 Using the Logic Networks to Programme Design Reviews</td>
<td>151</td>
</tr>
<tr>
<td>8.7 PARTITIONING ALGORITHMS</td>
<td>152</td>
</tr>
</tbody>
</table>
Chapter Nine:
CONCLUSIONS, REFLECTIONS AND FUTURE WORK

9.1 INTRODUCTION ........................................ 163
9.2 CONCLUSIONS ........................................... 163
   9.2.1 Overview & Problem Formulation .................. 163
   9.2.2 Development of the Design Process Model ........ 164
   9.2.3 Development of ADePT ............................ 166
9.3 REFLECTIONS ........................................... 167
9.4 FUTURE WORK .......................................... 169

REFERENCES:
Thesis References ........................................... 172
Appendix I:
THE GENERIC DESIGN PROCESS MODEL HIERARCHY

The Generic Design Process Model Hierarchy ........................................ (i)
Civil Engineering Design Process Hierarchy ......................................... (ii)
Architectural Design Process Hierarchy .............................................. (x)
Mechanical Services Engineering Design Process Hierarchy ................... (xvii)
Structural Engineering Design Process Hierarchy .................................. (xxvii)

Appendix II:
TYPICAL INFORMATION DEPENDENCY LIST

Typical Structural Engineering Design Task Dependency Lists ................. (xxxv)
Typical Civil Engineering Design Task Dependency Lists ..................... (xxxviii)

Appendix III:
DATA FLOW DIAGRAMS IN THE DESIGN PROCESS MODEL

Data Flow Diagrams in the Design Process Model ................................ (xlii)
Data Flow Diagrams for Structural Engineering Design ......................... (xliii)
Data Flow Diagrams for Civil Engineering Design .................................. (li)
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Thesis Breakdown</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Addis' Black Box Model of the Design Process</td>
<td>14</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Trends in Component Costs of Non-Residential Buildings</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>A Linked Bar Chart</td>
<td>26</td>
</tr>
<tr>
<td>Figure 2.4(a)</td>
<td>An Activity on the Arrow Network</td>
<td>28</td>
</tr>
<tr>
<td>Figure 2.4(b)</td>
<td>An Activity on the Node Network</td>
<td>28</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>A Typical Cyclic or Iterative Process</td>
<td>31</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Possible Sequences for Two Design Tasks</td>
<td>31</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Research methodology</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Luckman's Descriptive Model</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>French's Prescriptive Model</td>
<td>44</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Pugh's Total Design Model</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Typical Section of Wix's Model of Mechanical Services Design</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Notation for Wix's Model of Mechanical Services Design</td>
<td>50</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>The RIBA Plan of Work Structure</td>
<td>51</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Basic Associations for Entity Relationship Diagrams (ERDs)</td>
<td>55</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>A Typical Entity Relationship Diagram</td>
<td>55</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>A Typical Entity Relationship Diagram</td>
<td>56</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>A Typical Data Flow Diagram</td>
<td>57</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Structure Chart Notation</td>
<td>59</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>A Typical Structure Chart</td>
<td>59</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>Role Activity Diagram (RAD) Notation</td>
<td>60</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>A RAD Depicting the Collection and Analysis of a Client Brief</td>
<td>61</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>SADT Notation</td>
<td>62</td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>A Typical SADT Example</td>
<td>63</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>A Control Data Flow Diagram</td>
<td>66</td>
</tr>
<tr>
<td>Figure 5.12</td>
<td>DFD Context Diagram For Storm Drainage</td>
<td>67</td>
</tr>
<tr>
<td>Figure 5.13</td>
<td>Diagram 0 : DFD For Storm Drainage Calculations</td>
<td>68</td>
</tr>
<tr>
<td>Figure 5.14</td>
<td>Diagram 2 : DFD for Calculating Pipe Sizes</td>
<td>69</td>
</tr>
<tr>
<td>Figure 6.1</td>
<td>Simplification of the RIBA Plan of Work Terminology</td>
<td>71</td>
</tr>
<tr>
<td>Figure 6.2</td>
<td>DFD for the Sizing of Retaining Wall</td>
<td>74</td>
</tr>
<tr>
<td>Figure 6.3</td>
<td>A Typical Iterative Loop</td>
<td>75</td>
</tr>
<tr>
<td>Figure 6.4</td>
<td>A Typical Repetitive Loop</td>
<td>75</td>
</tr>
</tbody>
</table>
List of Figures

Figure 6.5 : Work to Date on the Design Process Model ........................................ 80
Figure 6.6 : Section of the Design Process Model Hierarchy .................................... 83
Figure 6.7 : Relationship Between DPM Hierarchy and DFDs .................................. 85
Figure 6.8 : The Context Diagram .............................................................................. 86
Figure 6.9 : Diagram 0 : The Detail Design Process .................................................. 87
Figure 6.10 : Diagram 6 : Document Control ............................................................... 88
Figure 6.11 : Diagram 4 : Civil Design ........................................................................ 89
Figure 6.12 : Diagram 4.1 : Civil Co-ordination ......................................................... 90
Figure 6.13 : Diagram 4.2 : Foundation Design ............................................................ 91
Figure 6.13(a) : Diagram Showing Need to Group Information Hierarchically ............. 92
Figure 6.14 : Diagram 4.2.1 : Pile Design .................................................................. 93
Figure 6.15 : Diagram 4.2.2 : Pile Cap Design .............................................................. 93
Figure 6.16 : Hierarchical Breakdown of Information Flows ........................................ 94
Figure 6.17 : Information Filtering ............................................................................... 101

Figure 7.1 : Directed Graph of Typical System ............................................................ 104
Figure 7.2 : Precedence Matrix Developed from Directed Graph ................................. 105
Figure 7.3 : Partitioned Matrix .................................................................................... 106
Figure 7.4 : Design Structure Matrix .......................................................................... 107
Figure 7.5 : Prototype Design Planning Methodology .................................................. 110
Figure 7.6 : Plant Room Scheme Drawings .................................................................. 111
Figure 7.7 : Plant Room Design Process Model Hierarchy .......................................... 112
Figure 7.8 : Data Flow Diagram for 'Frame Design' ..................................................... 112
Figure 7.9 : Data Flow Diagram for 'Air Handling Unit Design' ................................... 113
Figure 7.10 : Precedence Matrix ................................................................................ 114
Figure 7.11 : Partitioned Matrix .................................................................................. 115
Figure 7.12 : Partitioned Matrix with Tears Identified .................................................. 115
Figure 7.13 : Design Structure Matrix ....................................................................... 116
Figure 7.14 : Network developed From DSM with Iteration ....................................... 117
Figure 7.15 : Network developed From DSM without Iteration ................................. 118

Figure 8.1 : Stages within ADePT ............................................................................... 122
Figure 8.2 : The Design Example : A Plan of an Industrial Unit .................................. 124
Figure 8.3 : Data Transfer Between the DPM and DSMA ........................................... 126
Figure 8.4 : Precedence Matrix for Design Example ................................................... 128
Figure 8.5 : Partitioned Matrix for Design Example ..................................................... 129
Figure 8.6 : Contrasting Influences on Information Classification ............................... 131
Figure 8.7 : Information Flow Classification Flow Chart ............................................ 132
Figure 8.8 : Foundation GA Drawing DFD .................................................................. 133
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.9</td>
<td>Ground Floor GA Drawing DFD</td>
<td>133</td>
</tr>
<tr>
<td>8.10</td>
<td>Calculation of Electrical Loadings DFD</td>
<td>134</td>
</tr>
<tr>
<td>8.11</td>
<td>Storm Drainage Layout Drawing DFD</td>
<td>134</td>
</tr>
<tr>
<td>8.12</td>
<td>Calculation of Vertical Loads on Building DFD</td>
<td>134</td>
</tr>
<tr>
<td>8.13</td>
<td>Classified Precedence Matrix for Design Example</td>
<td>136</td>
</tr>
<tr>
<td>8.14</td>
<td>Design Structure Matrix for Design Example</td>
<td>137</td>
</tr>
<tr>
<td>8.15</td>
<td>Relationship Between Information Classification and Iteration</td>
<td>140</td>
</tr>
<tr>
<td>8.16</td>
<td>Logic Network Developed from Design Structure Matrix</td>
<td>141</td>
</tr>
<tr>
<td>8.17</td>
<td>Simplified Logic Network Developed from Design Structure Matrix</td>
<td>142</td>
</tr>
<tr>
<td>8.18</td>
<td>Unravelled Section of Logic Network</td>
<td>144</td>
</tr>
<tr>
<td>8.19</td>
<td>Design Programme Developed for Design Example</td>
<td>145</td>
</tr>
<tr>
<td>8.20</td>
<td>Impact of Fast Tracking Foundation Design</td>
<td>148</td>
</tr>
<tr>
<td>8.21(a)</td>
<td>Identifying Loops Using Boolean Algebra</td>
<td>153</td>
</tr>
<tr>
<td>8.21(b)</td>
<td>Identifying Loops Using Boolean Algebra</td>
<td>153</td>
</tr>
<tr>
<td>8.22(a)</td>
<td>Identifying Loops Using Boolean Algebra</td>
<td>154</td>
</tr>
<tr>
<td>8.22(b)</td>
<td>Identifying Loops Using Boolean Algebra</td>
<td>154</td>
</tr>
<tr>
<td>8.23(a)</td>
<td>Identifying Loops Using Boolean Algebra</td>
<td>155</td>
</tr>
<tr>
<td>8.23(b)</td>
<td>Identifying Loops Using Boolean Algebra</td>
<td>155</td>
</tr>
<tr>
<td>8.24</td>
<td>Fully Partitioned Matrix Using Boolean Algebra</td>
<td>155</td>
</tr>
<tr>
<td>8.25(a)</td>
<td>Identifying Loops Using Path Searching</td>
<td>156</td>
</tr>
<tr>
<td>8.25(b)</td>
<td>Identifying Loops Using Path Searching</td>
<td>156</td>
</tr>
<tr>
<td>8.26(a)</td>
<td>Identifying Loops Using Path Searching</td>
<td>157</td>
</tr>
<tr>
<td>8.26(b)</td>
<td>Identifying Loops Using Path Searching</td>
<td>157</td>
</tr>
<tr>
<td>8.27(a)</td>
<td>Identifying Loops Using Path Searching</td>
<td>157</td>
</tr>
<tr>
<td>8.27(b)</td>
<td>Identifying Loops Using Path Searching</td>
<td>157</td>
</tr>
<tr>
<td>8.28</td>
<td>Fully Partitioned Matrix Using Path Searching</td>
<td>157</td>
</tr>
<tr>
<td>8.29</td>
<td>Design Structure Matrix Developed with TERABL</td>
<td>159</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table 1.1</th>
<th>Breakdown of Thesis</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 5.1</td>
<td>DFD Element Descriptions</td>
<td>56</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>DFD Element Descriptions for the Design Process Model</td>
<td>66</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Comparison of Projects Studied in the Pilot Study</td>
<td>71</td>
</tr>
<tr>
<td>Table 7.1</td>
<td>A Precedence Table</td>
<td>114</td>
</tr>
<tr>
<td>Table 8.1</td>
<td>A Precedence table for Design Example Tasks</td>
<td>125</td>
</tr>
<tr>
<td>Table 8.2</td>
<td>Variation Effect Trace</td>
<td>151</td>
</tr>
</tbody>
</table>
SUMMARY

Much of the focus of the industry's research over the last two or three decades has been dedicated to improving performance on site, both in terms of management and construction techniques. This is in marked contrast to the work aimed at improving performance during the design phase. Historically, design has been manageable without the help of special planning and management techniques, whereas in construction there have been clearer, more easily realisable benefits. However, the relatively small cost of the design activity, compared to construction belies its true importance to the project as a whole. As buildings become technically more complex and design teams more specialised and fragmented, the need to plan and co-ordinate the design process with greater accuracy is becoming increasingly important.

Two major weaknesses in current building design planning were identified: the first is the difficulty planners have in identifying and understanding the complex inter-relationships between different design tasks from differing disciplines; the second is the unsuitability of traditional planning techniques to produce workable design programmes because of the iterative nature of design. The aim of this research was to develop a more sophisticated approach to the planning of building design work. This has been achieved by the development of a prototype methodology ADePT (Analytical Design Planning Technique). ADePT is based on two concepts: a Design Process Model and Design Structure Matrix Analysis. The Design Process Model (DPM) was developed using data flow diagrams to overcome the first of the problems described above providing a clearer understanding of the building design process for planners. A combination of the Design Process Model with Design Structure Matrix Analysis (DSMA) forms ADePT, which has many advantages over traditional planning techniques.

In current practice planners draw up new design programmes by determining the tasks and the links afresh at the beginning of each project. For large projects this is not only time consuming but subject to the limitations of the planner's knowledge and experience, often resulting in simplistic design programmes in which many of the dependencies are ignored. The Design Process Model eliminates much of this subjectivity by generically representing the tasks involved, and the information flowing in the design of any building. It is argued that the approach of developing design programmes from a model has several advantages over current planning techniques, producing better and more realistic design programmes, eliminating errors and reducing re-design. The DPM captures cross-disciplinary knowledge in a unified format, giving individual planners a greater understanding of the design process, that can be used repeatedly to produce multi-disciplinary design programmes, based on different design options, for different projects.
Summary

The second reason for poor design planning, associated with the unsuitability of traditional planning tools, has been addressed within ADePT by utilising Design Structure Matrix Analysis (DSMA), a technique used sporadically in the field of production engineering. DSMA analyses tasks and their dependencies represented within the Design Process Model and schedules the tasks in an order based on the optimal flow of design information. This property of ADePT is seen as critical because the effective management of information is crucial to the success of any project. It can also represent interdependent tasks and schedule them to minimise the number, and size of iterative cycles within the design process. DSMA has been refined to incorporate the classification of the relative importance of each information flow within the DPM and although a subjective process, this approach has been largely validated because the analysis highlights groups of tasks requiring detailed integration, which coincide with the unco-ordinated design problems often discovered during construction.

Traditionally design plans are drawn up to suit the requirements of construction and procurement, with design tasks scheduled to accommodate the important milestones, such as 'start on site dates' or 'out to tender dates'. ADePT works in the reverse order, firstly deriving the unconstrained design programme based on the optimal flow of design information and then modifying it to accommodate such influences as construction and procurement. ADePT allows the impact of external influences such as client instruction, construction programme, materials procurement, work packaging or resourcing demands to be superimposed on the unconstrained design programme, allowing their influence on the optimal design task order to be assessed. This particular attribute of ADePT can be used with a client to indicate clearly the impact their decisions have on the design process.
Chapter 1
Chapter One

INTRODUCTION

1.1 BACKGROUND TO RESEARCH

The growing complexity of modern buildings in a highly competitive market-place has significantly increased the pressures on contractors as they attempt to complete their projects on time and within budget. Phased construction and fast-tracking methods which overlap design with construction are now common place and have been introduced to reduce project durations and overheads; although generally successful, these techniques have placed added pressures on the whole building process, especially the design phase (Austin et al. 1993).

Much of the focus of the industry's research over the last two or three decades has been dedicated to improving performance on site, both in terms of management and construction techniques (Austin et al. 1993). This is in marked contrast to the work aimed at improving performance during the design phase, an imbalance partially explained by Edlin (1991), who suggests that as design only accounts for 3-10% of the total project cost the greatest savings, in financial terms, can be made most easily by concentrating on improving construction efficiency. Historically, design has been manageable without the help of special planning and management techniques, whereas in construction there have been clearer, more easily realisable benefits (Austin et al. 1995). However, the relatively small cost of the design activity, compared to construction belies its true importance to the project as a whole. Glavan & Tucker (1991) have shown how many minor design related problems significantly affect construction performance, an observation supported by the BEDC Report (1987) which found that the majority of construction problems are related to poor design information.

Good management is a key factor in promoting efficiency and systematic methods of working in any process. It is becoming increasingly apparent that the design phase, as well as the construction phase, requires effective management to ensure the requirements of the client are met satisfactorily. Various management philosophies, such as Value Engineering, Value Management and Quality Assurance have been developed, mainly at the insistence of clients, to aid the design process and improve the final product. However, the extra demands on the contractor caused by this drive for greater efficiency, have been compounded by traditional procurement techniques which divorce the design activity from construction, making the management of the overall project all the more difficult. These problems have long been recognised; committees, such as that of Emmerson (1962) and
Banwell (1964), have concluded that in no industry are the processes of manufacturing and design so far apart. While many of the problems that affect design are common across the engineering professions, the building industry has been particularly slow to understand the benefit of, and profit from the research performed by other design professions. Since the 1950's many researchers, including Jones (1953) and Alexander (1964), have striven to gain an analytical understanding of the process of engineering design and suggest management frameworks within which design could be more systematically performed. Unfortunately, very little of this knowledge has been disseminated and developed within the construction industry (Austin et al. 1994b).

The design and development of any technically complex product involves large numbers of design personnel making thousands of decisions, sometimes over a period of years. Very rarely are these decisions performed in isolation (Austin et al. 1994a). Traditional building design practices are being increasingly replaced by multi-disciplinary practices (Bennett et al., 1988), which encourage and ease information transfer between professions. The disadvantage is that communication is often informal and not documented, making the management of design more difficult. The successful performance of large multi-disciplinary projects requires enormous co-ordination to ensure all parties are constantly aware of the ever changing status of the project, so that design errors may be eliminated and design changes kept to a minimum. However, these ideals are rarely achieved.

The growing awareness that specifically tailored techniques are necessary to help manage the building design process was the catalyst for this project. In one sense the management of building design is the management of information. If design is to be managed properly, the activities involved need to be planned to a greater level of detail to ensure that information is produced, co-ordinated, disseminated, monitored and controlled effectively; it will be shown in the forthcoming chapters that existing programming techniques (i.e. network analysis) and existing methods of planning (i.e. via the production of drawings) are not sophisticated enough to achieve this.

Many of the problems outlined above were reinforced by the author's own experience while working for AMEC Design and Management, a large multi-disciplinary design organisation specialising in the development of technically complex buildings, such as laboratories for the pharmaceutical industry and facilities for food production. Despite being one of the largest multi-disciplinary organisation in the UK, with the maturity of nearly forty years inter-disciplinary working behind them, AMEC were far sighted enough to realise that the multi-disciplinary approach was not the panacea often made out and many problems still endured: where the planning of activities for individual disciplines was performed centrally by a planner, experience showed that the programmes produced
were insufficiently detailed to co-ordinate the flow of individual items of design information; in contrast when the day to day planning was left to designers within their own discipline, it was found that many of the critical cross-disciplinary interfaces were subsequently being ignored. Both methods invariably led to changes and variations, often resulting in a great deal of abortive work. A lack of understanding of the process of design, at the most detailed level, not only makes the planning and co-ordination of disciplines' work more difficult but makes the pricing and control of variations onerous.

This research project was set up to investigate the issues outlined above and to develop more efficient techniques to aid design management.

1.2 AIMS AND OBJECTIVES

The global objective at the start of this research was to examine the building design process and resolve some of the conflicts that make building design such a problematical and difficult process to manage. This was achieved by making a detailed study of the numerous factors that have a detrimental influence on a smooth progression through the design process. From this initial work the overall aim of the research was evolved and can be stated as thus:-

The development of more a sophisticated approach to the planning and co-ordination of multi-disciplinary building design.

This aim was approached by splitting the research in two distinct phases, each with their own objectives; in summary these were:-

Phase I: Objectives
(i) The development of a clearer understanding of the building design process to determine:
   • the key activities performed by each discipline;
   • the information flowing between tasks at all levels;
   • the critical interfaces (information flows) between disciplines;
   • the critical tasks within the design process;
   • the size and nature of iterative processes in design work;
   • the stages of the design process that would benefit most from a more efficient approach to planning;
   • the different forms of output from design; and
Chapter 1

- the aspects of the building design process which are not of primary importance to the function of producing design output (drawings, calculations etc.).

(ii) To develop models of the building design process that represent design tasks and information dependencies in a generic manner so that the model could be used, in the short term, to:

- aid planners using traditional techniques in the production of design programmes;
- gain an appreciation of the complex interactions between members of the design team;
- permit a more rigorous comprehension of the effects of change and variations;
- educate and inform designers of the information required for, and more importantly, from their individual design tasks; and
- give an indication of good design practice within an organisation.

Phase II: Objectives

(i) To develop a new planning methodology that could cope with the many complexities inherent in building design work. The methodology would utilise the models developed in the earlier stages of the research, so that data on design tasks and their information dependencies could be extracted from a generic model to formulate design programmes for different projects. The overall methodology developed would permit:

- the scheduling of tasks based purely on the optimal flow of information;
- the programming of iterative tasks;
- an optimisation of the number and size of iterative loops;
- the incorporation of differing design options into design programmes; and
- the scheduling of tasks to account for the phased information needs of the contractor or client.

1.3 RESEARCH METHODOLOGY

The methodology followed to meet the objectives of the two discrete phases outlined above and hence achieve the overall aim of the research, is summarised below:

(i) Existing and ongoing projects were studied and formed the main method of data collection so that:

- a clearer understanding of the building design process could be gained; and
Chapter 1

- concepts and philosophies for producing models of the building design process could be developed.

(ii) Numerous models of the design process and modelling techniques were examined to find the most suitable way of constructing models of the building design process.

(iii) Generic models were produced of the building design process and verified by a series of structured walk throughs with practising designers.

(iv) Techniques for converting the data stored in the models into workable design programme were investigated.

(v) The refinement and verification of the overall design planning technique was performed on selected case studies.

1.4 RESULTS OF RESEARCH

1.4.1 Research Output

The overall aim of this research project, to produce a more sophisticated approach to the planning of building design, was achieved by the development and subsequent refinement of a technique called ADePT (Analytical Design Planning Technique). Two reasons why building design is such a difficult process to plan were identified during the initial stages of this research and these formed the central hypotheses around which ADePT was developed.

(i) Modern buildings are so technically complex that all the processes involved in a multi-discipline design project cannot easily be assimilated by an individual.

(ii) Traditional planning techniques such as network analysis, are not sophisticated enough to handle the many complexities unique to design.

ADePT is based on a Design Process Model that generically represents the design of a building. Developing design programmes from a model is an approach that has several advantages over current planning techniques:-

(i) It captures cross-disciplinary knowledge in a unified format, affording individual planners a greater understanding of the design process which can be used repeatedly to produce varying multi-disciplinary design programmes for different projects.

(ii) As all the tasks and dependencies are defined within ADePT, it obviates the need to 're-invent the wheel' every time a design programme is drawn up.

(iii) It will help eliminate errors in design programme development.

(iv) It allows the choice of varying design options or alternatives to be considered and their implications on the design process to be identified.
Chapter 1

The second reason for poor design planning, associated with traditional planning tools, has been addressed within ADePT by utilising Design Structure Matrix Analysis, a technique that analyses tasks and their dependencies and schedules them in an optimum order. This approach has several benefits over existing techniques because:

• it schedules tasks based solely on the flow of design information;
• it can represent interdependent tasks and schedule them to reduce the number and size of iterative loops; and
• highlights areas where detailed integration is required between certain tasks.

Adaptations and improvements of Design Structure Matrix Analysis have been made to make ADePT more suitable for planning the building design process. While ADePT has been welcomed by practising engineers it is still only a prototype and, as such, requires further refinement and validation to produce a full working system. However, ADePT has been shown to provide the design team with the following advantages:-

(i) ADePT initially produces an unconstrained design programme based purely on the flow of design information. This is an optimum order for performing tasks from a design process perspective and is in marked contrast to traditional techniques which usually work back from dates imposed by the construction process to schedule design tasks.

(ii) ADePT then allows the impact of external influences, such as construction programme, materials procurement, work packaging or resourcing demands to be superimposed on the unconstrained design programme. This allows their influence on the optimal design task order to be assessed.

(iii) It can similarly be used to indicate clearly to the client the impact their decisions have on the design process, such as a start on site date or the desire to fast-track the project.

(iv) ADePT can reveal the most suitable timing of both internal design reviews involving the design team, and external design reviews involving the client.

(v) The technique can be used to predict the effect of change on activities within the design process.

1.4.2 Main Findings
During the work undertaken to develop ADePT several important conclusions were reached regarding the building design process, which justify inclusion in this thesis in their own right; they are :-
(i) The cause of problems associated with building design can be attributed to one of the following five inter-related categories:

- increasingly sophisticated client or employer;
- fast-tracking pressures on design;
- increasing building complexity;
- insufficient information management; and
- difficulty in planning the design phase.

Of these problems only the latter two can be addressed by the design team; the other three are beyond their direct control.

(ii) Building design is poorly planned for two fundamental reasons:

- modern buildings are so technically complex that all the design processes involved in a multi-discipline design project cannot be understood by an individual; and
- traditional planning techniques such as network analysis, are not sophisticated enough to handle the many complexities unique to design, such as iteration or choice.

(iii) Existing models of the design process from both manufacturing and construction domains cannot be used to plan design because they lack sufficient detail.

(iv) Data flow diagrams were found to be a suitable technique to model the design process for the following reasons:

- they are graphical;
- they can be partitioned;
- they are multi-dimensional;
- they emphasise the flow of data rather than control; and
- they represent a situation from the viewpoint of the data rather than from the viewpoint of a person or an organisation.

(v) The following features do not need to be represented in a model that will aid the planning of the building design process:

- management tasks;
- verbal and informal communication; and
- tasks repeated solely as a result of an approval or checking procedure.

(vi) Design Structure Matrix Analysis was found to be a suitable tool for manipulating tasks to optimise the information flow within a process.
Chapter 1

1.5 GUIDE TO THESIS

The thesis can be broken down into four major sections covering the topics shown in Table 1.1 below. This also is shown diagrammatically in Figure 1.1.

<table>
<thead>
<tr>
<th>CHAPTERS</th>
<th>GENERAL TOPIC COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>One - Three</td>
<td>Research problem formulation</td>
</tr>
<tr>
<td>Four - Six &amp; Appendices</td>
<td>Production of the Design Process Model</td>
</tr>
<tr>
<td>Seven &amp; Eight</td>
<td>Development of ADePT</td>
</tr>
<tr>
<td>Nine</td>
<td>Conclusions and recommendations for future work</td>
</tr>
</tbody>
</table>

Table 1.1: Breakdown of Thesis

Below is a summary of the content and purpose of each of the following eight chapters.

**Chapter Two : A Review of Current Design Process Problems**
This chapter discusses the study performed firstly to define the building design process and then to examine the many causes of problems afflicting the design of modern buildings. The chapter goes on to evaluate current methods and techniques of planning building design.

**Chapter Three : Research Objectives and Methodology**
This chapter outlines the methodology followed during this project to ensure the objectives of the research, developed from the work detailed in Chapter 2, were met.

**Chapter Four : Existing Models of the Design Process**
This chapter discusses the investigation of existing design process models from both manufacturing and construction domains. Each model was assessed to determine whether they were suitable and adaptable enough to be developed into detailed models of the building design process.

**Chapter Five : Techniques for Modelling the Design Process**
This chapter outlines the advantages and disadvantages of numerous diagrammatic modelling techniques used in the Information Technology industry to develop pictures and an understanding of systems. The chosen technique, data flow diagrams, is discussed in greater detail specifically in the context of this research.
Chapter 1

Short Comings of Existing Planning Techniques

Why Design is Difficult to Plan

Background to Research

Summary of Design Related Problems

Research Methodology

Concepts Behind the Design Process Model

Philosophy of Data Flow Diagrams

Requirements of A Design Process Model

Constructing the Design Process Model

Existing Design Models

Modelling Techniques Examined

Design Structure Matrix Analysis

Summary of ADePT

Development of ADePT

The Development of ADePT

Future Work Emanating from Research

Recommendations of Research

Conclusions/Future Work

Output from Research

APPENDICES

Figure 1.1: Thesis Breakdown
Chapter 1

Chapter Six : The Design Process Model
This chapter covers the concepts and philosophies developed to produce a generic model of a multi-disciplinary building design process. A typical section of the Design Process Model is examined to reinforce these points. Further sections of the Design Process Model are shown in the Appendices.

Chapter Seven : Development of a Prototype Design Planning Methodology
This chapter explains the initial development of a new prototype design planning methodology. It discusses the concept of analysing the Design Process Model with the technique of Design Structure Matrix Analysis to produce effective and workable design programmes.

Chapter Eight : The ADePT Planning Technique
This chapter discusses the refinement of the prototype planning methodology to produce ADePT, the Analytical Design Planning Technique. It explains and contrasts the improvements by the way of a case study.

Chapter Nine : Conclusions, Reflections and Future Work
This chapter concludes the work done in the three years of research and looks forward to the possible ways in which this project can be further refined and developed.
Chapter 2
Chapter Two

A REVIEW OF CURRENT BUILDING DESIGN PROCESS PROBLEMS

2.1 INTRODUCTION

This chapter discusses the work performed during the early stages of this project, covering a generic study of problems common to all engineering design professions and a more detailed examination of those specific to building design. The aim of this study was to investigate and appreciate design more fully, and to gain a greater understanding of the challenges and difficulties encountered when managing a modern building design process. This allowed the objectives of the research to be narrowed from the general aim of improving design management techniques to that of specifically improving the planning of building design work.

2.2 THE DESIGN PROCESS

2.2.1 Definitions

The word design can be defined both as a verb and as a noun. When used as a noun, design is defined as a preliminary plan or sketch for the making or production of a building, a machine etc. (OED 1990). In contrast, when used as a verb, the OED suggests design is the creation of almost any product: a building, a machine, a picture, a garment. This view is contradicted by BS7000 Part 1 (1989) (Product Design) which states that design is to generate information from which a product can become a reality. The former speaks of the creation of a product, i.e. the making of an entity, whereas the latter refers to the work necessary before its creation. BS7000 Part 4 (1994) (Construction Design) is even more vague, referring to design as the activities required to convert design input into design output. However, the main concern of this project is the management of the design process and therefore the following discussion will concentrate on the definition of the verb 'to design'.

The absence of a single universally accepted definition of a design process in the reference literature is mirrored by the variety of interpretations given by the following authors:-

The ultimate aim of the design process is form.

Alexander (1964).

The design process is a unique combination of problem solving, creative, need fulfilling and human activity processes.

Chapter 2

The design process is the intellectual attempt to meet certain demands in the best possible way.


The design process is the conception, invention, visualisation, calculation, marshalling, refining and specifying of details which determines the form of an engineering design project.


The design process is the creative and personal activity of taking the client's brief to develop a three dimensional interpretation.

Gray, Hughes and Bennett (1994).

No single definition was truly appropriate to building design, and it was therefore necessary to examine the properties of design processes to help clarify and define building design.

2.2.2 The Nature of the Design Process

Suh (1990) stated that design is one of the oldest endeavours purely because people have always had the inherent instinct to design things; a stance he supports with his belief that humanity has been responsible for the design and manufacture of all objects from basic hunting implements through to nuclear power stations. Historically, in traditional craft based societies 'designing' was not separated from 'making', whereas in modern industrial societies the processes have become quite distinct (Cross 1989). Usmani and Winch (1994) suggested there is a dichotomy of views on the nature of modern design processes: the first group, the integrators, believe that the nature and characteristics of design are the same for all professions; the second school of thought, the separators, believe design processes are fundamentally different between industries. Gregory (1966), a true integrator, believed a commonality of design exists across domains and concluded that the process of design is the same whether it deals with the design of a new oil refinery, the construction of a cathedral or the writing of Dantes Devine comedy. This view is reinforced by Stauffer (1989) who suggested that a general methodology, or a single general definition, can be applied to a design process from any domain. In contrast, authors such as Cross (1984) argued that architectural and engineering design are intrinsically different, although this is contradicted in later work when he combined the two extremes in a single consensus model of the design process (Cross & Roozenburg 1992).

The view taken for this thesis is that the basic properties of all design processes, at the highest level of abstraction, are the same. The following properties are common to any design process irrespective of the domain:
Chapter 2

- design generally begins with a need (French 1991);
- design results in information that ultimately leads to a physical process taking place (Slusher et al. 1989);
- design is never comprehensively specified (Whitefield & Warren 1989);
- design never has a single optimum solution (Gregory 1966);
- design is never a single problem but as a series of sub-problems (Cross 1989); and
- design is an iterative process (Eppinger 1991).

Modern design very rarely involves the production of a completely new, unique product; 99% of new cases, to a lesser or greater extent, are a derivative of an existing product (Oakley 1990). Pahl and Beitz (1988) take this view one step further, distinguishing between three different types of design work:

Original Design : involves developing an original solution for a system.

Adaptive Design : which involves adapting a known system for a new task.

Variant Design : which involves varying the size and or the arrangement of certain aspects of a system to suit a new task.

It is believed that original design work is not generally performed in the building industry and that the majority of designers call upon a range of precedents, adapting or varying them to suit the new requirements (Cleland & King 1993). However, whether the ultimate design is original, adaptive or variant the properties of the design processes stated are still valid.

Pugh (1990) uses the first of the two properties listed above to define a concept called the ‘total design process’, which is the systematic activity necessary from the identification of user needs to the realisation of information to create the final product. Pugh’s general definition can be applied to any design process and corresponds to the integrator’s view of design processes. However, to identify and define individual design processes the characteristics of the specific design process must be clarified by defining user needs and the nature of the information produced. The Black Box approach towards design put forward by Addis (1990) conveys a scenario very similar to that of Pugh’s total design but is described diagrammatically rather than descriptively. Addis’ model is shown overleaf in Figure 2.1.
2.2.3 The Building Design Process

Pugh's total design process definition can be applied to the building design process: the need is the client's wish to have a new building and the information produced allows the construction phase to proceed. This is neatly summed up by Harris (1975) who classified building design as the determination of what is to be built and the preparation of instructions necessary to build it.

Many authors, such as Usmani and Winch (1994) and NEDC (1990), highlight the importance of co-ordinated information transfer between the various disciplines as critical to the success of the building design process. The multi-disciplinary nature of information flow is complicated because building design, as with most other forms of design, is an iterative process, in that initially completed design often points to improvements that could have been made at some earlier stage of the process (Jewel 1986). However, all building design processes work towards milestones, at which point the client studies the design proposals, offers opinions and confirms the intention to proceed with the project. Although these milestones vary from project to project they usually include the completion of feasibility studies and the completion of scheme design work. If the project is terminated at one of these points, the solutions produced could not be used to construct the building, i.e. no physical process could take place. This leads us to conclude that the definition of the 'total design process' only holds true when applied to the complete building design cycle, and not to stages within it.

Enlarging on Pugh's concept of total design, it was possible to uniquely define the building design process for the purposes of this Thesis. This in turn allowed the terms building designer and building design to be defined. The descriptions drawn up for this work are described below.
Chapter 2

The building design process is a multi-disciplinary process, performed in a series of iterative steps, to conceive, describe and justify increasingly detailed solutions and costings to meet the needs of the client.

These steps include any work necessary from feasibility studies through to the production of information to allow procurement and construction to proceed. Each step should seek to meet the clients' requirements in the most acceptable, aesthetic, efficient and economic manner.

A building designer is any person who has a direct influence on the building design process.

This could be an architect or a structural engineer. Other likely members of a complete design team, such as a planner, a design manager or an administrator have an indirect influence on the outcome of the design and therefore are not be classified as building designers.

A building design is the ideas and philosophies of the building designer produced at any time during the building design process. This may take the form of an option sketch produced during a feasibility study, a set of calculations, specifications or construction drawings.

It should be noted that a building design is not necessarily information directly used in the formation of the physical product, i.e. a building.

For the sake of simplicity throughout this report the following abbreviations will be used: design process/phase refers to the building design process; designer to building designer; and design to building design.

2.3 PROBLEMS IN BUILDING DESIGN

2.3.1 The Causes

In recent years the building industry has undergone a substantial change (CIRIA Report 100, 1983 & NEDC Report 1990) and buildings must now meet more exacting and varied performance requirements to achieve targets imposed by clients. Design work often involves large numbers of design personnel making thousands of decisions, sometimes over a period of years. One way in which the industry has responded to these challenges is the increasing replacement of traditional professional practices by those of a multi-discipline basis (Bennett et al. 1988). These practices encourage and ease information
transfer between professions, but have the disadvantage that the communication is often informal and not documented, making design management more difficult. Although all stages of large multi-disciplinary projects require enormous amounts of co-ordination, it is in the detail design phase, where the quantity of information and number of personnel are at their greatest, that these problems are most acute. For this, and reasons outlined later in the thesis (Section 6.2.4), the review of design processes focused mainly, but not exclusively, on the detail design phase.

A study of current literature from all the engineering professions, backed up by interviews with various building design professionals and from the author's own personal experience, gave an insight into the strengths and weaknesses of current building design practice. From this it was concluded that all problems experienced during detail design processes can be attributed to one of the following five categories:-

(a) Increasingly sophisticated client or employer.
(b) Fast-tracking pressures on design.
(c) Increasing building complexity.
(d) Insufficient information management.
(e) Difficulty in planning design work.

These categories are inter-related but are discussed separately below in more detail.

2.3.2 Client Sophistication

The client has a very influential role to play during the design process and design teams are becoming increasingly aware that more efficient solutions can be developed if the client is integrated into 'The Team' (Sawcuzuk 1992). The type and nature of the client can have varying influences, not only on the ultimate design, but the design process itself. This is borne out by the comparison of two projects studied in conjunction with the work described in Chapter 6. The client team for Project A was directly involved in developing design solutions and took the view they were as responsible for the success of the final design as the design team. Project A was completed on time, with neither the design fees, nor the total projected project costs, exceeded. On the other hand Project B, designed by the same organisation, overran on time and on budget. This client took the view that the design should be left to the 'experts' and that his only input should be at various pre-determined design approval meetings; these being the only points at which the client could influence or change the design. When changes occurred, either through a misinterpretation of the brief or from a client induced change, the result was a great deal of aborted work, the cost of which was ultimately carried by the design organisation.

A significant change in the past three decades has been the emergence of the corporate client (Reading 1982). Whilst the traditional client might have been a local factory owner,
Chapter 2

the modern client is likely to be a representative of a large industrial concern, responsible for delivering a multi-functional building to suit the needs of diverse and often conflicting user groups. Corporate clients tend to manage their new projects via a committee, rather than a single individual. Committees are often made up of a project manager, representatives of the user groups, financial advisors and increasingly frequently members of the client’s own design staff. Committees, by their very nature, tend to be a very cumbersome, but democratic, way of making decisions and if poorly managed can have several detrimental effects on the design process. In a survey of design professionals, Cole (1990) found the major criticism levelled at clients was the speed with which they gave authorisation to sections of design work, often resulting in a slippage in programme or the design team proceeding with a solution the client could possibly change. This indecisiveness is often a direct consequence of management by committee. Effective management by the client can be facilitated by quickly resolving the conflicting interests of its members to prevent these problems occurring.

Obtaining information from the client, other than design approvals, sometimes causes difficulties. It is essential that each member of the design team knows and understands the correct communication channels through which they can seek authorisation or gain design information. Proper emphasis needs to be placed on the client in making timely decisions to ensure design information is always available to meet the needs of the design team (CIRIA Report 100, 1983).

The construction industry has a reputation, often well deserved, for producing buildings which overrun on time and cost and have not met the requirements of the client. A modern corporate client is likely to have experience of building procurement: a recent study suggests that the experienced client now accounts for over 75% of project clients (Gray et al. 1994). This has resulted in clients demonstrating a growing critical interest in their project’s development, placing new demands on the building industry in terms of cost and quality. This has led to new procurement systems (Franks 1984, Turner 1991) and the adoption of new procedures such as Value Engineering (Dell’Isola 1982 & Zimmerman & Hart 1982) and Quality Assurance systems (CIRIA Report 49, 1987 & Duncan et al. 1990) to ensure the parameters of time, quality and cost are met to the client's satisfaction. In an attempt to keep track of the complex technical aspects of new construction, clients often employ their own in-house design staff. Whether this has improved the quality of the end product remains to be seen; the extra checking and consultation in the author’s experience undoubtedly delays the overall completion of the project.
2.3.3 Fast-Tracking

Fast-tracking of construction projects has emerged as a popular and successful way of reducing the duration of construction work. For clients in today's competitive market, accelerating the delivery of a building is important to:

• get an early return on their investment (use the building sooner);
• minimise the effects of fluctuating interest rates and inflation; and
• beat rivals to the market place.

The compression of project duration is achieved by overlapping the design and construction of individual work packages. A trade off for reducing contract schedule dates has been the increased number of problems afflicting the design phase (Glavan & Tucker 1991). Accelerating the whole construction process puts added pressures on designers to ensure contract documents are fully co-ordinated, accurate and issued on time. The problems are intensified by the necessary overlapping of the design and construction phases. The impact of fast-tracking on design is best illustrated in two studies by Burati et al. (1992) and Fazio et al. (1988). Burati et al. found that deviations from total estimated project cost, for fast-track projects, were found to be on average 12.4%, of which 78% were design deviations. Fazio et al. found similar results when studying project delays. They found that 66% of all delays could be attributed directly or indirectly to fast-tracking the project. Of this total 48% of delays occurred during design and 52% occurred during construction. The surprisingly high latter figure indicates the influence design errors have on the construction process. A similar conclusion is arrived at by Glavan & Tucker (1991) who suggests the majority of construction delays can be traced back to minor design conflicts.

Many of these conflicts arise from poor co-ordination, not only within the design team but between design and construction. Poor co-ordination leads to omissions and errors which, in turn, may not be picked up during checking; from discussions with design professionals this is often neglected in an attempt to meet scheduled tender dates. The prompt arrival of vendor information and ensuring the scope of the project is accurately defined in the client's brief also becomes more significant as the schedule is compressed.

It is essential in fast-tracking that design and construction proceed simultaneously. It is rare that a complete set of working drawings is made available before site work commences (Kwakye 1991). The influence of construction on design means much of the project cannot be designed in a logical order. The building foundations, for instance, ideally should be designed towards the end of the design phase when accurate information on the loading from the superstructure is known. Fast-tracking the project means the foundation design is required earlier than normal, forcing the foundation package to be designed out of sequence, before accurate design information on loading becomes
available. To compensate for this, the foundations need to be over designed to ensure the eventual loads do not exceed the capacity of the constructed foundations. Obviously sub-optimal design, such as this, is often inefficient from a structural point of view.

Condensing the construction phase requires quick and easy methods and materials of construction. This increased buildability must be considered during design. High use of prefabricated materials and components are often encouraged, as is the avoidance of wet trade operations like concreting. Steel framed structures are prevalent as they can be constructed easily, arrive on site prefabricated, reduce site labour requirements and facilitate dimensional accuracy. This limits the choices and options open to the architect and the designers, as the emphasis is on ease of construction rather than the most efficient use of materials to fulfil the functional requirements of the building.

2.3.4 Building Complexity
The complexity of modern day buildings, both in their content and construction, has dramatically increased the pressure on the design team to satisfy the information requirements of the site process. Variety in the choice of construction materials, and the equipment and the paraphernalia associated with buildings, has made the job of determining the optimum solution, for a list of given constraints, a more onerous task. Increasing building complexity has also caused the structure of the design profession to change (Turner 1986); large numbers of designers per project, each with narrower specialities and responsibilities are now the norm on modern day design projects. These two effects have not been conducive to co-ordination and communication within the design team.

According to Groak (1993) current research in the field of building technology has been conducted along two different lines: either gaining a clearer insight of what users require, or a greater understanding of the science of building components. The results of both fields of research have led to technically more complex buildings.

Work in the areas of ergonomics, thermal comfort and building related illnesses has resulted in a clearer appreciation of what users experience in a building; an understanding that has led to an improvement in the internal environment of a building. These advances have led to the growth of intelligent buildings, where complex energy management systems control the internal environment. These systems have affected the content of modern buildings and have resulted in the broadening of the mechanical and electrical engineer’s role over the last three decades. Figure 2.2 shows how the comparative costs of the five major components of non-residential buildings have altered in the last two centuries. The cost of machinery, taken to include all items of plant, relative to other
components has risen sharply. This emphasises the increasing importance of the internal environment control in modern buildings.

![Graph showing trends in component costs of non-residential buildings]

**Figures 2.2: Trends in Component Costs of Non-Residential Buildings**

The interplay of the many components within a building has meant the design of the whole is much more complex than the sum of the parts. For instance the fabric, structure and thermal dynamics of a building are so interdependent that although a clearer understanding has evolved of the science of these components, their combined design has become more complicated. Steward (1981a) conforms to this view by stating that technology is now only advancing by the improved understanding, utilisation and combination of established components that behave according to well established laws and properties. One example of this is the subject of fire engineering where a greater knowledge of how fire spreads and interacts with components has led to the tightening of fire regulations and codes of practice, affording greater safety to the buildings occupants. This has obviously reduced the options available to the architects. Another consequence of the increasing integration of components is the effect of changes or revisions, with a change in one part having a 'domino effect' on other portions of the design (Crabtree *et al.* 1993).

In many large companies design has become a bureaucratic tangle, a process confounded by fragmentation, specialisation, power struggles and delays (Whitney 1988). Building design has progressed from the single master builder, responsible for the design and
construction of a building in the nineteenth century, into a process that involves many highly specialised groups. Design problems are now of such magnitude and complexity that no individual is capable of addressing all aspects of building design. The result has been the gradual crystallisation of individual disciplines, each responsible for certain parts of the design work. Specialisation has obvious benefits, but if the many individual groups are not co-ordinated properly the design work remains fragmented, resulting in errors and inefficiency.

The functional decomposition of a building design into individual disciplines is not always in empathy with the requirements of individual parts of the building. Each specialist works within their own well established paradigms, each with its own view on the most suitable solution (Rzveski 1993). For instance a wall can be a structural component (a load bearing element), an architectural component (an envelope or for aesthetical value) or a mechanical services component (an insulator). Each discipline will have its own viewpoint on the most suitable solution for their requirements; co-ordination and communication is needed across the disciplines to reach a compromise suitable to all in such a situation.

Problems can be heightened if disciplines don't have an understanding of the technology, terminology and philosophy of another disciplines work (Cooper & Jones 1993, Stockburger 1993) and is often impeded by architects and engineers jealously guarding their professional territory (Day & Faulkner 1988). It is also essential that team members know how others use cross-discipline information and how, and in what form, they require it to be structured.

2.3.5 Information Management

Information management has been an area of considerable study ever since the publication of the Emmerson Report in 1962, which highlighted the vital importance of effective communication of information amongst the various participants of a project. The availability, reliability and ease of assimilation of project information are known to be critical to the effective pricing, planning and execution of building design work (CPI 1987). Despite this the content, structuring and timing of all types of information within a project is highly variable. Information in this context concerns both product and process information. Product information relates to the building itself and usually is expressed in terms of drawings, calculations or specifications, whilst process information relates to the process of design rather than the design itself and could be a progress report or a programme. Information is the fuel of design and when managed and co-ordinated incorrectly results in delays, errors and omissions, effecting both the design and construction phases. Any improvements are often frustrated by the increasing number of drawings and documents produced to depict and manage the design of a complex modern building.
Chapter 2

A recent study by Noble (1989) suggests design engineers spend between 20% and 30% of their time searching for and handling information, thus reducing the time available for actual 'engineering' or problem solving. These findings are reinforced by Crabtree et al. (1993) who studied the delays associated with poor co-ordination in product design. This suggests the problems of information management are prevalent in all engineering domains. Poor information management can lead to missing, incorrect or unclear documents which will contain insufficient details and conflicting and unco-ordinated information. These deficiencies contribute significantly to major problems associated with the building industry: technical defects, quality of finished work, the frequency of variations and claims, and the late completion and overspend on projects.

Full and free information for all members of the design team is an appealing idea (Tenah 1984), but the proliferation in quantity of information and documentation in modern construction work makes this an impractical ideal (Bhandari 1978). In fact the trend towards specialisation has meant that few parties require all the available product and process information. Too much information, known as information overload, has been shown by Munday (1979) to have a detrimental effect on design output, resulting in an increase in design error rate, inappropriate generalisations and even the total avoidance of information. An important function of any information management procedure or system, is a filtering or discrimination mechanism that can strip away extraneous data ensuring only information required by individuals is received (NEDC Report 1990, Culley et al. 1992). Stephenson and Naylor (1993) state that very few organisations have invested in systems of this kind.

Attempts have been made to try to improve the management of information via such initiatives as Co-ordinated Project Information, in which co-ordinated conventions for use in detail design have been developed (CPI 1987). This covers a system for identifying similar sections of work under the Common Arrangement of Work Sections (CAWS) notation, which co-ordinates production drawings and links specifications and bill of quantities. While this initiative is undoubtedly a step in the right direction, it has yet to win universal favour throughout the building industry.

Computer Aided Draughting and Engineering (CAD and CAE) also have the potential to manage design information more efficiently, but are not a panacea as it must be realised that computers solely carry out functions that require and provide information (Wix 1986). Managing drawings within a CAD environment is absolutely essential to facilitate the communication of information from one discipline to another. Day and Faulkner (1988) suggest that the layering of drawings can aid selective information transfer. For example, if an architect's floor layout drawing is layered properly, only the basic floor details should be transferred to a structural engineer to calculate the floor loads, rather than superfluous
information such as floor finishes, furniture layouts, which may exist on the complete layout drawings. On a larger information management scale attempts are being made to improve the co-ordination of information between different CAD systems with initiatives such as IGES and STEP (Bjork 1989).

2.3.6 Planning of Design Work

The planning of site work is well established and plays an integral part in ensuring construction work proceeds as smoothly as possible. Traditional design planning is usually based on the deliverables for which the design team is contractually obliged, namely drawings (general arrangements, cross-sections, details, etc.) and specifications. Design 'management' consists of monitoring drawing completion against a planned release schedule. The use of drawings as a guide to the amount of work completed has inherent difficulties. The final specification of the design, namely drawings, does not contain any details of the design process itself (Perlman 1988) and it is often very difficult to assess the nearness to completion of drawings or to quantify the amount of revisions that may be necessary (Edlin 1991). This approach is crude and superficial, giving only a rough guide to progress of design work without consideration of the design activity itself.

A lack of design planning results in conflicting construction documents and insufficient information being available to complete design tasks. In turn these factors often lead to 'management by crisis' and abandonment of any attempt to control the design programme. This hampers any attempts to improve the organisational distribution and control of information which in turn affects the quality of the design.

Many authors agree that in the majority of projects little or no design planning takes place (Gray 1985 & Rizzo 1991). Modern day building design involves many professions: architects, mechanical, civil, structural, electrical, environmental and process engineers, quantity surveyors, estimators and planners. Successful design performance of large multi-disciplinary projects requires an enormous amount of co-ordination to ensure that all cross discipline interactions are facilitated, and all parties are constantly aware of the ever-changing state of the project. An accurate and workable design plan is one advance that would facilitate this (Nicholson 1992).

Work by Cole (1990 & 1992) has examined the reasons why many organisations forgo design planning. He concluded that the common belief amongst both designer and manager is that design does not lend itself to being planned, because it is a creative and iterative process; many managers insisted that the time spent analysing and collecting data for the programme could be utilised more effectively, whereas designers felt that their work was often impeded by factors outside their control which planning could not account.
for. In addition architects felt that planning work in too much detail acted as a straight jacket, suppressing their creativity.

The logic of these views has been contradicted by Rowden & Mansfield (1989), who suggest that many professional design organisations make substantially less profit than they could, largely due to the ineffective planning and control of design work. Under these circumstances design work tends to lack direction and may result in decisions being taken in a sub-optimal order leading to expensive, unforeseen discoveries towards the end of the design process (Whitney 1990).

2.3.7 Possible Solutions
The previous sections have highlighted many of the inter-related problems facing a design team, some of which can be addressed from within the design environment. The challenges created by increasingly sophisticated clients, the complex buildings they require and their desire to fast-track the project can't readily be solved by members of the design team; they could be considered as beyond their control. However, poor design planning and poor information management are questions that can be directly addressed by the design team, indeed these two functions are an integral part of the wider discipline of design management. Solving these two problems go hand in hand; improving the planning means the management of information becomes easier, improving the management of information makes the planning easier. More effective planning should highlight the cross-disciplinary interactions required throughout the design process, in doing so, reducing the time spent chasing and searching for information. An accurate design programme, indicating the source of all cross-disciplinary information, would reduce the quantity of documents studied and ease the problem of information overload.

By finding ways to improve design planning, managing information automatically becomes easier, making design management simpler. This hypothesis is examined in the following sections.

2.4 WHY IS DESIGN PLANNED POORLY?

2.4.1 The Basic Design Planning Procedure
Planning the design activity is fundamental to design management (Gray et al. 1994). Without an accurate design programme a project is more likely to stumble from one crisis to another, making its management and control virtually impossible. It is the author's view that the planning of building design work, when performed, is carried out in much the same way from one organisation to the next, whether it is a multi-disciplinary practice or a collection of individual practices. Planning is usually achieved by the individual
disciplines deriving programmes that are all co-ordinated by a central planning section (Rutter & Martin 1990). A simplified view of this procedure is given in the following section.

The first stage in the production of a design programme is to define the roles and responsibilities of each participant in the project (NEDC Report, 1990). The lead designer for each discipline usually produces a linked bar chart, depicting the proposed order of deliverables required, an estimate of task durations and an indication of the grade of staff required. The scheduling of the proposed activities and their information dependencies is beneficial in its own right as it forces the designer: to concentrate on the task ahead; consider the scheme of work; and examine how their discipline fits into the project as a whole. The individual discipline design programmes may be then co-ordinated centrally, and altered if necessary, to compliment procurement or the construction programme. Apart from planning the progress of work, an accurate design programme, can be used as a benchmark against which the progress of design work can be monitored. For the more predictable detail design phase it facilitates the flow of cross disciplinary information, ensuring work is performed in the correct order, enhancing the chances of accomplishing correct design solutions first time. Cole (1992) suggests other beneficial side effects of having a workable design programme are that it automatically commits the designers to a timetable and helps in the self regulation of their work.

It is the author's view that the existing design planning procedures often fail to deliver workable, representative design programmes because of the fragmented manner in which they are devised. Design programmes are traditionally constructed on a discipline basis and then co-ordinated centrally because as argued in Section 2.3.4, buildings are now so technically complex that a complete understanding and appreciation of the processes and interactions required to design a building are beyond the scope of the individual. If in some way design planning could be performed and co-ordinated by one party then it is believed that more accurate and workable design programmes would be produced.

Design is currently planned in the majority of cases by either network analysis or bar charting, often with the aid of a proprietary software package. Both techniques have been used extensively and successfully to plan the construction phase of both large and small projects. The following sections discuss the suitability of bar charts and network analysis to produce accurate design programmes.

2.4.2 Bar Charts
Bar (Gantt) charts were developed during World War One by Henry L Gantt and are still used to plan and monitor work in the construction industry (O'Brien 1972). Bar charts consist of bars or lines representing activities, superimposed onto a time scale; the length of
Chapter 2

bar or line representing the predicted duration of the activity. This representation produces a readily understood and convenient tool for presenting programming data. The main criticism of the simple bar chart is that they do not show adequately interactions and links between activities. This can be overcome in the simplest of cases by introducing linking lines between dependant activities (McCaffer & Harris 1983). This produces a cascade bar chart which can easily highlight the effect of a programme delay in a linked or dependant activity.

Figure 2.3 : A Linked Bar Chart

During a project, the progress can be monitored and controlled against the proposed programme by introducing a second bar, usually shaded in a different tone, below the bar for planned work duration, see Figure 2.3. This new bar represents the actual work completed for the specific activity and is expressed as a percentage of the planned work duration. This can then be compared with the 'real time' of the project, allowing the project manager to obtain an approximate guide to how smoothly the project is running. Adding the resource requirements to each activity allows the total resource needs to be determined at any time during the project.

Despite their apparent simplicity bar chart techniques are still widely favoured for planning large complex construction projects (Allam 1988, Alkayyi et al. 1993), as they give a clear, simple representation of when, and for how long, each activity should be performed during the project life cycle. In fact bar charts are usually the preferred technique to illustrate a programme of work for schedules developed by more sophisticated techniques, such as network analysis. Although bar charts are an ideal tool for representing information in the form of a design programme, they are not an effective tool to plan the order of complex activities which are highly dependant on one other.
2.4.3 Network Analysis

Network analysis was developed in the US in the late 1950's. Various organisations were experiencing difficulties in planning and controlling their larger, more complicated projects with existing basic planning tools, often resulting in large cost and time overruns. The need for a rigorous technique that could handle the complex inter-dependencies of large multi-activity projects was the catalyst in the development of network based techniques. Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) were two such planning techniques developed to fulfil these requirements.

PERT was developed by the US Navy in 1958 to assist the management of the Polaris missile project (O'Brien 1972). Three thousand different contracting agencies worked on the project, and PERT was used to schedule and programme the many diverse activities. PERT was a success and the project was completed ahead of schedule. CPM was developed at about the same time by E. I. DuPont Co. to meet its construction project management needs (Jewell 1986). Although developed independently both methods are very similar and rely on a network to schedule the sequence of work. The major difference between the two methods is the determination of activity duration estimates. CPM utilises a single expected completion time, while PERT time estimates are based on a weighted average of the most optimistic completion time, the most probable completion time, and the most pessimistic completion time. The following discussion will concentrate on the underlying suitability of network analysis to represent and plan design work rather than the individual merits of CPM and PERT.

Two basic forms of network exist: activity on the arrow networks and activity on the node networks (precedence diagramming). Both clearly show the inter-relationships among activities and are equally valid as a precursor to PERT or CPM analysis. Examples of an activity on the arrow network and an activity on the node network, depicting the same system, are shown in Figure 2.4(a) and 2.4(b) respectively. Both diagrams depict the same tasks and the same dependencies but in a slightly different manner. For example, both networks imply that:

- activities B and C can only proceed after activity A has been completed;
- activity G can only proceed after both activities D and C have been completed; and
- activities D, E and F can happen simultaneously.

The application of activity durations to the networks allows a design schedule to be constructed, depicting when each activities should start within the project life cycle. The application of durations can be achieved using either PERT or CPM techniques.
2.4.4 The Applicability of Network Analysis to Represent and Plan Design

Network Analysis is a valuable tool for expressing the inter-relationships between activities. When it is combined with the concept of critical path and float to determine priorities, it provides a powerful planning technique and is a marked improvement on bar charts. The advent of powerful computer packages has relieved the planner of much of the tedium involved in network analysis.

As with all techniques, network analysis is more suited to certain applications than others. It is widely accepted that network analysis can be used successfully to plan the construction phase of many projects. However, the arguments for its applicability to plan design work are well less documented. Some authors such as Nujhuis (1993) suggest that network analysis, under-utilised in product design management, would prove an efficient tool to predict which activities could be performed in parallel to reduce project durations.
This argument could equally be applied to building design. However, it is this author's view that specific aspects of design work do not encourage network analysis as an effective technique to plan it. These aspects are discussed in the following sections.

The Complexity of Design Work

The process of design is extremely complicated often involving the interactions of many designers, performing hundreds of design activities over a long period of time. Constructing a network to capture this complete process for each individual project would be extremely time consuming and result in a network of magnitude and complexity. Allam (1988) suggests that this impracticality is the single most important reason why network analysis is not fully utilised in construction as a whole. In examples where network analysis has been used to plan design, the networks have had the tendency to define tasks in the large, ignoring the multiple of engineering interactions that need the most careful thought and planning (Eppinger 1991).

The objectives and requirements of all projects vary during their life. A design programme is a working document that must be updated and revised quickly to reflect any changes which may occur. When time is spent on producing a detailed network there will be a need for a repeated input of a high level of expertise to adapt and modify the plans as the project conditions change (Levitt et al. 1988). Adapting a network is not an easy process and in recent years network analysis has received a great deal of criticism for its inability to cope with the dynamic nature of construction projects. These factors alone often lead to the rejection of network analysis for simpler methods of planning (Allam 1988).

Choice in Network Analysis

Planning the construction phase of a project from the relevant drawings and documents is a well defined process. Construction work is deterministic. Design work is not, especially during conceptual and scheme design. It is relatively easy to envisage the scheduling of activities that are definitely going to occur, i.e. the construction of a roof, a foundation or a column. In contrast the design manager or planner must try to plan the design phase before the major decisions on the materials and forms of construction have been taken by the design team. Although important initial decisions will have been made on the nature of the project, such as the number of storeys the building will have, it will be impossible to predict the outcome of all decisions yet to be taken. It is therefore imperative that any planning tool used to plan design allows a design manager to explore the possible implications of the various decisions made during the project's life. Each design decision taken may result in different tasks being performed, which in turn may give rise to new task dependencies, producing different task orders within the programmes.
Network analysis does not incorporate this option of choice or branching. It is best used to plan and schedule activities which are clearly defined and may be confidently expected to occur. Choices are constantly being made during design, and therefore all the possible outcomes of those choices need to be considered when putting together a programme for the proposed work. Network analysis can be used to overcome this problem by producing separate networks to show every possible combination of choices. However, this would be a cumbersome and repetitive process. If you were to draw networks to cover a project with five different decisions, each decision with two alternative choices, then a total of 32 separate networks would need to be produced to cover all eventualities. In the course of the design of any building thousands of decisions are taken and choices made. The production of networks to cover all the possibilities would be impractical.

Iteration in Network Analysis.
Dandy and Warner (1989) emphasises that design work can rarely, if ever, be carried out by working through a sequence of steps. It is an over simplification to regard design as a straightforward flow process moving from the general to the particular to the exact. Design is an iterative process, with different ideas and solutions considered and reconsidered for each problem (Rutter & Martin 1990). Some aspects of the building design process that give rise to iterative cycles are listed below:-

(i) Performing design tasks in a group which are reliant on each other for information means that no task within this cycle can begin until one or more of the information dependencies is estimated to break the dependency loop.
(ii) Detailed design work on the later stages of a project may suggest or demand a rethinking of the earlier schematic stages (Ministry 1967) requiring a reworking of many previously completed tasks.
(iii) The formal approval or rejection of a design proposal by a client.
(iv) An informal discussion of a design proposal with a colleague resulting in some rework.

An example of this cyclic process can be seen in Figure 2.5 which represents a party sending information to a client or regulating authority for approval. The receiver can accept or reject the proposal totally, or accept it with the necessary corrections. It can be seen that depending on results of the approval process the loop of process 2 and 3 could continue indefinitely until the outcome of process 2 results in the acceptance of the idea. In standard networking techniques, this system could not be represented as it is logically impossible to introduce a loop or feedback into network analysis techniques.
Eppinger et al. (1990) backs up these comments by putting forward three possible sequences for two design tasks, see Figure 2.6. Network analysis is capable of representing the first two sequences, where the tasks are either dependent on, or independent of, each other. The third sequence shows interdependent tasks; task A is reliant on task B and task B is reliant on task A. Tasks such as these are deemed to be coupled and need to be solved iteratively, i.e. by initially guessing an input and performing the tasks repetitively until the initial guess is validated. The introduction of coupled tasks into a network is logically impossible. Standard network analysis can only represent the one way progression of tasks.
2.4.5 Improvements to Network Analysis

Several attempts have been made to develop PERT and CPM analysis to overcome many of the shortcomings highlighted in the last three sections. Two methods, Decision CPM and Q-GERT analysis, hybrids of CPM and PERT respectively, have been developed to address some of the criticisms levelled at the original network techniques.

Decision CPM was developed by Crowston and Thompson (1965) to allow differing options, costs and task durations to be considered within a single network. In traditional CPM, two or more differing methods of work could only be compared by constructing separate networks. Rather than considering the permutations in advance the task durations, costs and activity dependencies for all the various options are entered into a single decision network. DCPM then calculates the minimum project cost or duration without needing produce alternative networks.

Pritsker (1979) believes that traditional techniques as they stand are limited forms of modelling and are unable to represent many real life situations. GERT, an advance on PERT, is an acronym for Graphical Evaluation and Review Technique and was developed by Pritsker (1979) to represent:

- multiple branching (choice);
- probabilistic branching; and
- repetition of activities via feedback loops (iteration).

Q-GERT is a further advance on GERT analysis as it can model queuing systems in a graphical form and can schedule more than one project at once (Taylor and Moore 1980).

Q-GERT networks are activity on the branch (arrow) networks with the events (nodes) representing milestones, decision points and, where necessary, queues. Flowing through the network are items called transactions representing either physical or information flows. These are directed through the model according to the branching characteristics at each node. Simulation procedures are then used to analyse the networks to give the outcome of possible activities and the operational statistics of project durations.

The two techniques described, DCPM and Q-GERT, are both improvements on standard network based analysis. They address the problem of choice (branching) and iteration (feedback). However, both these techniques, although developed at least fifteen years ago have failed to win much acclaim in either academia or industry. The reason for this is twofold: their complexity and the quantity of information required before the computation can begin. As normal CPM and PERT techniques are often rejected by many design organisations because of their apparent complexity it is unlikely that these advanced techniques are going to be adopted. Also traditional techniques when performed by computer, requires great precision in reporting and data entry (Gray et al. 1994).
advanced techniques, performed exclusively by computer, require even greater meticulousness. These techniques are also extremely difficult to perform by hand and therefore it is believed that designers and planners are unlikely to put their trust in a technique they fail to understand clearly.

2.5 CONCLUSIONS

This chapter firstly discussed and analysed the general nature of design and secondly critically examined the challenges faced in the management of modern building design projects as a result of: increasingly sophisticated clients; fast-tracking pressures on design; increased building complexity; insufficient information management; and poor or non-existent planning of design work.

The first three categories of problems cannot be directly controlled by the design team, but the design management, and hence the design process, would become more efficient if the last two challenges were addressed and solved. They are inextricably linked and it has been argued that an improvement in design planning will facilitate the management of information. The whole area of design planning was examined in some depth and two reasons have been identified as the major causes of poor building design planning. Firstly the current fragmented nature of design planning does not promote or duly consider cross-discipline co-ordination resulting in unrepresentative and unworkable design programmes. It was suggested that if in some way design planning could be performed and co-ordinated by one party then it was believed that more accurate and workable design programmes would be produced. The second reason identified for poor design building design planning is the lack of suitable tools; traditional tools such as network analysis are unable to represent complex issues unique to design like iteration and choice.

Solving these two problems was seen as a major step towards the more efficient management of the building design process. The work undertaken to achieve these objectives formed the basis of this research programme and is described in greater detail in the following chapter.
Chapter 3
Chapter Three

RESEARCH OBJECTIVES AND METHODOLOGY

3.1 INTRODUCTION

The initial aim of this research was to identify and resolve some of the many conflicts that make building design a problematical and difficult process to manage. The discussion in Chapter 2, on the broad spectrum of building design problems, concluded that design management would be greatly enhanced by planning design more accurately. Efficient planning would focus the problem solving behaviour of designers more effectively than intuitive, unsystematic methods of working and more detailed design programmes would not only promote a more co-ordinated approach to multi-disciplinary design work, but should ease the flow of information across discipline boundaries.

It was established that design is often planned poorly for two reasons. Firstly there is a lack of understanding in each discipline of how their work contributes holistically to the complete building design process. This causes a fragmented approach to planning, with each specialist group drawing up their own programme in isolation with little or no consideration of how their work affects, and is affected by, that of other disciplines. Secondly, the lack of synergy among the design team is further hampered by the range of appropriate planning tools; Gantt charts and network analysis are unable to represent and therefore plan, the many unique attributes of design work, such as iteration or choice, which results in planners having to define tasks in the large rather than in detail.

This project subsequently set out to address and overcome the problems of inadequate design planning. The following sections describe the exact objectives of the research and the methodology developed to realise them.

3.2 RESEARCH OBJECTIVES

3.2.1 The Aim of the Research

The ultimate aim of this research could be stated as:

The development of a more sophisticated approach to the planning and co-ordination of multi-disciplinary building design.
This was approached by considering the project in two distinct phases, the objectives of which were:

Phase I
To gain a greater understanding of the tasks involved, and the information flowing, in a multi-discipline design project so that planning the complete design process can be performed by a single party.

Phase II
To use the knowledge captured in Phase I, to develop a prototype methodology to plan and monitor design work more efficiently.

These two objectives are discussed in more detail in the following sections

3.2.2 Phase I : Understanding the Building Design Process

The Objective
Groak [1993] likened the building process to the concept of Chaos Theory. Chaotic systems are highly sensitive, with seemingly insignificant events giving rise to supposedly unforeseen consequences. This makes understanding the building design process so difficult, often putting its comprehension beyond the realm of the individual. Groak went on to argue that even chaotic systems are deterministic, and it was believed that determining the individual process and information flows, within our system, the design process, would lead to a clearer understanding of this 'Chaos', which in turn would allow the whole design planning operation to be performed by a single party.

The Hypothesis
It was proposed that mapping these features could be achieved most easily by producing graphical models to characterise the building design process. The models would represent the knowledge and 'know how' of designers, and capture it in such a way that it could be assimilated by designers and managers alike. Fully validated models would cover the design process from a multi-disciplinary viewpoint and could be analysed repeatedly, to form the basis of a new design planning methodology.

It was anticipated that as well as forming the basis of a new design planning methodology, the models of the building design process would have other more immediate and realisable benefits:-

(i) Studying the models would undoubtedly lead to a clearer understanding of the complex inter-relationships occurring in the design of a building.

(ii) The model would allow designers to appreciate the factors that should be considered before executing design tasks.
(iii) The model could be used as a way of expressing preferred design practice within an organisation.

(iv) The model would permit a more rigorous understanding of the effects of change, variations and design errors.

(v) If the model covered conceptual and scheme design it could be used to study the client/design team interface. From this it may be possible to suggest a more structured and minimalist approach to brief taking.

3.2.3 Phase II: Producing a Design Planning Methodology

The Objective

The objective of the second phase of the research was to develop a new design planning methodology that could cope with the inherent complexities of building design work. This methodology would integrate the models developed in Phase I so that data on design tasks and their dependencies could be repeatedly extracted from the models to formulate different design programmes for different projects.

The Hypothesis

It was proposed to adapt or develop a technique that would analyse the models developed of the design process and predict the most efficient manner in which to perform cross-disciplinary design.

3.3 RESEARCH METHODOLOGY

The methodology developed to test the hypotheses and realise the two major objectives, discussed in the previous section, is outlined diagrammatically in Figure 3.1. The methodology was developed by splitting the two major objectives into further sub-objectives: stages 1 and 2 to complete the main objective of phase I; and stages 3 to 4 to complete the main objective of phase II.

Stage One

The aim of stage one was to discover the most suitable way to construct models of the building design process. Two different approaches were considered: firstly, existing models of the design process from manufacturing and constructions domains were examined and an assessment of their applicability to represent building design made. The second approach examined the suitability of using graphical modelling techniques, predominately from the electronic data processing industry to produces models of the building design process.
Chapter 3

Narrowing Objectives of Research

A More Efficient Design Process

Improved Building Design Management

The Aim of this Research: Improving Building Design Planning

Phase I Objective: A Greater Understanding of the Design Process

Phase II Objective: Development of Prototype Design Planning Methodology

Figure 3.1: Research Methodology
Chapter 3

Stage Two
The aim of stage two was to produce concepts and philosophies for developing models of the building design process by examining a selection of existing projects. The modelling proposals developed were then combined with the most suitable way of modelling the design process (stage one), to produce paper based models representing the design of typical sections of a modern building.

Stage Three
Stage three took the models of the design process, developed in stage two, and used them as the basis of a prototype design planning methodology to produce effective design programmes. A technique was developed that analysed and manipulated the data stored in the model. This development process was initially conducted on a small case study that allowed the parameters of the design planning methodology to be established without being distracted by detail.

Stage Four
The design planning methodology developed in stage three was tested, refined and partly automated on a larger case study that incorporated many of the co-ordination problems faced when designing a building. The design planning methodology developed was called ADePT (Analytical Design Planning Technique).

Chapters four to six cover stages one and two of the research and chapters seven and eight cover stages three and four.
Chapter 4
Chapter Four

EXISTING MODELS OF THE DESIGN PROCESS

4.1 INTRODUCTION

The research methodology described in the previous chapter outlined the objective of the first phase in the development of a new design planning methodology as the gaining of a clearer understanding of the building design process. This chapter discusses the first stage of work undertaken to achieve this. An examination of relevant literature revealed a vast number of design models, each with their own objectives and idealisations, all of which had the potential to represent the building design process in the manner required for this research. To examine the applicability of adapting these models it was first necessary to identify and define the specific requirements of a building design process model and then use the resulting definition to assess existing design models from both manufacturing and construction domains.

4.2 MODELS AND MODELLING

4.2.1 General Definitions

Models are descriptions of systems [Pritsker 1979]. The descriptions can take one of three forms: physical, mathematical, or graphical, examples of which are shown below:

- **physical models**: Globes, ionic models or scale models;
- **mathematical models**: Differential equations; and
- **graphical models**: Charts, maps, atlases.

Models and modelling are very much in vogue in current research circles and there appears to be as many different definitions of a model as there are models themselves. A selection of the many definitions and purposes of design models are shown below to emphasise the wide variety of thinking on design modelling:

"A model allows questions about a system to be answered accurately."

Walker [1984]

"A model is employed for learning from experience in a more rigorous way."

Marca & McGowan [1988]

"A descriptive model describes the sequence of activities that typically occur in design. A prescriptive model attempts to prescribe a better, more appropriate pattern of activities."

Cross [1989]
Chapter 4

Models are a means of generating information about phenomena; to predict their behaviour."

Rouse (1991)

This lack of consensus made it necessary to state succinctly the aims and objectives of our model of the building design process, allowing a unique definition to be drawn up.

4.2.2 Requirements of a Model to Represent the Building Design Process.

In considering the development of a design process model the following questions were addressed:

(i) What should be represented in the model and to what level of detail?
(ii) What is the model going to be used for?
(iii) Whose work should the model represent?
(iv) What is the exact boundary of the system to be modelled?
(v) In what format should the model be presented?
(vi) What aspects should the model incorporate that were absent in existing planning techniques?

The answers to these questions, outlined below, formed the backbone of the approach developed to produce a model of the building design process. These were the preliminary thoughts for the concepts and philosophies of the model.

What will be represented in the model?

The model, as part of a planning tool, will need to represent the processes of design work designers perform. If the model is to be re-used it must capture these features in a generic way so that the model is not project specific. Another critical property must be the representation of information flow. Information is seen as the fuel of design and it is vital that the model will capture all information flowing within the design process (information in this context is defined as anything required as an input to a task so that it can be processed). An important output from the design process is the design deliverables, i.e. the drawings, the specifications, etc., and any modelling technique must also be able to represent these.

The level of detail to which these three properties must be represented will be very salient. If tasks and information are modelled in too much detail the model would be excessively large, making it cumbersome and complicated to manipulate. If the tasks and processes are modelled too generally the multiple engineering interactions required to complete them are ignored. It is necessary therefore to strike a compromise between the model becoming too large and complicated and being too simplistic and vague.
Chapter 4

What will be the uses of the model?
The model will ultimately form an integral part of a methodology that can be used to plan the design of a building more efficiently than existing techniques. The model will also act as a knowledge base for a design planner and will obviate the need to define the tasks and their dependencies before each project begins.

Who will be represented in the model?
The model shouldn't represent individual design personnel. All that will be necessary is an indication of which discipline will perform each process.

What is the boundary of the system?
The system to be modelled is the process of designing a building as defined in Chapter 2.

What will be the format of the model?
The constructed model will need to be verified and therefore have to reflect the designers' concept of the design process. A graphical technique will be the most suitable modelling technique as it should facilitate model construction and verification. It would be advantageous if the model could be drawn and adapted via a computer based software. It is envisaged that any model should be hierarchical in nature; the design process being represented in different levels of detail dependent on the position in the hierarchy.

Additional aspects required of the model
Existing planning techniques such as network analysis fail to define the design process adequately because they can not represent iterative cycles nor the aspect of choice (refer to Chapter 2). The design model must represent interdependencies between tasks (tasks that need to be solved iteratively) and not purely a one way progression of tasks. If the model is to represent the design of a general building it must allow decisions to be considered. One building may have a structural steel frame another a concrete frame, one may have full air conditioning another may be comfort cooled; decisions such as these must be captured in the model if it is to represent the design of a general building completely.

It is also important the model does not pre-define the order in which tasks should be performed. If it were necessary to do so the whole point of constructing such a model would become irrelevant. Tasks can only be performed when the information they require is available and it should be this that dictates whether a task can proceed.
From the desired properties listed above the definition of a model to represent the building design process in the context of this report should be taken as:

A graphical record that hierarchically represents the building design process in terms of tasks performed and the information both required and produced by the tasks.

4.3 DESIGN MODELS FROM THE MANUFACTURING DOMAIN

The majority of engineering or manufacturing design process models tend to be idealised [Taylor 1993] and it is widely recognised that models fall into one of three categories: descriptive, prescriptive and consensus models, all of which are put forward as rational, systematic frameworks for increasing the effectiveness of design [Cleland & King 1993]. Stauffer [1989] suggests a general methodology (a model) can be applied to all domains of design providing the description is at the most general level of abstraction. The following sections seek to test this hypothesis by discussing whether existing models of the engineering design process, whether prescriptive, descriptive or consensus, are sufficiently adaptable to represent building design in a manner suitable for this research.

4.3.1 Descriptive Models

Cleland and King [1993] consider descriptive models as describing how engineering designers perform the process of design, whilst Whitefield and Warren [1989] believed they should describe the designers mental processes and are usually based on empirical evidence. Luckman's model [1984], shown in Figure 4.1, is typical of many descriptive models describing the processes of analysis, synthesis and evaluation employed by the designer to develop a solution to a problem.

**Analysis**
The collection and classification of all relevant information relating to the problem.

**Synthesis**
The formation of potential solutions to parts of the problem which are feasible when judged against the information contained in the analysis stage.

**Evaluation**
The attempt to judge, by the use of criteria, which of the feasible solutions is most satisfactory in answering the problem.
March [1984] believed not all designers follow the analysis, synthesis and evaluation route to evolve a solution. He argued that a rational design process has three different tasks:-

(i) the creation of a novel composition which is accomplished by productive reasoning;
(ii) the prediction of performance characteristics by deduction; and
(iii) the accumulation of habitual notions and established values by induction.

### 4.3.2 Prescriptive Models

Prescriptive models are essentially prescriptive proposals as to how designers should work more systematically. They usually offer a more algorithmic, systematic procedure to follow and are often regarded as providing a particular design methodology [Cross 1989]. Prescriptive models usually emphasise the sequence of stages that is likely to occur during a project's development [Cross and Roozenberg 1992]. Prescriptive models depict what activities should occur in the design process, rather than give a description of how the activities could be performed.

A simple prescriptive model of the design process, put forward by French [1991], is shown in Figure 4.2. The model consists of activities represented by boxes, and design outputs represented by circles. The model is based on four basic activities which are typical of a conventional engineering design project: analysis, conceptual design, embodiment design, and detail design. The aim of the model is to encourage designers to follow the various stages of the design process and to work in a systematic manner.
4.3.3 Consensus Models

Consensus models of the engineering design process are combinations of prescriptive and descriptive models. Roozenberg and Cross [1991] state these models are formed on two axes. The vertical axis represents the various stages of design (prescriptive models), the horizontal axis represents the problem solving processes employed by designers throughout the various stages of design (descriptive models).

Much work has been performed in this area by the German Engineering Designer's Institute on the VDI model and by Pahl and Beitz [1984]. However, the model most common in design literature is the total design activity model put forward by Pugh [1990] and adopted by the SEED Working Group [1985] as a basis of teaching design. Figure 4.3 shows part of the total design activity model. The core of the model represents the prescribed steps through which design should proceed. The main flow of design work is shown by the heavy black arrows, although the model does permit the flow of feedback
information up the model (iteration). For each stage of the design the methods and procedures employed to complete the design activity are described by the horizontal component of the model. The model also depicts the information required at each stage of the design.

Figure 4.3: Pugh's Total Design Model

4.3.4 Applicability to Represent the Building Design Process
Taylor [1993] points out that no single model has universally been accepted as truly representative of the design process. This is indicative of the complexities of the design process and the nebulous nature of the work being modelled. One reason put forward for this lack of acceptance is that many of the models have been proven to be neither accurate descriptions, nor feasible prescriptions, of the design process. For instance, Hiller et al. [1984] argue strongly that descriptive models are inherently incorrect if applied to the architectural design process. They believe architects develop a solution first, and then subject the solution to analysis and evaluation rather than problem analysis preceding development of a solution synthesis. These comments highlight the practical difficulties of using a model from one engineering field to represent the design process of another.

Descriptive models are of little use for planning engineering design as they describe how activities are performed. If a model is to be used as a planning tool, it must represent each
individual design activity needed to complete the product. Although prescriptive models describe the stages performed within the design process, they are insufficiently detailed to plan a sequence of individual design activities. Hollins et al. [1993] concur with this view adding that present models lack features that can be used for planning design, as they are too vague. It is therefore concluded that engineering design models are not adaptable or sufficiently detailed to represent building design.

While the models discussed above were not deemed appropriate to this project, it is appreciated that many emerging techniques in manufacturing, especially in concurrent engineering, may have the potential to fulfil the requirements discussed in Section 4.2.2. The power of modern computing systems has allowed large Knowledge Based Engineering (KBE) models of both products and processes to be stored, harnessing both the experience and knowledge of practitioners in a consistent and re-usable manner. Fisher (1993) believes that the successful application of KBE models to the design of buildings, will eliminate another distinction between the manufacturing and construction industries promoting an improved cross-fertilisation of ideas and philosophies. Many other authors, such as Hsu (1994), Kysiak (1994) and McCord & Eppinger (1993), describe advanced model based techniques for improving manufacturing processes, that could feasibility be applied to the management of building design in the future.

4.4 DESIGN MODELS FROM THE CONSTRUCTION DOMAIN

There are two distinct types of building design models: models that represent the product (the building itself), and those that represent the process (the work to design a building). Traditionally, building design process models have not been divided along the lines of prescriptive, descriptive or consensus models and have not been examined in this context. However, the building design process models discussed in sections 4.4.2 and 4.4.3 attempt to show how designers could work more systematically and therefore in essence are prescriptive models.

Although this research was concerned with the planning of the process, product models were examined as they have been developed to address the co-ordination and communication problems associated with multi-disciplinary design work [Ahmed et al. 1992]. No single product model is described, instead the underlying philosophies and concepts are discussed to determine whether a product model could be constructed to develop an improved understanding of building design work.
4.4.1 Building Product Models

Building product models have been developed to increase the automation of building design and construction. The key to their success is seen as the integration of the information processing required by the various disciplines, at the various stages of design [Roseman et al. 1993]. A building product model provides an information structure, in which data about a building can be stored in an integrated way [Van Nederveen 1993]. It should contain all data needed to construct and maintain a specific building and should describe parameters such as shape, location, relationships, physical properties, materials used and cost [Bjork 1989]. Building product models are structured in such a way as to facilitate the storage, retrieval and revision construction data by all the different design disciplines [Evt et al. 1990]. These concepts underpin the majority of building product models developed over the last decade, such as the RATAS model [Bjork 1992].

A building product model considers a building as a hierarchy of compositions [Eastman 1992], and represents this by decomposing the product, rather than the process, into systems. Examples of systems within a building include the heating system, the communication system and the structural frame. Each system is decomposed into the necessary levels of abstraction to adequately describe it. A heating system for a building could be decomposed into the heating required by each floor, which in turn, could be decomposed into the heating requirements for each room on that floor. Some of the factors that affect the heating of that room are its size, height, the internal and external temperatures and the 'U' value of the walls. These, the lowest levels of the decomposition, are referred to as the attributes of the system.

The difficulty in producing a model comes when different objects within the model are present in two or more systems. A wall in a room, for instance, may form part of the load bearing system, the partitioning system and the heating system. As the different members within the design team need to co-operate in the design process, it is essential that when an object is changed in one part of the model it is also updated and communicated to all. Modelling techniques such as the NIAM modelling language and object oriented methodologies have been used to cope with these complexities.

At present there is no universally accepted building product model, nor method of producing one. However, a large international effort is underway to produce a product model standard STEP (Standard for the Exchange of Product Model Data) which is a common exchange format for all the relevant engineering data relating to a product. The principal goal of the STEP initiative is the creation of a standard that enables the capture of information for a computerised product model in a neutral form [Wix 1989]. This will allow many existing CAD and CAE packages to communicate with each other, or be used
directly as a database to access information, without the loss of completeness or integrity of data. Work has taken place on STEP throughout the last decade and it is not expected to produce a comprehensive standard available to the industry until 1996.

A building product model can give an in-depth understanding of a building and its parts, their properties and their relationships. This insight will undoubtedly lead to more efficient designs. However, it will be pointed out later in this chapter that the requirements of the proposed model mean the activities necessary to design a building need to be described, rather than the outcome of the activities, i.e. the product.

As processes are not represented, a building product model is not suitable for the purposes of this research. However, it will be seen later in Chapter six, that many of the concepts for decomposing a building into systems are relevant for decomposing the design process.

### 4.4.2 Wix’s Model for Representing the Design Process

Flow charting has been a widely used technique to produce process models of the construction work (Addis 1990); these tend to produce prescriptive models. One process model based on flow charting that stands out is Wix’s model which attempts to model the flow of work in mechanical services design [Wix 1986]. It is the only attempt known to the author which models part of the building design process in detail. The model produced by Wix uses a technique similar to that of flow charting but allows more detailed and flexible modelling of data and its flow. A typical section of Wix’s model is shown in Figure 4.4.

The building services process has been divided into well-defined sections of work, each of which are shown in their own flow charts within the model. These separate charts are then linked by the information that flows between them. The notation for Wix’s flow charts is shown in Figure 4.5. It can be seen that there is notation for three different types of data: fixed, project and transient. Wix defines and distinguishes between these types of data in the following manner.

**Fixed Data**

Fixed data are properties or characteristics of items of data independent of an individual project. This would include the physical properties of materials, details of manufactured items or regulations laid down in such documents as British Standards. This type of data doesn’t change from one project to another.
Project Data
Project data are data that specifically relate to a project under consideration and that ultimately need transferring among other disciplines within the building design team. This data includes, for instance, dimensional data stored on drawings or in specifications, financial data, resource data and project performance data.

Transient Data
Transient data are data that are produced and then solely utilised by another process within a single discipline, i.e. it does not pass to other disciplines. For example, the process of calculating amount of heat produced by people, equipment and lighting for each room.
would result in transient data. This data would need to be included in the calculations for total heat gains but would not be required by another discipline.

![Diagram](image.png)

**Figure 4.5 : Notation for Wix's Model of Mechanical Services Design**

These three types of data are shown in the flow diagrams along with the processes themselves and any communication required in five separate columns. A typical section of the model, shown in Figure 4.4, shows the processes involved in the calculations of heat losses. Each flow chart has an individual reference number and the reference numbers in the transient data column are the processes that supply information to, or require information from, the heat loss section of the model.

Wix's model is a detailed account of mechanical services design. It describes both processes and the information flowing between them. However, the technique has several serious flaws. Primarily it is based on flow charting methodologies and incorporates many of their failings, such as the pre-defining and sequential task ordering. Although the technique can adequately represent the flow of single discipline information (transient data), it would be difficult to show succinctly the originator of cross discipline information. It is also difficult to see to see how the aspect of choice could be incorporated into this model.

**4.4.3 The RIBA Plan of Work**

The RIBA Plan of Work, originally published in 1964, is a prescriptive model procedure for the methodical working of a multi-disciplinary team on a typical building project valued up to £300,000 (1973 price) [RIBA 1973]. The model is a two dimensional representation of the building design process and is described from several viewpoints. The stage or phase of the project is detailed on the vertical axis and the project's participants are detailed on the horizontal axis.
Figure 4.6 : The RIBA Plan of Work Structure

An outline view of the model is shown in Figure 4.6. There are twelve stages shown in the Plan of Work, represents the logical sequence of design work; passing from stage A: Inception through to stage M: Feedback. For each stage there are eight design functions performed by the various participants involved in the project. It is assumed the architect is responsible for leading the client and the design team, and therefore, has two functions within the Plan of Work, a management function and a design function. Within the body of the Plan of Work the actions and duties of each participant are listed for each stage of the work. The typical functions described include:

- design studies/work to be performed;
- proposals and options to be considered;
- decisions and actions to be taken;
- discussions and meetings to be accomplished; and
- information to be elicited and provided.
These duties represent an outline method of working on a general project and need adapting to the specific needs of individual projects.

The RIBA Plan of Work is a document accepted throughout the construction industry. It forms the basis of the terms of engagement between parties, the fee scales for an architect's professional services [RIBA 1982] and the specific stages are often quoted in contract documents as the extent to which the design should be developed.

Although the Plan of Work attempts to provide a systematic framework within which design can be managed, it should be considered as more of a checklist, describing what should be done at which stage of the design life cycle. It is not detailed enough to schedule individual activities nor define task durations to produce workable design programmes. The Plan of Work also describes the complete project life cycle from Inception through to Hand Over and Feedback, making parts of the model redundant for the purpose of planning design.

The design functions within the Plan actually describe many management functions, such as overseeing design work and task co-ordination. Tasks such as these would not appear in a design programme as they would be deemed to be ongoing throughout the project. Again these parts of the model would be redundant.

In all the RIBA Plan of Work is an accurate representation of the design process for the level of detail shown. The stages listed form a well recognised framework from which a more detailed model could be developed. However, the Plan of Work, as it stands, is too generic to plan the individual participants' work on a day to day basis.

4.5 CONCLUSIONS

The research methodology discussed in Chapter 3 required existing design process models, from both manufacturing and construction domains, to be examined in the belief they could be adapted to represent the building design process.

As each design model has different priorities and idealisations and because a universal definition for the term 'model' did not exist, it was necessary to define the required properties of our building design process model. The various objectives of the proposed model were examined and it was found that the most important requirement was the necessity to capture the generic nature of building design, in such a way that it could be used to predict the work and information flows in future design processes. From the aims
of the model it was possible to distil a definition of the term model in the context of this research. This is given below:-

A graphical record that hierarchically represents the building design process in terms of tasks performed, and the information both required and produced by the tasks.

Initially design process models from the manufacturing domain were examined and their suitability assessed. Of the three types of model studied: descriptive, prescriptive, and consensus, it was found that prescriptive models, in principle, could be used to represent building design. However, all the prescriptive models were insufficiently detailed to be used as the basis of a technique to plan design work.

Two types of building design models were examined: product models representing the building itself and prescriptive process models representing suggested systematic ways of working. Product models, although developed to aid understanding, represented building components rather than design processes and were therefore inappropriate for the purposes of this research. Two different process models were examined: Wix's model of building services design and the RIBA Plan of Work. Wix's model represented design in sufficient detail and captured individual design tasks and the information flowing between them. This technique could feasibly be adapted to generically represent building design. However, Wix's approach used flow charting methodology which requires the predetermined task order. This conflicted with the prerequisite that no control should be imparted on the order of tasks within the model and therefore this approach was adjudged to be invalid for the purposes of this research.

The RIBA Plan of Work has been frequently used to provide a systematic framework within which design could be managed and has gained wide recognition as a benchmark within the building industry. However, the plan is only a checklist describing what should be done and at what stage of the design life cycle, rather than a detailed model of individual design tasks. Therefore, the Plan of Work was found not to be detailed enough to schedule individual activities nor define task durations to produce workable design programmes.

It is therefore concluded that existing design process models examined from both the manufacturing or building domains were neither adaptable, nor sufficiently detailed, to represent building design in a manner required to gain a clearer understanding of the whole design process.
Chapter 5
Chapter Five

TECHNIQUES FOR MODELLING THE DESIGN PROCESS

5.1 INTRODUCTION

The objective of the first phase of the research, as set out in Chapter 3, was to gain a clearer understanding of multi-disciplinary building design; it was hypothesised that this could be achieved by developing and studying models of the design process. The previous chapter examined the possibility of adapting existing models and concluded that this approach was not feasible. This chapter discusses another approach and assesses whether diagramming techniques, commonly used in the Information Technology industry, can be used to develop models of the building design process from first principles.

5.2 STRUCTURED METHODOLOGIES & TECHNIQUES

In the electronic data processing (EDP) industry structured techniques, known as methodologies, are used to ease the development of systems software. A methodology is used to build a model of the observed phenomena (Checkland & Scholes 1990), and is a collection of procedures and techniques, called methods, that perform particular tasks within the software development cycle. An integrated set of these methods, used in a constant manner to produce an overview of the system, is called a methodology. Structured techniques evolved in the early 1970's and initially concentrated on structuring programming languages. By the late 1970's a set of structured techniques had been developed that encompassed the whole software development cycle (Martin & McClure 1985). Under the genre of 'structured design' the design of the software itself became more rigorous, embracing the concepts of modularisation and standardisation. This in turn led to the development of 'structured analysis', a more systematic approach to analysing and formulating the problem, rather than concentrating solely on the solution.

The many methodologies developed over the last two decades for either structured analysis, or structured design, utilise similar tools which are based heavily upon charting or diagramming techniques, supported by appropriate narrative. The intrinsic differences between the methodologies are the subtle differences in the tools, their combination and their use (Yourdon 1989).

Flow charting techniques, although commonly used in the EDP industry, have not been considered in the following section because many of the models discounted in the previous
chapter, such as Wix's model or French's model, were developed using this technique. Only those techniques deemed most appropriate to produce models of the design phase are discussed in the next sections.

5.2.1 Entity Relationship Diagrams
Entity relationship diagrams (ERDs) are used in many structured design methodologies. Although there are many hybrids and aliases for ERDs, such as object relationship diagrams, object association diagrams, information modelling, or data modelling, they all share the basic philosophy of describing the relationships between data within a single system (Bingham & Davies 1992). Martin & McClure (1985) define an ERD as a technique to identify the entities involved in the running of an enterprise, and determining the relationships between them; an entity being either something real or abstract. All ERDs use the basic notation of a box used to represent an entity and a series of 'crowsfeet' used to represent all possible relationships between them. This notation and the various relationships are shown and explained in Figure 5.1.

An example ERD can be seen in Figure 5.2 showing the relationships between the entities piles, pile caps and columns. From the diagram the following statements can be deduced:
- a pile cap is supported by one or many piles;
- piles can be tied by one pile cap only;
- a pile cap supports zero or one column; and
- columns can sit on one pile cap only.

It can be seen that ERDs can succinctly show the relationships between parts of a system, in the example shown the system being a building. The major drawback of ERDs is they model entities rather than processes, making the representation of information flow...
impossible. This obviously precludes them from being capable of modelling either interdependent tasks or design options, making ERDs unsuitable to model the building design process in the format required. However, it is plausible that ERDs could be used to show the relationships between design documents and the information stored in them. For example Figure 5.3 shows that a pile cap drawing would store information on one or more set of pile cap dimensions and foundation levels.

![Figure 5.3: A Typical Entity Relationship Diagram](image)

### 5.2.2 Data Flow Diagrams

Data flow diagrams (DFDs) are a hierarchical graphical technique used in many structured analysis methodologies (Gane & Sarson 1978, Ward & Mellor 1986, DeMarco 1979). The technique in each methodology is slightly different in terms of notation and emphasis but are all based on the central philosophy of representing a system by a network of activities that accept and produce data (Ward & Mellor 1986). A DFD is made up of four basic elements: data flows, processes, files and data sources or sinks; denoted by the symbols shown in Table 5.1.

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>SYMBOL</th>
<th>SOFTWARE DEVELOPMENT DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA OR INFORMATION FLOW</td>
<td></td>
<td>A connection between processes etc., representing an input and/or an output.</td>
</tr>
<tr>
<td>PROCESS</td>
<td><img src="image" alt="process" /></td>
<td>Individual functions that a system carries out. They transform an input into an output.</td>
</tr>
<tr>
<td>DATA STORE OR FILE</td>
<td><img src="image" alt="store" /></td>
<td>A collection of information that must be remembered for a period of time.</td>
</tr>
<tr>
<td>SOURCE/SINK</td>
<td><img src="image" alt="sink" /></td>
<td>External entities with which the system communicates.</td>
</tr>
</tbody>
</table>

Table 5.1: DFD Element Descriptions
DeMarco (1979) defines these elements in the following way:

- a *data flow* is a pipeline through which packets of information of known composition can flow;
- a *process* is a transformation of incoming data flow(s) into outgoing data flow(s);
- a *file*, also known as a *store*, is a temporary repository of data; and
- a *source or sink* is a person or organisation lying outside the context of the system.

These elements are most easily explained by way of a small example. Figure 5.4 shows three basic tasks involved in the production of a set of storm drainage calculations. To allow each of the three processes to proceed, certain pieces of information are required. These are shown as arrows (data flows) entering each process. Also shown is the originator of these pieces of information, i.e. a file, a source or from another process. The files shown in this example are all drawings and the source shown is a British Standard.

![Figure 5.4: A Typical Data Flow Diagram](image)

Data flow diagrams are capable of modelling processes and the information flowing between them. This is an essential prerequisite of a suitable technique to model the building design process. DFDs view a system from an information point of view, mapping information flows, their transformation and their co-ordination. As DFDs do not impose or record any managerial control on the timing of the flow of information, it is not necessary to pre-define the order in which tasks should be performed. This aspect is often missed as DFDs appear, at a superficial level, deceptively easy to use (Fisher & Lin 1992). They identify, in a graphical form, the information and its sources, necessary to allow a process to proceed. Information flows dictate the order in which individual design tasks may proceed and not the other way around. As stated earlier this is an essential requirement of the model.
Chapter 5

The 'data view' of DFDs allows tasks that form part of an iterative cycle to be modelled and understood. This is essential, as design is an iterative process and can rarely, if ever, be performed as a series of sequential processes, especially in a multi-discipline project. This can clearly be seen in Figure 5.4. Tasks 2 and 3 are interdependent and therefore require solving iteratively; the size of a pipe is dependent on the fall of the pipe which in turn can only be calculated once the size of the pipe is known.

Although not shown in the example, DFDs can be layered or 'partitioned' by breaking each task down into further sub-tasks. Each DFD is restricted to a page of A4, producing a compact, hierarchical, multi-dimensional model. The top of the model will represent the design process in the most general terms and descending the model will increase the level of detail represented. This top down analysis allows the higher regions of the model to be read to obtain an overview of the system, and if more detail is required the lower levels can be studied as and where necessary (DeMarco 1979). However, for large systems collecting and analysing data, and drawing and validating models can be a time consuming process.

5.2.3 Structure Charts

Structure charts are a hierarchical diagramming technique that uses functional decomposition to examine a system and construct a model that goes from the most general representation at the top to the more specific at the bottom (Jones 1989). Structure charts were initially developed to describe the overall architecture of a piece of software by showing the program modules and their inter-relationships (Martin & McClure 1985). The term structure chart is taken here to include all sideways and vertical hierarchical diagrams as the underlying philosophies are all very similar; these include HIPO charts, Warnier-Orr diagrams, Jackson diagrams and action diagrams. The slight differences of emphasis in these types of structure charts are discussed briefly later in this section.

The basic element of a structure chart is a module, which historically represented a section of code in a computer program although in the context of this work it could represent any process or task. Connecting these modules in a hierarchical fashion are arrows representing their inter-relationships. The system being modelled is represented as single block at the top of the model, which can be decomposed into any number of sub-modules, which in turn, can be decomposed as many times as necessary. The flow of data transferring from one block to another is represented by flags that are placed along side the arrows. There are two types of flags: those representing the flow of general information and those representing the flow of control information. These are shown and explained in Figure 5.5.
Chapter 5

Module: A module representing a section of work

Data Couple: Information used to direct the system

Control Couple: General information used in the system

Figure 5.5: Structure Chart Notation

An example structure chart can be seen in Figure 5.6. It shows the relationships between the various sections of work that contribute to the design of a storm drainage system. This work is made up of three sub-sections: calculations, drawings and specifications. The flags on the arrows describe the information required and outputted from each section of work. The diagram should not be read in any particular direction as no task order is inferred.

Structure charts can model, in a hierarchical manner, both tasks and the information flowing between them. Most types of structure chart, for example the Yourdon and Constantine (1979) variety, have the advantage of not imposing a sequence or order upon the tasks shown in the model. This in theory means that representing iteration is possible. However, for large systems, structure charts are not sensitive enough to track the large quantities of information flowing between tasks and therefore are not sophisticated enough to model the vast numbers of interactions known to occur in the design of a building.

Figure 5.6: A Typical Structure Chart

Although not capable of modelling building design, a structure chart may be a useful technique to represent the main hierarchy of design tasks within the building design process.
Chapter 5

5.3 OTHER STRUCTURED MODELLING TECHNIQUES

Two other graphical techniques are examined in the following sections which are not used as a structured methodology tool in the electronic data processing industry. The two tools examined are role activity diagrams (RADs) and the structured analysis and design technique (SADT), frequently referred to as IDEF drawings.

5.3.1 Role Activity Diagrams

Role activity diagrams (RADs) are a graphical tool used in process modelling for managing the development of organisational processes in a planned way (White 1992). RADs capture the essentials of a process in terms of roles performed (usually by a person), the actions undertaken by that role and the interactions between roles (Wharton 1992). The notation for RADs is shown in Figure 5.7 and are best explained by way of an example as shown in Figure 5.8.

![Role Activity Diagram (RAD) Notation](image)

This RAD shows the participants, and their roles and duties, in a project considered as part of a pilot study (Section 6.2). The parties involved were engaged in the collection of the client brief. The complicated nature of the project meant there were six satellite design teams each responsible for their own projects but co-ordinated by a core design team. This RAD shows how one satellite team collects and analyses the client information and then distributes it to the core design team via the administration department. They, in turn, pass on any information relevant to the other satellite design teams. Each group is clearly
Chapter 5

represented by a shaded column, and the tasks and duties of each role shown sequentially
with the initial task shown at the top of the page.

Figure 5.8: A RAD Depicting the Collection and Analysis of a Client Brief.
Chapter 5

RADs are concise and easy to read and understand. Their main advantage is that they show who is responsible for what task and with whom they interact to perform it. However, because RADs are 'role-centric' they do not model the movement of information in any great detail and it is difficult to model groups of iterative tasks. Again the major drawback with this diagramming technique is a need to understand the order in which tasks should be performed before the model can be constructed. A practical reason for not using RADs to form the model is that lack of commercially available computer tools to ease the drawing of diagrams.

5.3.2 Structured Analysis and Design Technique

SADT was developed by Douglas Ross nearly 30 years ago as a graphical technique to aid systems description (Ross 1977) and is very similar to DFD techniques. Since that time the method has been standardised by the US Department of defence and renamed IDEF or IDEF0, although SADT will be the terminology adopted in this thesis. Unlike other structured analysis techniques, which have their origins in software design, SADT was originally developed to describe a system from any environment (Marca & McGowan 1988). The SADT or IDEF technique appears to be gaining popularity with researchers producing process models because the diagrams can be linked directly to the STEP initiative via the programming language EXPRESS.

A SADT diagram contains boxes and arrows. The boxes represent activities and the arrows the interfaces between them. The arrows can represent anything, from data to machines, from reports to raw materials. Unlike all other graphic based diagramming techniques each side of the activity box has a special meaning. This is shown in Figure 5.9. The left side of the box is for input arrows, the right for output arrows, the top for control arrows and the bottom for mechanism arrows. Input arrows are transformed into output arrows, control arrows describe information that usually constrains or dictates what activities can do and mechanism arrows describe how the activity is accomplished.

![SADT Notation](image)

Figure 5.9: SADT Notation
A single SADT diagram contains a maximum of six boxes keeping the diagrams readable and succinct. A model is constructed by functionally decomposing systems and producing diagrams from the most general to the most specific. Within each diagram, each box or task is placed according to their relative importance to each other, the most dominant box being placed in the top left hand corner and the remainder placed below and towards the bottom right hand corner. The most dominant task is assumed to be the first task to be performed and the subsequent tasks performed in the order of the tasks in the diagonal. These subtle differences mean that SADT diagrams are not flow charts, nor data flow diagrams; they are constraint diagrams that describe both input and output transformations and the constraint rules imposed upon them (Marca & McGowan 1988).

Figure 5.10: A Typical SADT Example

Figure 5.10 shows an SADT diagram for the storm drainage scheme design. The three processes are placed in the order that they will occur, with the calculations preceding the drawings which in turn precede the costings. If the calculation function is considered then the following are explicit in the diagram:

- the inputs into the calculation function are outputs of other functions in a completed model;
- the transformation of the inputs is controlled by Building Regulations, British Standards and the capacity of the existing drainage system; and
• the whole process can be calculated by a computer software package that is represented by the mechanism arrow.

SADT diagrams can represent many of the effects required of a suitable modelling technique for design. Tasks or activities can be represented hierarchically, as can the flow of information. Each diagram is restricted to the number of processes it can describe, making each diagram easy to construct and read. Although SADT has the facility to distinguish between the four types of information, a model of the building design process would only require an appreciation of the input and output of information from processes, making the control and mechanism influences superfluous to requirements. A control arrow in the context of a building design process model would represent the restrictions on the design output from regulations imposed by the Codes of Practice or British Standards. It will be argued in the next chapter (Section 6.2.4) that information of this sort does not influence the co-ordination of design tasks because it is freely available and therefore obviating need for it to be modelled. A mechanism arrow describes how an activity is accomplished and it is considered that these arrows do not represent true information flows but restraints on design tasks; they indicate how tasks should be performed rather than what information is necessary to allow a task to proceed.

While it could be argued that both these types of arrow could be ignored in a building design process model a more critical flaw in the technique is that although the ultimate task order within the completed model is dictated by the availability of information, it is necessary to pre-define the task order on each diagram, depending on their relative importance. The requirements of the building design process model, stated in Section 4.2.2, mean that the pre-ordering of tasks precludes SADT from being an ideal modelling tool. On a more pragmatic level there does not appear to be an extensive range of computer packages that facilitate the drawing and checking of these drawings.

5.4 CONCLUSIONS ON THE MOST SUITABLE TECHNIQUE

The five modelling techniques examined in detail in the previous sections were: Entity relationship diagrams; Structure charts; Data flow diagrams; Role activity diagrams; and Structured analysis and design technique, or IDEF, diagrams. Each technique was assessed for its suitability to formulate a model of the building design process with the necessary characteristics (described in the previous chapter). Although each technique had advantages and disadvantages, the technique most suited for the purposes of this research were data flow diagrams. DFDs were the most flexible and appropriate technique because, as Fisher (1990) neatly sums up, they have the following properties:
• they are graphical;
• they can be partitioned;
• they are multi-dimensional;
• they emphasise the flow of data rather than control; and
• they represent a situation from the viewpoint of the data rather from the viewpoint of a person or an organisation.

DFDs have also been successfully used by Fisher & Lin (1992) to model the information flow in a contractor's organisation and by Gharib (1991) to model information flow in a design and build organisation. Their work has concentrated on the complete construction cycle and only represents design work in the most general of terms. However, their work does prove that DFDs can be used successfully to model aspects of the construction industry.

The decision to choose DFDs to model building design was further substantiated towards the end of the research period by Pollard and Plume (1993a & 1993b). They have used design decision diagrams (a pseudonym for data flow diagrams) to model the architectural design process. Their approach is slightly different in that they have attempted to represent all the important design decisions taken throughout the life cycle of design rather than the tasks performed during design. However, the predicted advantages of the models are similar. Pollard and Plume believe their models could be used in any of the following ways:-

(i) As a communication tool. This will help the design team and client communicate and allow their discussions to be conducted over a negotiating framework.
(ii) As a framework for the production of the building specification, the drawings, the detailing and the calculations.
(iii) As a starting point for brief taking. The majority of information inputs come from the client and these are captured within the model.

Data flow diagrams will form the basis of the design process model. They are discussed in the context of this research in more detail in the following section. The following chapter then addresses how DFDs are used to formulate models of the building design process.

5.5 DATA FLOW DIAGRAMS IN DETAIL

5.5.1 Application of DFDs to the Building Design Model
The four basic building blocks of a data flow diagram and the corresponding property of the building design process they represent are explained in Table 5.2. These will be the
only symbols used in the model and the diagramming technique followed will be that of 'pure DFDs' as described by DeMarco (1979).

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>SYMBOL</th>
<th>DESIGN PROCESS MODEL DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA OR INFORMATION FLOW</td>
<td></td>
<td>Design information flow.</td>
</tr>
<tr>
<td>PROCESS</td>
<td></td>
<td>Individual design tasks e.g. calculation drawing, specifying.</td>
</tr>
<tr>
<td>DATA STORE OR FILE</td>
<td></td>
<td>Drawings, sketches, calculation files, reports, documents, specifications, computer files etc.</td>
</tr>
<tr>
<td>SOURCE/SINK</td>
<td></td>
<td>Any external data source e.g. Client, Local Authority.</td>
</tr>
</tbody>
</table>

Table 5.2: DFD Element Descriptions for the Design Process Model

Various other DFD techniques, i.e. Ward and Mellor (1986) or Gane and Sarson (1978), use other symbols, such as control processes or control flows, to model real-time systems. A control transformation maps the event flows into and out of the process, an example of which could be a pull cord for turning a lamp on or off. When the cord is pulled, the lamp may go on or off depending on what happened before, as shown in Figure 5.11. These types of processes and flows are not deemed necessary to model the design process.

![Figure 5.11: A Control Data Flow Process](image)

When a system is too large for its DFD to be shown on a single page, as will be the case with the building design process, the system has to be partitioned into sub-systems. If the
sub-systems are still too large they are further sub-divided, and so on. To ensure consistency between systems and sub-systems the DFDs have to be balanced to guarantee that the flow leaving or entering the part of the system is exactly the same as the flow leaving or entering the sub-system. The concept of balancing will be explained by enlarging upon the example DFD shown in Figure 5.4.

Every set of DFDs (we shall term this the DFD model) is made up of a top, a bottom and a middle. At the top of the model is a Context Diagram. This is a single representation of the whole process being modelled and contains a single process named to reflect the domain of the study. The DFD in Figure 5.12 is the Context Diagram and represents the whole process of storm drainage calculations. It can be seen that any number of flows can enter or leave this single process as long as they flow into, or out of, a file or a source/sink. The process in the Context Diagram, according to convention, is always numbered with a zero and is the parent diagram of the first level breakdown diagram, which in turn is a parent to its child diagrams, the second level breakdown diagrams. The child of the Context Diagram is shown in Figure 5.13 and is similar to the DFD shown in Figure 5.4. Each process within this diagram is also numbered to define uniquely each process in the model.

The data flows existing storm drainage and storm drainage calculations entering and leaving the process on the Context Diagram are also shown on the child diagram (Figure 5.13). This means the two diagrams are balanced. The balancing rule ensures that the data flows entering or leaving a process on a parent diagram are identical to those crossing the context boundary of the child. Decomposing the system further produces Figure 5.14, which is the child diagram of process number 2, calculating pipe sizes. Each process is now numbered by the addition of an extra digit to its parent number. This has the duel effect of uniquely defining the process and identifying its parent. Again it can be seen that the net input and outputs of the parent diagram are shown on the child diagram, ensuring the two are balanced. These three DFDs completely describe the system and when correctly balanced constitute a set of levelled data flow diagrams.
At the bottom of every diagram are processes that are not decomposed further. These processes are called functional primitive tasks (FPTs). The decision not to decompose these tasks further is one the modeller must address and is dependent on the ultimate function of the model. Functional primitive tasks are not restricted to the lowest level diagrams and in all the following examples they will be denoted by circles with a slightly thicker line.
When DFDs are traditionally used in software development it is customary to describe each FPT in terms of what it does and how it does it. These descriptions are called process specifications and are used to make the task of the programmer simpler. Process specifications, or mini specs as they are commonly known, are unnecessary for the purposes of this work and will not be drawn up.
Chapter 6
Chapter Six

THE DESIGN PROCESS MODEL

6.1 INTRODUCTION

The previous two chapters have examined the most suitable method of developing models of the building design process. Data flow diagrams were found to be the most suitable technique for producing graphical models of the information flowing between tasks and disciplines. This chapter discusses the practical application of data flow diagrams in drawing up a generic Design Process Model to represent the building design process. A pilot study, performed to determine what aspects of the design process should be covered, was performed prior to the construction and validation of sections of a Design Process Model.

6.2 PILOT STUDY

6.2.1 Model Formation

Designing a building is a complex process, requiring the co-ordination of many hundreds of personnel and a multitude of decisions to ensure a smooth progression from establishing the client's needs, to handing over the completed building. This makes any attempt to model the process a difficult and time consuming task. The aim of the pilot study was to deduce what aspects of the design process should be represented in the Design Process Model, what could be ignored, and to suggest the most suitable approach to modelling the process itself. The following issues were addressed during the study:

(i) Which stages of the design process could be planned most effectively?
(ii) Which stages of the design process could gain the most benefit from improved planning?
(iii) Which aspects of the design process do not need to be modelled?
(iv) How do differing procurement routes affect the design process; would it be possible, or necessary, to incorporate these in the Design Process Model?
(v) What would be the most effective way of representing the different mechanisms for transferring information between all parties in the design process?

6.2.2 Pilot Study Projects

Three projects, from a single multi-disciplinary design organisation, were examined in the pilot study to gain an appreciation of the differences and similarities in designing different buildings. The three projects were chosen to ensure they covered:
Chapter 6

- different stages of the design process;
- different procurement methods; and
- different types of building.

The three projects studied in the pilot study are contrasted in Table 6.1.

<table>
<thead>
<tr>
<th>Project</th>
<th>Building Type</th>
<th>Design Phase</th>
<th>Method of Study</th>
<th>Procurement Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Pharmaceutical</td>
<td>Conceptual</td>
<td>Shadowing</td>
<td>Design &amp; Manage</td>
</tr>
<tr>
<td>B</td>
<td>Corporate Offices</td>
<td>Scheme/Detail</td>
<td>Historical</td>
<td>Design Only</td>
</tr>
<tr>
<td>C</td>
<td>High Bay Warehouse</td>
<td>Detail</td>
<td>Historical</td>
<td>Lump Sum Design &amp; Build</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison of the Projects Studied in the Pilot Study.

Originally each project was studied using the RIBA Plan of Work (RIBA 1973) as a benchmark. For reasons of simplicity and the desire to mirror the terminology used on the pilot study projects, the stages outlined in the Plan of Work were condensed into three phases: conceptual design, scheme design and detail design. These terms, defined in relation to the Plan of Work and shown in Figure 6.1, are used throughout the remainder of the thesis.

<table>
<thead>
<tr>
<th>RIBA Plan of Work Stage</th>
<th>Terminology Considered for this Research</th>
<th>Also Known As</th>
</tr>
</thead>
<tbody>
<tr>
<td>A) Inception</td>
<td>CONCEPTUAL DESIGN</td>
<td>Feasibility Design</td>
</tr>
<tr>
<td>B) Feasibility</td>
<td>CONCEPTUAL DESIGN</td>
<td>Concept Design</td>
</tr>
<tr>
<td>C) Outline Proposals</td>
<td>CONCEPTUAL DESIGN</td>
<td>Client Definition</td>
</tr>
<tr>
<td>D) Scheme Design</td>
<td>SCHEME DESIGN</td>
<td>Estimating Design</td>
</tr>
<tr>
<td>E) Detail Design</td>
<td>SCHEME DESIGN</td>
<td>Pre-tender Design</td>
</tr>
<tr>
<td>F) Production Information</td>
<td>DETAIL DESIGN</td>
<td>Enquiry Design</td>
</tr>
<tr>
<td>G) Bill of Quantities</td>
<td>Production Design</td>
<td>Front End Design</td>
</tr>
<tr>
<td>H) Tender Action</td>
<td>Production Design</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.1: Simplification of the RIBA Plan of Work Terminology
Chapter 6

The historical analysis of the completed projects was achieved by consulting minutes of meetings, drawings, project documents and by conducting informal interviews. This analysis was backed up a personal involvement in two of the projects. The analysis of the ongoing project was conducted by attending meetings and shadowing the design team over the duration of the conceptual design study.

6.2.3 Pilot Study Observations

A number of observations were made during the pilot study, some relevant to specific projects, some relevant to all three. The features common to all three projects, and therefore the most pertinent to the development of the modelling proposals, are discussed below:

(i) The type and nature of work carried out by designers (design work as defined in Chapter 2) was consistent among the three projects. Design work was also independent of the type of building, the procurement route and the influence of the management. A clear example of this was the design work involved in the sizing of a rainwater down pipe: the only pieces of information required to perform this task were the area of roof drained and the predicted design storm; no other information or factors affected this process. Many design tasks common to all three projects had identical information dependencies and all that distinguished these dependencies from one project to the next was the exact attributes of the information input and output from each task.

(ii) Individual design tasks in each of the projects were often repeated. There were two reasons for this: some tasks were reworked to validate, clarify and improve design solutions (iteration), others were reworked because a design error had been discovered or the proposed solution was inappropriate (repetition).

(iii) It was impossible to track the flow of verbal and informal information transfers.

(iv) It was believed that the many requests for information between designers occurred because there was a lack of forethought about future information requirements. It was discovered that most information transfers between disciplines were not sent pre-emptively, probably because designers rarely have a clear understanding of the information requirements of other disciplines.

(v) The outcome of many management tasks was dependent on the type of procurement route chosen, e.g. the determination of resources required to complete a work package was dependent on the tender dates of each package, which in turn, was a function of the procurement route chosen.

(vi) Clients took varying degrees of involvement in the project and the way in which they organised and managed their own team affected the design process.
(vii) Detail design work was much more predictable than either scheme or conceptual design. The level of detail and accuracy required in the construction documents meant that detail design was also the most time consuming process.

6.2.4 Design Process Model Proposals

The following requirements of the Design Process Model were determined from the observations discussed in the previous section.

Design Phase

The Design Process Model was chosen to represent detail design work because:

(i) Detail design is much more predictable than either scheme or conceptual design which are very nebulous processes and therefore the work required to complete this phase can be anticipated fairly accurately. Detail design tasks tend to be fairly discrete entities; in contrast, it is more difficult to decompose conceptual or scheme design into individual tasks as it is often hard to deduce when one task finishes and another task begins.

(ii) Detail design fees are much greater than for the other phases and it is common sense to concentrate on the areas of greatest spend.

(iii) More designers are involved in detail design, requiring greater communication and co-ordination to ensure design has a chance of proceeding smoothly.

(iv) As design schedules are compressed it is important that detail design is planned accurately to ensure a smooth transition into the construction phase.

Modelling scheme design is currently being undertaken by Baldwin et al (1994), under a parallel research initiative. A section of this work is addressing whether the model of the detail design process, developed as part of this project, can be adapted to represent scheme design as it has been hypothesised that many of the tasks, and information dependencies, are inherently the same in each phase. A model constructed from data flow diagrams is chiefly concerned with the mapping of information flows not processes. They do not represent how a process is performed, only what is needed to perform the task and what is produced. What makes a scheme design task different from its counterpart in the detail design phase is the accuracy and the level of detail of the information processed (Austin et al. 1993).

This point can be supported by using the design of a small retaining wall as an example. In a data flow diagram this function could be represented as in Figure 6.2. During the scheme design the task may be performed on a 'rule of thumb' basis; the same task being performed in a more analytical way at the detail design stage. Although the same task is performed in two different ways, at different stages of the design, the categories of
information required to allow that task to proceed for both stages are the same. All that differs is the level of detail of the information and its accuracy. The 'Black Box' approach of DFDs allows a single model to be formulated, representing two differing ways of performing a single task, simply because the generic category of information required to process the task is the same in both cases.

Management Tasks.
The design process consists of two separate functions: 'pure' design work and management work. This view is reinforced by Munday's (1979) observations regarding poor design performance, which he stated falls into one of two categories: the first being the poor design of the building itself and secondly organisational failures causing poor design, where management is unable to cope with the increasing complex and technological changes in the building design process. Separate work by Hollins et al. (1993) and Nijhuis (1993), in the field of production design, reaches the similar conclusion that the design process is a combination of management and design tasks.

Management tasks are not modelled in the Design Process Model because they are not necessary for it to fulfil its purpose: its aim is to capture and represent the flow of design information involved in the design of a building, rather than the processes that control its progress. Design is planned to aid the management of 'pure' design, and is not a function of design itself. This is true of all management functions, which only exist to facilitate the work performed by a building designer (see Chapter 2).

Reworking of Tasks
Within the design process there are two types of loop or cycle: iterative and repetitive. A similar conclusion is also documented by Smith and Eppinger (1992) who make a distinction between expected and unexpected iteration.
An iterative cycle, such as the coupled tasks ABCD in Figure 6.3, can be solved by firstly guessing pieces of information to allow the circuit of information requirements to be broken. Solutions to all the tasks can then be found, and the validity of the first estimates established. If the estimate is not accurate, the tasks are reworked to improve the initial estimates. This cycle continues until all the estimates are validated.

A repetitive cycle involves the reworking of various tasks to improve or change solutions, and may contain an iterative cycle. When a set of tasks has been completed, an approval is usually sought; this may be from a client or other external authority, or an internal design progress or QA check. The outcome of processes such as these is unpredictable and its effect on the design process unknown. The approval or checking processes may accept the design proposals as they stand, may totally reject the proposals or, more commonly, may suggest the proposals be fine tuned. All but the first response, requires an unknown number of tasks to be reworked. A typical repetitive cycle is shown in Figure 6.4.
The tasks (A to G) in a repetitive cycle depend on the result of the approval or checking process. The output of this process (shaded on Fig 6.4) can have various effects on the design process and depending on the output a repetitive cycle may or may not be set up. This variety and unpredictably is very difficult to model because in a DFD all the possibilities need to be captured. It was decided that any repetitive information flows and therefore repetitive loops would not be represented in Design Process Model. This is not to neglect the important issue of repetitive cycles in the design process as it will be shown in the later chapters that the techniques developed to analyse the Design Process Model allow the impact of repetitive cycles on the design process to be studied.

Information Modelled

The Design Process Model captures the flow of recordable information transferred within the design process. All information must be checked for design errors and pass quality assurance checks. This is not possible with non-recordable verbal and informal information and therefore the only mediums of information transfer considered within the model are via paper or computer. Requests for information between designers, and design queries, whether formal or not, are not modelled; these types of information flows, although common in design, are a consequence of poor planning and co-ordination and as a result all information flowing in the model is deemed to be pre-emptive, i.e. available when it is required by a task.

Recordable or traceable information flows will almost always be part of a drawing or a calculation document. This automatically defines the tasks that constitute the lowest levels of the DPM. For instance, a task that produces a set of calculations will not be decomposed as the information flowing between them would be unrecordable.

Information dependencies of a fixed nature, i.e. information from British Standards, Codes of Practices and design guides are not modelled as they are readily available and unnecessary for cross-discipline task co-ordination.

6.2.5 Modelling Approach

The requirements discussed in the previous sections influenced the modelling approach taken to form the Design Process Model and the way the design process was hierarchically decomposed for modelling purposes. Although they made modelling the design process simpler, it was essential that the model was structured in a logical fashion. The ultimate aim is to represent the design of a typical building, thus allowing the flow of information to be predicted in future projects. For the model to achieve this, it had to be structured in a manner receptive to these goals and be consistent with data flow diagramming methodology.
Chapter 6

Three modelling approaches were considered: project specific, global & tool kit models.

In a project specific approach a library of project models would be collected from individual projects. A future project would then be planned by choosing a model of a similar project from the library, and using it to predict future information flows and task orders. Although each model would be a true representation of an actual project this technique had the following disadvantages:
• it would be time consuming to produce enough models to cover all the permutations likely to occur in the design of a building;
• large parts of each model would be similar with only small differences distinguishing individual models; and
• no two projects are exactly the same; using a model of a previous project to predict the happenings of a future project is unlikely to produce a practical and representative design programme.

A global model approach would attempt to represent all the possible design eventualities in any building design project in a single model. A new project could be planned by removing the irrelevant design options from the model that would not be present in the proposed project. The remaining model would be a true representation of the processes and information flows likely to occur in the proposed project. However, this approach had several disadvantages:
• it would be extremely difficult to produce a model that covers all the eventualities likely to occur in the duration of any project; and
• the model would be cumbersome and unwieldy with large parts redundant for each proposed project.

A tool kit model approach would consist of two basic elements that would contribute to the complete Design Process Model. The first would be a 'basic framework' that modelled the high levels of the design process which were common from one project to the next. The second element would represent the various 'design options' likely to occur in the design of a building. This approach utilises the areas of commonality between different building design projects and models them separately to those areas that differ from project to project. The various 'design options' or design alternatives would be modelled discretely in separate sub-models as they contain different tasks, with different information dependencies. A Design Process Model would be formed by choosing the necessary sub-models and connecting them to the 'basic framework'.

An example of a 'design option' could be the design of the ground floor slab in a building; one of two alternatives exist: a suspended ground floor slab or a ground bearing floor slab.
Each 'design option' would be modelled separately as sub-models. For a new project, the sub-model for the preferred option would be connected to the 'basic framework' of the tool kit model.

The tool kit modelling approach was chosen as the most suitable technique for manually modelling the design process because it offered the most flexibility. This approach involved less modelling, was less time consuming and made validation simpler. It was also easier to model in smaller discrete entities (sub-models) rather than attempting to model the whole process at once.

It should be noted that although the tool kit approach was chosen to model the design process, it was appreciated that ultimately the formation of an automated model, stored in some form of computer database, may have to be stored in a global format, with all the various sub-models and the basic framework combined in a single model. This theory can be examined via a simple analogy. Consider a model containing spheres, representing design processes on a DFD, and wires representing information flows. If a tool kit approach was taken, a Design Process Model for a new project would be formed by linking the relevant spheres by plugging in all the necessary wires. This would be a complicated activity. If a global model approach was taken, the unnecessary spheres would have to be removed along with the necessary wires. Disconnecting parts of a model, rather than connecting up different sub-models, might be a simpler technique for maintaining the integrity of the model.

In summary the main attributes represented in the Design Process Model are those of:

- detail design;
- the tasks performed by designers and not those conducted as functions of management;
- the links that give rise to iterative, but not repetitive, loops;
- the flow of pre-emptive, recordable information and not informal or verbal queries or requests; and
- The Design Process Model was constructed using the tool kit modelling approach.

6.3 DEVELOPING THE DESIGN PROCESS MODEL

6.3.1 Data Acquisition for the Design Process Model

Four projects were examined in depth to provide the data to develop the Design Process Model (DPM). Projects were chosen that displayed a broad cross-section of building features and types. They are outlined below.
PROJECT ONE

Building Type : Two storey laboratory.
Contract value : £10 Million (1991)
- suspended ground floor slab;
- suspended holorib non-ground plant floor;
- insitu concrete walk on ceiling;
- steel framed structure;
- piled foundations;
- extensive internal drainage; and
- metal clad envelope.

PROJECT TWO

Building Type : Two storey corporate headquarters office building.
Contract value : £5 Million (1992)
- ground bearing floor slab;
- screeded precast concrete upper floor;
- precast concrete structural frame;
- piled foundations;
- precast concrete shear wall; and
- curtain walling envelope.

PROJECT THREE

Building Type : High bay warehouse & low bay packing facility.
Contract value : £5 Million (1989)
- 'super flat' ground bearing floor slab;
- dock levellers in ground floor;
- steel portal structural frame;
- pad foundations; and
- metal clad envelope.

PROJECT FOUR

Building Type : Three storey laboratory.
Contract value : £15 Million (1994)
- suspended & ground bearing floor slabs;
- suspended holorib non ground floors;
- steel framed structure;
- piled foundations;
- extensive internal drainage; and
- metal clad envelope.
Chapter 6

The data required to construct the Design Process Model (DPM) was collected for each project in two stages: firstly by examining all the tasks conducted during the detail design phase for each project and secondly determining all the relevant information dependencies.

Design tasks were determined by studying existing drawing lists and calculation documents, and by questioning designers on their methods. A definitive list of all tasks, and their groupings for each project was drawn up which was used to produce the design process hierarchy shown in Appendix I.

6.3.2 An Overview of the Extent of the Design Process Model

It was not possible nor appropriate to produce a complete Design Process Model as part of this research project. A complete DPM would contain a full, accurate set of data flow diagrams, representing every likely design alternative, which would have been too time consuming. This research investigated the concept of a Design Process Model and its feasibility. The following ideas were developed and some steps taken to construct one:-

(i) A hierarchy of design tasks for four design disciplines: mechanical, structural, civil and architectural; was drawn up for Project One, which was used as a standard against which Project Two was compared. Any variations in the design processes were highlighted and the differing tasks added to the existing hierarchy. Projects Three and Four were then examined in a similar fashion to form a generic hierarchy that covered the design work of the four design disciplines in all four projects. While it is appreciated that the hierarchy produced may not be truly representative of the design work involved in all future building projects, it covers a sufficient number of design options and tasks to demonstrate the feasibility of a DPM and a basis from which a full DPM can be constructed. The hierarchy developed is shown in full in Appendix I. This can be augmented by studying and incorporating design alternatives from different projects.

(ii) Project One was studied in much greater depth than the other three projects, with the aim of recording and validating all information dependencies of the design tasks performed by two disciplines: civil and structural; during the project. It was not possible to determine the information dependencies of all the remaining tasks in the hierarchy because of volume and time consuming nature of the data involved. The dependencies were collected for Project One by producing task dependency list. A typical list for each dependency is shown in Appendix II.
Figure 6.5: Work to Date on the Design Process Model
(iii) From a combination of the hierarchy (Appendix I) and the dependency task lists (Appendix II) it was possible to draw up a set of DFDs for the detail design process of the civil and structural disciplines for Project One; these are shown in their entirety in Appendix III.

(iii) The steps detailed above and the work still required to produce a complete design Process Model is shown diagrammatically in Figure 6.5.

The concepts discussed to develop a set of DFDs representing the detail design process of two disciplines studied in Project One can equally be applied to develop a complete set for the Design Process Model. To simplify matters these concepts are discussed in the following sections in the wider context of producing this complete DPM.

6.3.3 Model Hierarchy
The Design Process Model, constructed from many levelled data flow diagrams, was formed hierarchically on a discipline basis. The design process was decomposed in a top-down fashion: the top of the model represents the design process in the most general fashion; descending the model reveals an increasing level of detail until each individual task is shown at the very lowest levels. The easiest way to explain the construction of the DPM is via an example. The following discussion uses a section of the civil engineering detail design to examine all concepts of the DPM.

A typical section of the Design Process Model hierarchy is shown in Figure 6.6. This is a small section of the overall hierarchy shown in Appendix I. Level one, the detail design process, is decomposed into the five separate design disciplines of level two. These discipline design processes are architectural design and civil, structural, mechanical services and electrical engineering design. The detail design process was split in this fashion because whether a project is designed by a truly multi-disciplinary organisation or a group of separate practices, building design still occurred on an individual discipline basis. The DPM needed to mirror this traditional design environment.
Level two of the hierarchy in Figure 6.6 is then sub-divided into major sections of design work for civil engineering to form level 3. This breakdown was dependent on each individual discipline's perception of how they classify their own work. The architectural and mechanical services disciplines split their design work to mirror the way the construction work is split into work packages, i.e. a general builder's work packages or air conditioning package. The other disciplines do not breakdown their design work on this basis. For instance, one of the major civil engineering work packages is Concrete Works, covering foundations, all floors, walls and ground beams. This grouping covers a vast array of different design work and is not a sophisticated enough method for decomposing the design work for these three disciplines.

Figure 6.6: Section of Design Process Model Hierarchy
Chapter 6

The example Figure 6.6, shows civil engineering design work, as an amalgam of ground floor slab design, non-ground floor design, survey work, drainage design, pit and basement design, underground services design, foundation design and enabling work design. It is at level 3 of the DPM that the concept of a tool kit modelling approach becomes apparent; several of these sections of design work could take one of many forms. Each of these 'design options' or choices are modelled as separate sub-models. For this example the sub-models formed for foundation design are pile foundations, raft foundations, strip footing foundations and pad foundations. Each alternative involves different design tasks with different information dependencies. For the particular project shown, a piled foundation design has been chosen.

It is of interest that between the layers of the DPM there are two types of relationships, referred to as abstraction hierarchies. These are generalisation/specialisation and aggregation/decomposition (Dias 1994). The higher levels of the model which form the basic framework are based on a aggregation/decomposition or a one to many relationship (Eastman 1992). Where a design option of choice is given in the model the relationship turns to a generalisation/specialisation type. These concepts are very important in the formation of building product models.

The major sections of design work, shown in level 3 of the Design Process Model hierarchy (Figure 6.6), are decomposed into their constituent design tasks, over one, two or possibly three further levels. Each branch, at each level in the hierarchy corresponds to a single DFD. DeMarco (1979) suggests that a single DFD should fit on a single page of A4 paper; it was found if more than five design processes were shown on a single DFD below level 4, the diagram became difficult to read.

Figure 6.6 has been adapted to form Figure 6.7, which shows the link between the hierarchy and a set of levelled data flow diagrams; the added dotted lines indicate the grouping of processes to form single DFDs. The information requirements for each task in the DFDs were collected using the task dependency lists similar to those in Appendix II. The following sections explain each of the DFDs formed in detail.
6.3.4 The Data Flow Diagrams

The following sections explain how the DFDs were formed for the sections of design work outlined in the DPM hierarchy. The rationale described below is equally applicable to the formation of all DFDs within the DPM. DFDs can be drawn quickly by hand, but when large numbers of DFDs are required, seemingly simple tasks, such as revising a diagram, can make the process of re-balancing the DFDs time consuming and error prone. To overcome this CASE tools (Computer Aided Software Engineering) have been developed to support many structured analysis techniques. Traditionally, DFDs and other graphical modelling techniques, have been used in the development of software and most CASE tools convert the models of the system directly into lines of code. In addition most CASE tools, currently on the market, also have the following capabilities:

- a graphical interface to facilitate the construction of diagrams;
- support many structured methodology and diagramming techniques;
- perform automatic checking to the models produced;
- have easily understood and accessible data dictionaries; and
- have a wide range of report generating facilities.
A CASE tool called SELECT, produced by Select Software Ltd., was used initially in the research to prepare DFDs. This software was replaced by a more sophisticated tool called SYSTEM ARCHITECT, two years into the project; all the DFDs shown in the following sections and in Appendix III were produced by the latter tool. CASE tools are capable of generating code from data flow diagrams with a complete data dictionary, in which all the items (processes, information flows etc.) are internally balanced. As the primarily goal was to produce a paper based model of the design process, it was not necessary to perform the time consuming task of defining all items in the data dictionary. However, data flow diagramming methodology has been strictly adhered to, and the balancing of information flows, between levels has been considered, although not formally defined, in terms of data dictionary entries.

Level 1 : The Context Diagram

The DFD that constitutes the context diagram is shown in Figure 6.8. It is a representation of the whole system under consideration and what it communicates with. Our system is The Detail Design Process, which is dependent on a combination of client provided information and scheme design information. It has been assumed for this model that the scheme design has been conducted by an external design organisation. The context diagram also shows the destination of the output from The Detail Design Process, namely the detail design information required for the construction site.

![Figure 6.8 : The Context Diagram](image)

Level 2 : Diagram 0 : The Detail Design Process

The Detail Design Process represented in the context diagram is partitioned into the relevant discipline designs to form diagram 0. This partitioning follows the breakdown set out in the DPM hierarchy shown in Figure 6.9.
It can be seen that a single information flow leaves this DFD: detail design information. This corresponds to the information being outputted from its parent DFD. However, this is not the case for the information entering diagram 0, which are client provided information and scheme design information. This is equal to that entering the context boundary of the child DFD which is detail design briefing information. As all DFDs must be balanced between levels, these three information flows must be equated in the data dictionary. A data dictionary defines all components of a DFD by classifying their individual characteristics, and allows a top-down partitioning of data (DeMarco 1979). Therefore, for all the information flows to balance the detail design briefing information can be made up of either client provided information or scheme design information or both. This concept of grouping information together is very important in the formation of the DPM and will be expanded upon in a following sections.

In a design project each discipline is in a constant two way correspondence with all the other disciplines. If the five disciplines were shown on a single DFD, twenty information flows would be required to show this two way communication (which would also result in the breaking of DeMarco's rule of never allowing information flow lines to cross). This has been avoided by introducing a process called Document Control, which has the effect
of reducing the number of information flows and making the DFD more readable. This process collects *checked discipline designs* from each discipline and distributes it to the others as *issued design information*. The *Document Control* process, although introduced to keep the DFD readable, could conceivably represent a central administration department in a design organisation; the typical processes of which are shown diagrammatically in Figure 6.10. This DFD shows that all *checked discipline designs* are initially registered and then printed or copied before issuing to other disciplines. It is assumed that the process of *Document Control* could be adapted to model the administration department of any design organisation.

![Diagram 6: Document Control](image)

Figure 6.10: Diagram 6: Document Control

Figure 6.9 is not as a strict representation of design in that no two disciplines are shown to communicate directly. However, one of the modelling assumptions made earlier in this chapter was that all information flows must be recordable and it has been suggested by Murray (1993) that this representation of inter-disciplinary communication is more representative of a traditional design project than a multi-discipline organisation, where all the disciplines are separate practices and the correspondence is more formal.

**Level 3: Diagram 4: Civil Design**

This data flow diagram, shown in Figure 6.11, is formed by decomposing process 4 of diagram 0. This decomposition follows the DPM hierarchy, which indicates the major sections of civil engineering design work. This DFD and its parent are balanced because the information crossing both context boundaries is identical.
Again, the large number of processes in this DFD dictates the need for a central co-ordination process, to keep it readable. As with the Document Control process on the parent DFD, Civil Co-ordination does represent a function performed by each discipline. It was found from the study of design teams that for Quality Assurance purposes, all information issued to other disciplines must be checked and signed off. The Civil Co-ordination, partitioned to form Diagram 4.1 (Figure 6.12), models these processes. All unchecked design information, from the major sections of design work, entering the process 4.1.2, is processed and then corrected and/or sent to a regulating authority for checking. Other checked documents are submitted for QA checks and then distributed.

Issued design information from any of the other disciplines does not need checking as part of the Civil Co-ordination process and is therefore distributed directly via process 4.1.1, the output of which takes one of two forms: checked civil design or issued or checked design information (ICDI). ICDI, in this case, is a combination of checked civil design and issued design information and is defined in the data dictionary as being so. ICDI is distributed to all sections of work, whereas checked civil design is distributed to the Document Control process in diagram 0. The rationale behind ICDI is further explained in the next section.
Level 4: Diagram 4.2: Foundation Design

The process Foundation Design, shown in Diagram 4 (Figure 6.11), is decomposed into the various attributes that constitute piled foundation design in line with the sub-model chosen in the DPM hierarchy to form Figure 6.13. The hierarchy also shows that the piled foundation design is made up of seven individual design tasks which are partitioned over levels 4 and 5 to keep the number of tasks per DFD, at the lowest levels of the DPM, to below five.

Three information flows leave the context boundary of diagram 4.2: unchecked pile design, unchecked pile cap design and unchecked column casing drawings, compared to the single flow leaving the parent. Again to balance the child and parent, the information flows must be equated in the data dictionary. The relationships between these items is such that the information flow unchecked foundation design could be either unchecked column casing drawings or unchecked pile cap design or unchecked pile design or any combination of the three. However, using the notation of DeMarco for describing the top-down partitioning of data, the relationship can be more simply stated as thus:

\[
\text{unchecked foundation design} = \text{unchecked column casing drawings} + \text{unchecked pile design} + \text{unchecked pile cap design}
\]
This shorthand method, developed to overcome writing out each relationship long hand, is explained in more depth by DeMarco (1979).

From the DPM hierarchy it can be seen that the process *Column Casing Drawing* is not decomposed into any further sub-tasks. A task such as this is known as a functional primitive task (FPT), and is the lowest level of a particular branch of the DFD model. Each functional primitive task is indicated as a box with rounded corners on the DPM hierarchy (Figure 6.6) and as a process with a thicker outline on a DFD (process 4.2.3 in Figure 6.13).

Flowing into this functional primitive task, *column casing drawing*, are the pieces of information, required to perform this process. The dependencies are shown on this particular DFD as coming from documents, produced by other FPTs, elsewhere in the DPM. Also the DPM hierarchy explicitly shows the physical locations (Drawings,
calculations etc.) of all the design inputs and outputs. In a true DFD the source and destination of these information dependencies would not be shown as the relationships would be defined uniquely in the data dictionary. The sources have only been shown here to highlight how one FPT relates to another in the model.

In reality to ensure the parent and child balance the information flows entering the process *Column Casing Drawing*, such as *junction with ground floor*, must equate to the flow entering the parent, *ICDI*. The term *ICDI* has been introduced to generically group information flows together to ensure DFDs in the higher levels of the DPM do not become cluttered up with individual information dependencies. If, for example, the information dependencies for the process *Column Casing Drawing*, plus those for the six tasks that make up *Pile Design* (4.2.1) and *Pile Cap Design* (4.2.2) (Figures 6.14 and 6.15 respectively) were not grouped, the information flowing between the processes *Foundation Design* and *Civil Co-ordination* would be as shown in Figure 6.13a. This is clearly impractical, hence the need to group information flows as the model is ascended.

Figure 6.13a: Diagram Showing Need to Group Information Hierarchically
Chapter 6

4.2.1 Piling Design

4.2.1.1 Piling Calculations

4.2.1.2 Piling Layout Drawings

4.2.1.3 Piling Schedule Drawings

4.2.2 Pile Cap Design

4.2.2.1 Pile Cap Calculations

4.2.2.2 Pile Cap Layout Drawings

4.2.2.3 Pile Cap Detail Drawings

Figure 6.14: Diagram 4.2.1: Pile Design

Figure 6.15: Diagram 4.2.2: Pile Cap Design

93
The grouping of information flows is done on a hierarchical basis in the data dictionary. A typical breakdown of the information flows for the Foundation Design section of the DPM is shown in Figure 6.16. This diagram is consistent with the data dictionary examples previously discussed. For example it can be seen again that unchecked foundation design could be either unchecked column casing drawings or unchecked pile cap design or unchecked pile design or any combination of the three.

![Diagram 6.16: Hierarchical Breakdown of Information Flows]

**Figure 6.16: Hierarchical Breakdown of Information Flows**

**Level 5: Diagram 4.2.1: Pile Design and Diagram 4.2.2: Pile Cap Design**

Partitioning the processes 4.2.1 and 4.2.2, on diagram 4.2, results in the DFDs shown in Figure 6.14 and Figure 6.15 respectively, and the DFDs contain only FPTs. Again, the sources of the information dependencies for each FPT have been shown, although they would not normally be required. To balance the two DFDs with the parent the following relationships are defined in the data dictionary.
Chapter 6

details from pile cap design = \begin{align*}
pile position and number + 
ground floor/beam loads on pile cap + 
pile setting out dimensions
\end{align*}

and

details from pile design = \begin{align*}
pile cut off level + 
pile diameters + 
pile groupings + 
piling layout
\end{align*}

These two DFDs form the bottom level of the DPM because the functional primitive tasks cannot be divided into sub-tasks because the information that would flow between them would not be recordable; all FPTs in the DPM produce a recordable and traceable document.

The three recognised outputs or deliverables from the design process are drawings, calculations and specifications and, as discussed in Section 6.4.3.4, these documents permit the flow of information from one process to another to be recorded. Addis (1990) described calculations as the 'justification' of the design process and drawings and specifications as the 'description'. However, only one form of 'description' has been represented in the model, namely the drawings, because the production of a specification was not found to affect the co-ordination of design tasks, nor were tasks found to be dependent on information from a specification.

6.4 VERIFICATION AND VALIDATION OF THE DESIGN PROCESS MODEL

6.4.1 Definitions

The terms verification and validation are often used synonymously; even the dictionary (OED 1990) defines verification as the process of validation. Many authors in the field of computer simulation refer solely to using validation to check the accuracy of their models (Wilson 1990, Pidd 1992). However, it is convenient to differentiate between the two terms because two different procedures were used to ratify the Design Process Models.

- verification is an internal check that the components of the DPM are a correct representation of the data flows and processes that were observed in building design (based on the projects studied); and
• validation is an external check of the integrity of the DPM as a representation of the observed building design process and its suitability for modelling other (future) building designs, including predicting the effects of changes in data flows.

Verification and validation of the models was a cyclic process. Firstly the model produced for Project One was verified by the designers and architects responsible for the design. Validation occurred by using the model of Project One to predict the design process of Project Two. Any differences were accommodated in the revised model, which was then verified in the same way as Project One. This cycle of verification/validation produced a model that got bigger (more components) and more generic.

6.4.2 Verification

At an early stage preliminary data flow diagrams were constructed to model small sections of the civil engineering design process, such as storm drainage design. These were drawn up, not only to ascertain the level of detail to be represented in the model, but to identify the quickest and simplest way of verifying the complete DFDs with engineers and architects. Several difficulties were encountered with designers directly verifying the DFD models of the design process, with the following questions being constantly asked: Why have you opted to model these tasks? Why to this level of detail? Why has the design process been decomposed in this manner?

As a result, it was decided to verify the models produced at each stage in a series of steps, ensuring the designers were not confronted with large quantities of information, as well as new concepts, in a single session. As the Design Process Model was decomposed on a discipline by discipline basis, verification could follow along the same lines and was conducted in the following steps:

• verification of the task hierarchy for each project;
• verification of the information dependencies for the functional primitive tasks; and
• verification of the DFDs.

The construction and verification of each project hierarchy proceeded iteratively and was conducted in close consultation with the relevant designers. The first step in forming the hierarchy was to determine the major sections of design work for each discipline, and was integral with the breakdown of individual design processes (the factors that influence this have been discussed previously in Section 6.3.3). The second step involved the study of drawing lists, calculation documents and the like to deduce the functional tasks for each discipline and incorporate them into a single hierarchy. This was then presented to the
various engineers and architects and a fully verified design process hierarchy evolved from a series of consultations.

Task dependency lists were constructed for all FPTs occurring in Project One. This was a straightforward, but time consuming, process that involved considering the information needs of each task, and then verifying them with the appropriate designer. It should be noted that the source of each information dependency was also verified at the same time.

Verification of the DFDs (a combination of the DPM hierarchy and the task dependency lists), was a simpler process that was not absolutely necessary. However, typical DFDs were put in front of the designers involved in the respective projects to explain what their previous work had helped to create. Although it was felt that DFDs were self-explanatory, it was found that three particular doubts were raised:-

(i) It was suggested the model was too idealistic and ignored many of the complexities that make design unique.

(ii) The validity of the document control and discipline co-ordination processes were questioned.

(iii) The emphasis of information flow in the model, rather than its control or timing was also queried.

The ultimate aim of the DPM was to be an integral part of the design planning methodology and by bearing this in mind the doubts listed above were shown not to be critical.

(i) The model had to be idealistic; it was not necessary to represent every meeting, communication or decision occurring during the design process, only those tasks that may appear in a design programme.

(ii) These two tasks suggest that no two disciplines are in direct communication with each other. However, they are purely artificial processes introduced to keep the DPM readable and will not be analysed or considered in any design programme produced subsequently.

(iii) Data flow diagrams represent a system from a data or information viewpoint. Information availability dictates whether a task can be performed, not a management process. Managing or controlling information flow can be improved if information flow is more clearly understood.

Overall, the engineers and architects, some twenty in total, consulted during the many stages of verification were impressed by the clarity with which DFDs summarised their design processes. At the time of writing this thesis a major collaborator in this research is
actively pursuing the production of data flow models of each disciplines' design process based on the DPM developed in this work.

6.4.3 Validation

Validation of any model is an ongoing process and was primarily performed by using the models produced from previous projects to predict the likely design process for another. Although this research based itself around multi-disciplinary design organisations the DPM has not been validated on single discipline design practices. However, it is felt that there should be few problems in using the DPM in this regard as the modelling proposals were developed specifically to pre-empt many of them.

Further validation of the DPM occurred when the model was used as part of the prototype design planning methodology. This was a truer test of the validity of the model because the design programmes produced had to be not only representative but workable. The analysis of the DPM also revealed natural clusters of design tasks that designers readily identified as being areas in which particular co-ordination problems occur. This is discussed further in Section 8.3.6.

The detail design phase was modelled because it was more predictable than either scheme or concept design and much the same tasks were performed from one project to the next. This was borne out in the development of the DPM hierarchy which was based on the detail design work from four separate projects. As approximately 80% of the tasks and their information dependencies appeared in all four projects, it is felt that the DPM hierarchy, although based on a limited number of projects, is likely to cover approximately 85-90% of design tasks encountered in the detail design of similar buildings.

The DPM does not model the way in which each task is performed, it only shows the information required and produced by each task. This black box approach allows the same tasks, usually performed in differing ways for different designers, to be modelled in a unified manner. Therefore, the DFDs drawn up to show the information inputted and outputted from tasks should be valid one project to the next. For example, a pad foundation, however it is designed, will always need an appreciation of the column loads, soil properties and structural philosophy. These information dependencies do not vary from building to building or from project to project and therefore, a DFD representing this process, if verified, will be valid for all projects.

As the managerial structure varied from organisation to organisation, management influences on the design process were ignored in the model, making the model less specific to individual projects and easier to validate.
Chapter 6

6.5 APPLICATIONS OF THE DESIGN PROCESS MODEL

Although the DPM was constructed as an integral part of a methodology to improve the planning of building design work, it was found to have more immediate and realisable applications:

- an indicator of good design practice;
- an immediate aid for traditional methods of planning;
- a tool to study the impact of design errors, omissions and variations; and
- a tool to analyse and suggest improvements to the design process.

These applications of the DPM are studied in more depth in the following sections.

6.5.1 Developing Good Design Practice

The DPM is a simple representation of the design tasks performed, and the information requirements necessary to perform the detail design of a building. This detailed breakdown of the design process can form the basis of a communication tool or procedural manual, that will unify designers' thoughts and help introduce consistency into design work. The model not only reminds engineers and architects what information is required to perform their tasks, but also identifies the recipient of the information their tasks produced, which should improve cross-discipline co-ordination.

Pollard and Plume [1993] suggest their detailed model of architectural design is an ideal tool to assist communication, either among disciplines or between the client and design team. Traditionally, a lack of communication has been a major criticism of the building industry, and any 'working negotiating instrument' (the DPM) is a step in the right direction.

6.5.2 Design Planning

The DPM is a tool that could aid planners currently using conventional scheduling techniques. This view is reinforced by Hanby et al.(1993) who, as part of parallel initiative to this work, have produced detailed models of HVAC design. They believe that the identification of critical design interfaces for 'lay' (in design terms) construction planners is undoubtedly of benefit. Studying the model can help the planner understand the flow of information and predict where cross-disciplinary iterative design is necessary. This will then allow the interdependent tasks from different disciplines to be scheduled in the same time span to facilitate the flow of cross-disciplinary information. The task
hierarchy of the DPM can also be used as a checklist when determining what tasks should be planned and for estimating the fees for design work.

As will be explained in the following chapters the DPM forms an integral part of the prototype design planning technique (ADePT) developed as part of this research.

6.5.3 Analysis of Change
The DPM produced as part of this work and DFD models produced to represent multidisciplinary scheme design, are currently being used by Baldwin et al. (1994) to study the effect of change and variations on the design process. Their approach applies a discrete-event simulation package called GENETIK to DFDs and the work hopes to study the effects of:

- changes in design tasks;
- the availability of design information;
- changes in procurement strategy;
- the delivery of design information for construction; and
- changes in the availability of design staff.

Another approach by Hedges et al. (1993) uses directed graph theory and applies it to detailed DFD models of the HVAC design process. They use a process called event tracing to study the effect of changing an input information flow in which a depth-first search traces all the tasks affected by the changed information flow and produces a list. They are currently trying to refine this analysis by assigning salience values, or a weighting, to each design task, so a certain degree of selectivity can be achieved in the reporting of those tasks affected.

While both approaches discussed above promise to yield practical tools that may help design managers forecast the impact of change, in terms of cost and time, the work is not yet complete and the validity of their results is absolutely reliant on accurate models of the design process.

6.5.4 Design Process Analysis
During the pilot study conducted to determine proposals for the DPM, a set of ten or so DFDs were drawn up to represent the conceptual design work undertaken in Project A (see Section 6.2.1). This model, developed at an early stage of this research, is not presented in this thesis as subsequent work has concentrated on detail design. Although the model was not comprehensive, its analysis yielded some thought provoking insights into conceptual design. The DFDs were used to draw up a retrospective design programme which allowed the flow of information from the client to be studied with respect to time. It was shown
that the client briefing meetings were very detailed; the end users gave the design team a
great deal of information, much of which was superfluous to the design work being
performed for that particular stage of design. This is shown diagrammatically in Figure
6.17. Too much information was given to the design team at too early a stage in the
project enhancing the chance of information overload. The DFDs showed what
information was needed from the client for each task, and when combined with the design
programme it was possible to plan a more structured approach to briefing.

![Figure 6.17 : Information Filtering](image)

6.6 CONCLUSIONS

This chapter has discussed the formation of a model to represent the detail design of any
modern building using data flow diagrams, a technique examined in depth in Chapter 5.

An initial pilot study of three different projects, conducted to determine the exact
requirements of the Design Process Model (DPM), concluded that the model should only
represent:

- detail design;
- the tasks performed by designers and not those conducted as functions of management;
- the information links that give rise to iterative, but not repetitive, loops; and
Chapter 6

- the flow of pre-emptive, recordable information and not informal or verbal queries or requests.

These decisions were taken to simplify the process being modelled to such an extent that the data acquired to formulate the model could be collected and used in a consistent manner, whilst not compromising the objectives of the model.

Four projects were studied in detail to formulate the Design Process Model. The data was collected and analysed in two stages: data to form the design task hierarchy; and data to determine the information requirements of each task in the hierarchy. The results of these stages were then verified by practising designers and combined to form the data flow diagrams in the Design Process Model. This method of gradually forming the model allowed the verified model of the first project to validated by using it as a starting point for analysing the second project. Using this method a Design Process Model was formed that was detailed and accurate enough to represent 85-90% of similar building design projects.

The objective of this phase of the work (as defined in Chapter 3) was to gain a clearer understanding of the building design process and to form a model that can be an integral part of an alternative approach to planning design work. The Design Process Model maps the flow of information between tasks and disciplines in a generic way so as to achieve these objectives.

Although the DPM was constructed as an integral part of a methodology to improve the planning of building design work, the following 'stand alone' applications of the DPM were identified:

- an indicator of good design practice;
- an immediate aid for traditional methods of planning;
- a tool to study the impact of design errors, omissions and variations on the design process; and
- a tool to analyse and suggests improvements to the design process.
Chapter 7
Chapter Seven

DEVELOPMENT OF A PROTOTYPE DESIGN PLANNING METHODOLOGY

7.1 INTRODUCTION

This research project set out to develop a fresh approach to the planning and co-ordination of multi-disciplinary design work. As discussed in Chapter 3 this aim was approached in two separate phases: first by gaining a clearer understanding of the building design process and second by developing a prototype methodology to plan building design work more efficiently. The first phase was discussed in the previous two chapters, in which a Design Process Model was proposed and constructed. This chapter describes a technique called Design Structure Matrix Analysis previously used in product design and examines the potential of combining it with the Design Process Model to produce an alternative approach to planning building design work.

7.2 DESIGN STRUCTURE MATRIX ANALYSIS

7.2.1 History

Design Structure Matrix Analysis (DSMA) evolved from a technique developed to solve large sets of simultaneous equations within engineering systems. The original technique, based on Boolean algebra, was developed at the beginning of the 1960's by Steward (1962). Traditionally, simultaneous equations were solved using iterative algorithms to produce numerical solutions; as engineering problems became more complex the number of equations increased, making analysis and data handling more onerous and often exceeding the computer storage and processing capabilities of the day. Experience had shown that large systems need not be solved simultaneously, but could be broken up into a series of sub-systems, containing smaller sets of equations which could be solved simultaneously but independently (Ledet & Himmelblau 1962). If the sub-systems were ordered into a sequence such that each group could be solved independently of the remaining sub-systems, a solution could be deduced sequentially rather than iteratively. This had the following benefits:

- the computer storage required corresponded roughly to the amount of memory required to solve the largest sub-system; and
- the computing time required to realise a solution was substantially reduced.
Steward (1962) defined this process of re-arranging sub-systems as decomposition, in which each sub-system of equations was ordered by the natural information flow among the equations. Steward continued to develop these concepts into Design Structure Matrix Analysis (DSMA) and, apart from isolated attempts by groups to use DSMA as a tool to analyse models of the economy and the control system of a rocket (Steward 1981a), the technique received scant attention and development. However, work by Eppinger et al. (1990) showed that this technique had potential in the design domain with his work in the product design environment. The following sections will discuss the general philosophies developed by both Steward & Eppinger of applying Design Structure Matrix Analysis to a design process.

### 7.2.2 Representing the Design Process

Consider a set of seven tasks, T to Z, representing activities from a small design problem. The design process is a system and therefore each task and their information dependencies can be represented as a directed graph, see Figure 7.1. The task dependencies can be represented by crosses or marks in a matrix; a mark in row $i$, column $j$ means task $i$ is dependent on task $j$. From the directed graph (Figure 7.1) task U is dependent on task V, a relationship also shown in the Precedence matrix (Figure 7.2); the mark in row U, column V, indicates that task U is dependent on task V. It should be noted that diagonal marks in the matrix where $i = j$ are essentially meaningless and have been shown shaded to highlight the leading diagonal which separates the upper and lower sections of the matrix.

![Directed Graph of a Typical System](image)

**Figure 7.1 : Directed Graph of a Typical System**

This precedence matrix, prior to any further analysis can be a useful tool in its own right. Although designers usually know the information on which their design tasks are dependent (shown by marks in the relevant row) the matrix allows designers to predict which tasks use the information resulting from their work. This can be achieved by
examining the marks in the relevant column; for example looking down the column Z shows that tasks T, V, and Y are all dependent on information from task Z. This property of the precedence matrix can be used to ensure only those parties requiring the necessary information receive it, therefore reducing the chances of information overload. This property has been highlighted by Steward (1981b) as a method for controlling changes and revisions. He uses an algorithm called 'Aspect' to trace through a succession of tasks to find and select those tasks that will be affected by a variation so they may be notified. In the example above, a change to task Z means task T, V and Y may need to be revised.

Figure 7.2: Precedence Matrix Developed from Directed Graph

7.2.3 Task Order Within a Matrix

The order for performing tasks within any matrix is assumed to begin in the top left hand corner and proceed to the bottom left. In the precedence matrix in Figure 7.2, task T would be performed first, task U second and so on, finishing with task Z. In a matrix a mark below the leading diagonal means the information has been produced because the generator task has been completed. Conversely, any mark above the leading diagonal means the generator task has not been completed and an estimate of the information will be required. Performing the tasks in the order shown in the precedence matrix means task Y is performed after task V, but before task Z, even though task Y is dependent on both tasks V and Z for information. An estimate of the information from task Z would need to be made if task Y was to be completed in the order prescribed in the matrix. Estimating information gives rise to a phenomenon called iteration in which the estimates have to be verified and possibly refined, potentially involving the reworking of tasks previously completed. The task order shown in the precedence matrix in Figure 7.2 actually suggests all tasks are in a single iterative loop; the first task, task T, is dependent on the last, task Z and therefore the original estimate cannot be verified until task Z has been completed. Ideally the number and size of iterative cycles should be kept to a minimum if an expeditious design process is sought.
Chapter 7

Therefore, the first stage in DSMA is to re-order the tasks in the matrix with the aim of forcing all the marks below the diagonal and eliminating the need to estimate information. If this can be achieved the matrix becomes 'lower triangular'. This process of re-ordering tasks to try and achieve a lower triangular matrix is called 'partitioning' and is explained below.

7.2.4 Partitioning the Matrix

The aim of partitioning is to re-sequence the design tasks to maximise the availability of information, (Gebala and Eppinger 1991). The re-ordering, achieved by simply swapping rows and the corresponding columns, tries to confine all information dependencies below the leading diagonal. The partitioning process also tries to manipulate any iterative cycles, so that their number and size are kept to a minimum, thus reducing the total amount of iteration required, and consequently the duration of the design process.

Partitioning the matrix in Figure 7.2 leads to the matrix Figure 7.3. The number of information dependencies above the diagonal, and therefore the number of estimates required, has been reduced dramatically. However, it has not been possible to confine all the dependencies below the diagonal because no matter how the tasks Z, X, V and Y are ordered, marks will always occur above the diagonal. These tasks are said to be 'coupled' or 'interdependent' and produce a 'block' or 'circuit' of tasks that require solving iteratively. Partitioning aims to reduce the number of tasks in a circuit to a minimum and group the block astride the leading diagonal. Once this circuit has been resolved, task U, T and W can be solved sequentially, without the need of further iterative deduction.

\[
\begin{array}{cccccccc}
Z & X & V & Y & U & T & W \\
Z & \times & \times & \times & \times & \times & \times \\
X & \times & \times & \times & \times & \times & \times \\
V & \times & \times & \times & \times & \times & \times \\
Y & \times & \times & \times & \times & \times & \times \\
U & \times & \times & \times & \times & \times & \times \\
T & \times & \times & \times & \times & \times & \times \\
W & \times & \times & \times & \times & \times & \times \\
\end{array}
\]

Figure 7.3 : Partitioned Matrix

The task order presented by partitioning is more efficient than the unanalysed order suggested in precedence matrix. Partitioning is a systematic process and may be conveniently performed by computer. Partitioning is the same as the process of decomposition mentioned in Section 7.2.1, which aims to divide the system into the smallest irreducible sub-systems, defined by Eppinger (1991) as independent tasks,
dependent tasks or groups of interdependent tasks. Most partitioning algorithms consist of two basic steps which are repeated until all tasks have been scheduled: firstly scheduling the tasks that are not components of any circuits and secondly identifying the circuits of coupled tasks. The first step is a straightforward process. In contrast the identification of the loops is more involved and several different approaches have been put forward to achieve this. The most well documented of which are: path searching Steward (1981a), Boolean manipulation of the Adjacency Matrix (Ledet and Himmelblau 1962), and rule based expert systems (Rodgers 1989 and Rodgers and Padula 1989). These methods are examined and evaluated in more detail in the following chapter.

After partitioning further analysis can be performed to optimise the task order within these circuits. This process is called 'tearing'.

7.2.5 Tearing the Matrix

The aim of tearing is to re-sequence tasks within a circuit of coupled tasks to find an initial ordering to start the iteration (Gebala and Eppinger 1991). The ordering of tasks within a block affects the number of marks above the diagonal, and hence the number of estimates required. Tearing involves eliminating dependencies from the matrix so that the reduced block can be re-partitioned, or re-ordered, making it lower triangular. This facilitates the quickest solution to the iteration. Tearing a block is an arbitrary procedure done by considering a combination of:

- the re-ordering of tasks to minimise the number of estimates; and
- choosing the dependencies that can most be suitably estimated.

Effective tearing requires a detailed knowledge of the problem domain so that the less important elements are torn from the block to leave only the essential ones (Eppinger et al 1993). Although essentially a subjective process analytical techniques, such as shunt diagrams, have been developed to aid the choice of the most suitable tear (Steward 1991). Tearing the block in Figure 7.3 leads to the Design Structure Matrix in Figure 7.4.

![Figure 7.4: Design Structure Matrix](image-url)
As this example is purely hypothetical, the block has been re-ordered to reduce the number of estimates to a minimum, rather than address the semantics and imposing engineering judgements upon the tearing procedure. This has reduced the number of estimates from 4 to 3, which should permit a more expeditious solution than using the task order from the partitioned matrix.

Partitioning and tearing a matrix produces the Design Structure Matrix which indicates the most efficient order to perform design tasks. This order has been derived from the co-ordination of the flow of information between tasks to minimise the number and size of iterative cycles and ensure all other tasks are carried out in series or parallel.

7.2.6 Product Design

The use of DSMA, as a management tool, was initially considered by Steward (1981a) and later by Eppinger and his team (1990). Both authors have applied these techniques to the design of products in the manufacturing industry. Eppinger has worked closely with General Motors to develop matrices for the design of car parts and to suggest ways in which product design team organisation may be improved. McCord & Eppinger (1993) in a study of the design of a small engine block compared the existing design team structure against the design groupings suggested by the matrix analysis. The analysis clearly indicated a more logical grouping of tasks which would facilitate information transfer between the teams performing them.

Both Steward, and to a lesser extent Eppinger and his team, have shown that DSMA can effectively analyse and order tasks within a design environment and suggested that DSMA has the potential to be a practical design management tool.

7.2.7 Building Design

Chapter 2 showed that the problems of poor information co-ordination and poor design planning seriously affect building design performance. Project managers, now common place on building projects, have the responsibility of overcoming these problems by ensuring information transfer within the design team occurs in an efficient and co-ordinated manner. However, at present this is achieved by a reliance on intuition and experience rather than analytical techniques. DSMA, in describing and planning the transfer of critical information between activities, can be successfully exploited to assist a project manager to programme and manage the design process.

DSMA is not strictly applicable to building design. This is because building design requires the co-ordination of hundreds as opposed to tens of design tasks. Both Steward
and Eppinger base their analysis on matrices of less than fifty tasks and they assume that
design tasks and their dependencies are pre-defined when forming the initial precedence
matrix. Buildings are now so technically complex that the detailed comprehension of the
building design process requires a method of analysis to explicitly define the tasks
involved. Hence the formation of the precedence matrix for a building design must
incorporate some form of methodology for task analysis.

Both DSMA and the Design Process Model (DPM) are independent tools that can benefit
the management of the design process. However, this research set out to develop a fresh
approach to the planning and co-ordination of multi-disciplinary building design. Neither
DSMA nor the DPM can be directly used to plan design work to produce effective and
workable design programmes: DSMA because it requires a detailed pre-determination of
tasks and their dependencies; and the DPM because DFDs are not time dependent.

It was hypothesised that the combination of the two techniques would give rise to a
powerful design management tool. DSMA had the potential to represent and analyse the
complexities of the building design process and programme the work based on the optimal
flow of information whereas the DPM, developed in Chapter 6, represented the tasks and
dependencies required to form a precedence matrix. The hypothesis that the DPM and
DSMA could be combined to produce prototype planning methodology was tested initially
on small design examples to determine its practicality and robustness. The results were
encouraging and led to the testing of larger case studies to develop a more rigorous
technique, ADePT, which will be discussed in the following chapter. The remainder of
this chapter concentrates on the concepts of the prototype methodology and goes on to
discuss an example which was used to determine the feasibility of combining DSMA with
the DPM.

7.3 PROTOTYPE DESIGN PLANNING METHODOLOGY

7.3.1 Summary

The prototype design planning methodology developed by combining the DPM and
DSMA consisted of 4 steps. These are shown diagrammatically in Figure 7.5 and are
summarised below.

Step 1
Use the DPM to choose the tasks required to perform the detail design of the project under
consideration and then extract the data on the relevant tasks and their dependencies to
formulate a precedence table.
**Step 2**
Convert the precedence table into a precedence matrix, and then perform Design Structure Matrix Analysis to optimise the information flow and produce an efficient task order.

**Step 3**
Convert the Design Structure Matrix into a suitable discipline based logic network.

**Step 4**
Unwrap any iterative cycles and define task durations to produce a design programme.

---

Figure 7.5: Prototype Design Planning Methodology

The following sections discuss these steps with reference to an example, the design of small plant room as shown in Figure 7.6. The plant room design was considered to have been developed up to a scheme design and the subsequent detail design work required the input of four separate disciplines. This example, used in the development of the prototype design planning methodology, was chosen because the limited number of interdependent design tasks enabled the methodology to be assessed without becoming engrossed by detail while still requiring cross-discipline co-ordination.
7.3.2 Step One : Analysing the DPM

The first step in producing a design programme was to consider the tasks needed to complete the detail design phase. Normally the tasks and their dependencies would be described in the DPM and a model specific to an individual project formed from it. However, this example was tested before the Design Process Model was finalised and some of the tasks shown in the task hierarchy for this project in Figure 7.7 were defined slightly differently to those in the finalised DPM hierarchy. However, the principles for the constructing of the DPM were adhered to and the model developed for this example was analysed in exactly the same way as the DPM would have been.

The task hierarchy, shown in Figure 7.7, shows seventeen functional primitive tasks (FPTs), divided among four design disciplines, forming the lowest level of the model. The FPTs, indicated in Figure 7.7 as boxes with rounded corners, were given a unique identifier, from A through to Q, that corresponded to the document it produced. Two typical DFDs, relating to the FPTs ringed with the dotted lines, are shown in Figure 7.8 and Figure 7.9. Although the remaining DFDs for this model have not been included in this thesis a complete set were constructed to assess the information dependencies of each FPT in the hierarchy.
Chapter 7

Figure 7.7: Plant Room Design Process Model Hierarchy

Figure 7.8: Data Flow Diagram for 'Frame Design.'
Each functional primitive task in a DFD was dependent on information from documents produced by other FPTs in the model and any design programme developed needed to order these tasks by co-ordinating the information flows amongst them. The DPM at its functional primitive level was considered as a closed system, in that all information entering a FPT was the output of another FPT in the model. This property of the model allowed a precedence table to be constructed, listing each FPT and their dependencies. The precedence table developed is shown in Table 7.1 and contains only information extracted directly from the DFDs; for example the table shows that task K was dependent on tasks F, H & J for information which corresponds to the information shown in the DFD in Figure 7.9.
7.3.3 Step Two: Design Structure Matrix Analysis

Precedence tables, being a list of tasks and their dependencies, were found to be ideal because they could be directly converted into a precedence matrix. Figure 7.10 shows the precedence matrix formed from Table 7.1 and its subsequent partitioning produced Figure 7.11.

Table 7.1: A Precedence Table

<table>
<thead>
<tr>
<th>Tasks</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Floor Calcs</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Ground Floor Drgs</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<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>Foundation Calcs</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<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>Foundation Drgs</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<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>Column Casing Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<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>Air Handling Unit Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Duct Route Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Ahu/Ducting Calcs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Mezz. Floor Slab Calcs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Mezz. Floor Slab Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Mezz Floor Swt Calcs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Column Sizing Calcs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Bwk. Wall Details Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Steelwork Plan Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Swt. Sections Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Elec Switchgear Calcs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>Elec Switchgear Drgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>

* - Information Dependency

Figure 7.10: Precedence Matrix
A mark in the matrix only represents dependency and does not express to an engineer or planner the nature of the information, or indeed the number of dependencies. The semantics of the circuit was studied by consulting the relevant tasks and information flows in the DFDs. In assessing the relative importance of each information flow in the circuit, the least essential dependencies were highlighted. Figure 7.12 shows the partitioned matrix in which certain dependencies were identified as suitable placed to make tears.

![Partitioned Matrix](image)

Figure 7.11: Partitioned Matrix

![Partitioned Matrix with Tears Identified](image)

Figure 7.12: Partitioned Matrix with Tears Identified
For example task K, the mezzanine floor steel work calc, was dependent on three pieces of information: AHU weight and position, from task F; floor slab thickness, from task I; and size of hole required by duct from task H. From Figure 7.12 it was judged that information from task I and H was not absolutely necessary to perform task K, whereas information from task F was deemed essential. These decisions, based on engineering judgement suggests that the floor slab thickness (task I) could be predicted with a fair degree of accuracy and the size of the holes required by the duct (task H) would not affect the steel work spacing. Further possible tears were identified and are also shown in Figure 7.12.

The circuit of coupled tasks was then re-ordered or re-partitioned so that only information identified as 'tearable' appeared above the diagonal; any possible tears falling below the diagonal meant the information did not need to be guessed or estimated. The fully ordered set of tasks in the Design Structure Matrix (DSM), shown in Figure 7.13 was based on optimal information transfer and was deemed to be the most effective order to perform the set of design tasks for the plant room and as such could form the basis of a design programme.

Figure 7.13: Design Structure Matrix
7.3.4 Step Three: Producing the Logic Network

The simplest way to produce a design schedule from the Design Structure Matrix was via a logic network which showed diagrammatically the dependency of one task on another. The logic network was divided on a discipline basis so that the cross-discipline interfaces could be highlighted. From the Design Structure Matrix two different logic networks were developed, their difference relating to the manner in which the circuit of coupled tasks were considered.

The first type of logic network, shown in Figure 7.14, is based on the assumption that all tasks in an iterative cycle must be completed before any non-interdependent tasks could begin. Within this set of coupled tasks the validity of the initial estimates must be confirmed by repeating the tasks until the estimates were deemed correct. In Figure 7.13 the circuit of coupled tasks was outlined and all the tasks within the circuit need to be completed before either tasks Q, N or A can begin. The logic network clearly shows which tasks from one discipline have to be performed in a similar time span to tasks from another. Ensuring these tasks were performed at the same time would facilitate cross-discipline information flow, reduce the duration of task and minimise the likelihood of design errors.

Estimating the number of iterations required to solve the circuit of tasks permitted the iterative cycles to be unwrapped. Allocating durations to each task and considering the logic of the network allows a design programme to be drawn up based on the optimised flow of inter- and cross-discipline information.
Chapter 7

The second type of logic network, shown in Figure 7.15 was based on the assumption that the places chosen to make possible tears produced estimated information that did not need validating. In this case the iterative cycles disappeared and the whole design process was a set of tasks that could be performed either in series or parallel. This obviously affected the logic of the network as it was not necessary for tasks, \( N, A \) and \( Q \) to wait until the iteration was completed. From Figure 7.15 it can be seen that task \( N \) could start directly after task \( I \) and \( L \) rather than the alternative shown in Figure 7.14 in which task \( N \) had to wait until task \( M \) was finished.

![Figure 7.15 : Network Developed From DSM without Iteration](image)

The choice of which type of logic network is the most appropriate will vary from project to project and is dependent on many factors, such as the quality of the information estimates and the project duration. A more detailed discussion on the factors that influence the choice logic network is given in the next chapter.

7.3.5 Step Four: The Design Programme

Design Programmes were not produced from the two types of logic network as the procedure was exactly the same as developing a bar chart from a traditional network using CPM and PERT analysis. The logic networks produced are not subject to the failings of traditional network analysis. They have been developed via the DPM and DSMA, where the aspects of choice and iteration have been addressed on a cross-disciplinary basis.

7.3.6 Conclusions on the Prototype Methodology

Before any further work was performed in developing the prototype methodology, its potential to be a useful design management tool was assessed. The prototype clearly showed that the combination of the two separate techniques of data flow diagrams and Design Structure Matrix Analysis produced a method that could both analyse and optimise
the building design process. However, the prototype planning tool was very basic and cumbersome, involving a number of stages, all of which were performed manually. Also the technique was only verified on small design projects of less than 20 tasks and for it to be of use as a practical tool the technique needed to be refined by considering the following points, which were deemed as weaknesses in the prototype methodology:

- the prototype methodology needed to be verified on larger matrices;
- methods for automating the methodology needed to be considered;
- more rigorous techniques for tearing the matrix needed to be developed; and
- more rigorous procedures for developing the design programme from the Design Structure Matrix were required.

The process of refining the methodology was conducted on larger design examples comprising of over 50 design tasks and resulted in the development of the planning technique ADePT, the Analytical Design Planning Technique. One such example is used in the following chapter to explain the refinement process and the methods developed to overcome the perceived weaknesses listed above.

7.4 CONCLUSIONS

The second stage of the research outlined in the methodology in Chapter 3 was to develop a new design planning methodology that could cope with the inherent complexities of building design work. This methodology would integrate the models developed in Phase I so that data on design tasks and their dependencies could be repeatedly extracted from the models to formulate different design programmes for different projects.

This chapter has examined Design Structure Matrix Analysis (DSMA), a technique that was found to be capable of analysing and manipulating design tasks and then ordering them based on the optimal flow of information. This technique dovetailed easily with the Design Process Model and produced a prototype design planning methodology which was examined by using many small design examples of less than 20 tasks. The following four steps constitute the prototype technique developed.

**Step 1**

Use the DPM to choose the tasks required to perform the detail design of the project under consideration and then extract the data on the relevant tasks and their dependencies to formulate a precedence table.
Step 2
Convert the precedence table into a precedence matrix, and then perform Design Structure Matrix Analysis to optimise the information flow and produce an efficient task order.

Step 3
Convert the Design Structure Matrix into a suitable discipline based logic network.

Step 4
Unwrap any iterative cycles and define task durations to produce a design programme.

Several weaknesses were identified in the methodology. To meet the overall aim of the research and if the prototype was to be refined into a practical design management tool to aid the planning of building design these had to be overcome. This development is discussed in the following chapter.
Chapter 8

REFINEMENT OF THE DESIGN PLANNING METHODOLOGY

8.1 INTRODUCTION

The previous chapter discussed a prototype planning methodology formed from a combination of the Design Process Model (DPM), developed in Chapter 6 to generically represent the building design process, and Design Structure Matrix Analysis (DSMA), a technique that can optimise the information flow in a process. The hypothesis that the prototype methodology would have the potential to be a more effective design planning technique than network analysis was tested and verified on a series of small design problems, in which no greater than 20 tasks were considered. The examples proved that the methodology had the potential to organise building design work by effectively coordinating the transfer of information amongst different tasks and disciplines and was therefore worthy of further development.

The subsequent refinement of the prototype methodology produced a technique which we have called ADePT, the Analytical Design Planning Technique. The following sections give an overview of ADePT and then discuss the refinements made to the prototype to produce it.

8.2 SUMMARY OF ADePT

8.2.1 The Procedure

A framework, shown in Figure 8.1, highlights the various stages to ADePT and, although it is ultimately hoped that all the separate procedures can be linked and integrated within a single piece of software, ADePT is sufficiently developed that a mixture of computer and manual analysis allows an effective and workable design programme to be produced. Listed below is a summary of the steps that should be followed to produce a workable representative design programme:

**Step One**

Study the Design Process Model and chose the tasks necessary to complete the detail design stage of the proposed project. Choosing the required design tasks from the generic model automatically generates a set of information dependencies and classifies their relative importance to that task.
Design Process Model

Proprietary Database System

Precedence Charts

Design Structure Matrix Analysis

Discipline Based Logic Network

Task Durations Unwrapping of Iterative Cycles

Project Management Software

Design Schedule

Figure 8.1: Stages within ADePT
Chapter 8

Step Two
The data outputted from the DPM and subsequently represented in a precedence table can then be converted into a precedence matrix. In a fully automated methodology it is hoped that the data from the DPM can be stored in a database so that the precedence tables and matrices can be generated automatically.

Step Three
Partition the precedence matrix to produce the idealised task order based on the optimal flow of design information. This can be done manually for small design projects, but for larger matrices of more than 20 tasks the TERABL algorithm, developed by Steward (1981a) and based on a technique called Path Searching, was found to be an ideal piece of software to perform the analysis. It is ultimately hoped to link the TERABL algorithm to the DPM via a database.

Step Four
From the idealised design task order in the Design Structure Matrix a logic network can be developed on a discipline by discipline basis. This is done by studying each task in turn and tracing its dependency path. The logic network highlights the critical cross-disciplinary interfaces required during the design process as well as all the interdependent tasks that require solving iteratively.

Step Five
The circuits in the logic network can be unravelled by estimating the number of iterations required to solve the interdependent tasks. This is used along with the proposed durations of all the design tasks to convert the idealised design network into a design programme based purely on the optimal flow of design information.

Step Six
External influences such as the construction programme, materials procurement delays and resource restrictions can then be imposed on the idealised design programme and the design tasks re-organised to suit these important criteria.

Step Seven
The task order in the Design Structure Matrix can then be altered to reflect external influences in the adjusted design programme. This allows the impact of performing design in a sub-optimal order to be studied.

The following sections discuss these steps and describe the development of ADePT using a design problem larger than previously tested. This example is used firstly to highlight the
Chapter 8

weaknesses in the prototype methodology and then discuss the solutions put forward. The improvements made in the prototype methodology to produce ADePT have been grouped, and are discussed as follows:-

(i) Improvements made to Design Structure Matrix Analysis. (This covers steps 1 to 3 in the above summary).

(ii) Improvements made in the development of design programmes. (This covers steps 4 and 5 in the above summary).

(iii) Improvements made by imposing internal and external influences on the design programme. (This covers steps 6 and 7 in the above summary).

(iv) Methods for automatically performing Design Structure Matrix Analysis.

Where necessary the design example is used to highlight the inadequacies of the prototype methodology as well as the improved methods developed for ADePT.

8.2.2 The Design Example

The design example chosen to explain the refinements made to the prototype design planning methodology was that of an industrial unit, as shown in Figure 8.2. It comprises a structural steel frame supporting a metal clad flat roof and a vertical metal cladding band, extending one metre below the top of the roof. The lower portion of the external wall is a brick & blockwork cavity wall, supported on strip footings and extending up to the underside of the vertical cladding.

![Figure 8.2 : The Design Example : A Plan of an Industrial Unit](image)

124
### Table 8.1: Precedence Table for Design Example Tasks

<table>
<thead>
<tr>
<th>PROCESS/TASK</th>
<th>INFORMATION REQUIRED</th>
<th>Architects</th>
<th>Structural</th>
<th>Civil</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Ground Floor Plan Dwg</td>
<td>A2, A4, A5, A10, A11, A13</td>
<td>S8</td>
<td>C2</td>
</tr>
<tr>
<td>A2</td>
<td>Elevation Dwg</td>
<td>A1, A3, A5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>Section Dwg Thr Bidg</td>
<td>A1, A2, A5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>External Wall Layout Dwg</td>
<td>A1, A2, A5, A8, A11</td>
<td>S8</td>
<td>C20</td>
</tr>
<tr>
<td>A5</td>
<td>Roof Plan Dwg</td>
<td>A2, A3, A11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>Roof Section Dwg</td>
<td>A3, A5</td>
<td>S8, S10</td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td>External Wall Section Dwg</td>
<td>A3, A4</td>
<td></td>
<td>C11</td>
</tr>
<tr>
<td>A8</td>
<td>Window/Door Details Dwg</td>
<td>A4, A7</td>
<td>S15</td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>Lintel Dwg</td>
<td>A7, A8</td>
<td>S15</td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>Floor Finishes Dwg</td>
<td>A1, A11</td>
<td></td>
<td>C7</td>
</tr>
<tr>
<td>A11</td>
<td>Toilet Layout Dwg</td>
<td>A1, A4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A12</td>
<td>Eaves/Gutter Detail Dwg</td>
<td>A5, A6, A7</td>
<td>S8, S10</td>
<td></td>
</tr>
<tr>
<td>A13</td>
<td>Site Layout Dwg</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Civil</th>
<th>Structural</th>
<th>Architects</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Pit Calcs</td>
<td>C4, C5, C2, C3</td>
</tr>
<tr>
<td>C2</td>
<td>Pit GA Dwg</td>
<td>C1</td>
</tr>
<tr>
<td>C3</td>
<td>Pit Reinforcement Dwg</td>
<td>C1, C2, C7</td>
</tr>
<tr>
<td>C4</td>
<td>Gnd Floor Prelim Calcs</td>
<td>C1</td>
</tr>
<tr>
<td>C5</td>
<td>Gnd Bearing Calcs</td>
<td>C4, C2, C3</td>
</tr>
<tr>
<td>C6</td>
<td>Gnd Floor Plan Dwg</td>
<td>C2, C5, C15, C16, C18, C20, C21</td>
</tr>
<tr>
<td>C7</td>
<td>Gnd Floor Sections Dwg</td>
<td>C5</td>
</tr>
<tr>
<td>C8</td>
<td>Found Prelim Calcs</td>
<td>C5, C7, C23</td>
</tr>
<tr>
<td>C9</td>
<td>Strip Footing Calcs</td>
<td>C6, C2, C23</td>
</tr>
<tr>
<td>C10</td>
<td>Foundation Layout Dwg</td>
<td>C9, C12</td>
</tr>
<tr>
<td>C11</td>
<td>Strip Footing Section Dwg</td>
<td>C9, C10, C13</td>
</tr>
<tr>
<td>C12</td>
<td>Pad Calcs</td>
<td>C6, C2, C23</td>
</tr>
<tr>
<td>C13</td>
<td>Pad GA Dwg</td>
<td>C10, C12</td>
</tr>
<tr>
<td>C14</td>
<td>Pad Reinforcement Dwg</td>
<td>C12, C13</td>
</tr>
<tr>
<td>C15</td>
<td>Storm Drain Calcs</td>
<td>C2, C10, C11, C13, C17, C21, C22</td>
</tr>
<tr>
<td>C16</td>
<td>Storm Layout Dwg</td>
<td>C6, C22</td>
</tr>
<tr>
<td>C17</td>
<td>Foul Drainage Calcs</td>
<td>C2, C10, C11, C13, C15, C21, C22</td>
</tr>
<tr>
<td>C18</td>
<td>Foul Layout Dwg</td>
<td>C17, C22</td>
</tr>
<tr>
<td>C19</td>
<td>Drainage Details Dwg</td>
<td>C1, C13, C15, C16, C17, C19</td>
</tr>
<tr>
<td>C20</td>
<td>Column Casing Dwg</td>
<td>C6, C7, C13, C14</td>
</tr>
<tr>
<td>C21</td>
<td>Site Survey Dwg</td>
<td>C7</td>
</tr>
<tr>
<td>C22</td>
<td>UIG Co-Ord Services Dwg</td>
<td>C10, C16, C18, C21</td>
</tr>
<tr>
<td>C23</td>
<td>Site Investigation Report</td>
<td>C21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structural</th>
<th>Architects</th>
<th>Civil</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Wind Load Calcs</td>
<td>A2, A3, A13</td>
</tr>
<tr>
<td>S2</td>
<td>Vertical Load Calcs</td>
<td>S6, S12</td>
</tr>
<tr>
<td>S3</td>
<td>Load case Calcs</td>
<td>S152</td>
</tr>
<tr>
<td>S4</td>
<td>Structural Philosophy</td>
<td>S3, 55</td>
</tr>
<tr>
<td>S5</td>
<td>Prelim Calcs On Constn Method</td>
<td>S3, 54</td>
</tr>
<tr>
<td>S6</td>
<td>Structural Layout Calcs</td>
<td>S3, 54, 55</td>
</tr>
<tr>
<td>S7</td>
<td>Member Sizing Calcs</td>
<td>S3, 54, 56</td>
</tr>
<tr>
<td>S8</td>
<td>Swt Pan At Grd Level Dwg</td>
<td>S6, 57</td>
</tr>
<tr>
<td>S9</td>
<td>Swt Pan At Roof Level Dwg</td>
<td>S6, 57, S8, S13, S14</td>
</tr>
<tr>
<td>S10</td>
<td>Swt Sections Dwg</td>
<td>S8, 59</td>
</tr>
<tr>
<td>S11</td>
<td>Base Plate Dwg</td>
<td>S7, S8, S10</td>
</tr>
<tr>
<td>S12</td>
<td>Liftng Beam Calcs</td>
<td>S6</td>
</tr>
<tr>
<td>S13</td>
<td>Liftng Beam Dwg</td>
<td>S8, S12</td>
</tr>
<tr>
<td>S14</td>
<td>Purin Dwg</td>
<td>S8, 59</td>
</tr>
<tr>
<td>S15</td>
<td>Door/Window Swt Dwg</td>
<td>S10, S14</td>
</tr>
</tbody>
</table>

Table 8.1: Precedence Table for Design Example Tasks
Chapter 8

The data flow diagrams for the work involved in this project were again drawn up before the DPM hierarchy was finalised and as the tasks chosen for this project varied so minimally with the final DPM, neither the analysis nor the results were affected. In total the model had 51 functional primitive tasks: 13 architectural, 23 civil and 15 structural. Neither the complete model hierarchy, nor the DFDs from the model have been included in this thesis as they were both very similar to the finalised DPM shown in Appendices I and III. It should be noted that an input from the electrical and mechanical services disciplines was not deemed necessary as the example was only used to refine the methodology and it was felt that their inclusion would only complicate the refinement process.

The precedence table for the 51 tasks is shown in Table 8.1: A1-A13 are the architectural tasks; C21-C23 the civil tasks and S1-S15 the structural tasks. This was extracted directly from the DFDs of this project.

8.3 IMPROVEMENTS TO DESIGN STRUCTURE MATRIX ANALYSIS

8.3.1 Link Between the DPM and DSMA
The manual manipulation and transfer of data on tasks and their dependencies, from the DPM, via a precedence table, into the precedence matrix was an exacting and time consuming process, which is ideally suited to automation. Although outside of the direct scope of this research, a conceptual framework was developed to help achieve the goal of automatically transferring data from the DPM to the precedence matrix. This is shown diagrammatically in Figure 8.3.

![Diagram](image)

Figure 8.3 : Data Transfer Between the DPM and DSMA

126
Chapter 8

The proposed framework requires the design tasks to be chosen from the DPM and then all the relevant data on the tasks and their information dependencies are stored in a neutral database format. This data can then automatically be accessed, converted into a precedence table and used to generate the precedence matrix. It should be noted that this conceptual framework was also developed to provide a suitable link between the DPM and a simulation package used in the work of Hassan [1994], who in a parallel research initiative is using the DPM to simulate the effects of change on the building design process.

At the end of this three year research project very little of this automation had taken place; the whole methodology required more detailed verification and validation before the expense of developing software could have been justified. However, the CASE tool used to construct the DPM was specifically chosen because it was one of the few commercially available tools that could produce files in the universally accepted database dBase III format.

8.3.2 DSMA : The Problem

The precedence matrix, shown in Figure 8.4, was developed manually with the aid of a computer drawing package from the precedence table in Table 8.1. The partitioning of this particular example was again done by hand, although methods and algorithms for automating DSMA are used and evaluated later in this chapter. It can be seen from the partitioned matrix, shown in Figure 8.5, that representing and analysing a large design problem, with a large number of tasks, produced a new set of problems not encountered in the examples considered in the development of the prototype. Of the 51 tasks, 44 tasks were found to be interdependent, forming a large circuit of coupled tasks indicated by the shaded block. In theory, if the tasks were performed in the order shown, task S15 would have to be completed before the estimates in the circuit could be validated.

Although building design is an iterative process, iterative cycles of the size indicated by this analysis very rarely exist because designers intuitively take steps to reduce design iteration. Therefore, the matrix shown in Figure 8.5 is of limited use, both as a representation of the building design process and as the basis of a co-ordinated design programme. Tearing a circuit of this size would also be time consuming and probably involve several attempts at identifying 'tearable' pieces of information.

One of the objectives of the refinement process was to develop more rigorous procedures for tearing blocks in the partitioned matrix. This has been achieved by studying the various affects information has on a task and then classifying the flows accordingly.
Figure 8.4: Precedence Matrix for Design Example
Figure 8.5 : Partitioned Matrix for Design Example
8.3.3 Classifying Information Dependencies

A Design Structure Matrix, as presented by Steward, only represents strict information dependencies which do not indicate the importance of information to a task. In the example discussed above, and in other complex design problems, partitioning based solely on these dependencies does not greatly clarify or simplify the structure of the design process. This is because the matrices often contain many weak dependencies that appear to make design tasks highly interdependent, giving rise to large blocks of coupled tasks. However, it is reasonable to assume that some tasks are more reliant on certain pieces of information than others and to aid the tearing process it was necessary to consider the relative importance of these dependencies.

The relative importance of each dependency was addressed by classifying each information flow to establish its relative importance to a task, when compared against other information flows. This permitted the most important information dependencies for each task to be identified and the subsequent tearing of each circuit ensured the most important information flows were below, or closer to, the diagonal than the weaker dependencies. This was found to be more representative of the building design process and would converge more quickly towards a solution. Eppinger and his team (1992 & Smith and Eppinger 1992), in their work in production design, have addressed the problem of relative importance by defining an 'importance ratio' for each information dependency in terms of a percentage (0 to 100%); the larger the number the stronger the dependence. Evaluating the dependencies, as Eppinger himself admits, is a very subjective process as the views and perceptions of the importance of information often changed from engineer to engineer and usually from project to project. It was felt the application of a percentage classification was unsuitable and unjustified because the procedure was vague and unstructured. This led to the development of a different classification system which took a more practical approach to the problem in which information flows are classified into one of three groups: Class A, B & C; the definitions of which are below.

**Class A Information:**
It is absolutely essential to a task that class A information be made available prior to its commencement.

**Class B Information:**
It is not essential to a task that class B information be made available prior to its commencement but it would be preferable.

**Class C Information:**
It is not essential to a task that class C information be made available prior to its commencement.
Three factors were found to affect the relative importance of an information flow to a task: the dependence and sensitivity of a task and the ease of estimating or 'estimatability' of the information.

**Properties of the Task**

*Dependence:* The dependence of a task is defined as how much the task relies on a certain piece of information. The dependence can either be *strong* or *weak*.

*Sensitivity:* The sensitivity of a task is defined by the affect small changes in information has on the output of the task. Tasks are either *sensitive* to small information changes or *insensitive*.

**Properties of the Information**

*Estimatability:* The *estimatability* of information is a measure of ease with which information can accurately be estimated within defined limits. Information is either *estimatable* or *unestimatable*. This definition allows for 'fuzzy' information to be incorporated into our classification. If the task can proceed on 'fuzzy' information, i.e. information that will be updated at a later date, the information is deemed 'estimatable'.

<table>
<thead>
<tr>
<th>Information Flow</th>
<th>Task is</th>
<th>Task is</th>
<th>Information is</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLASS A</td>
<td>INCREASINGLY DEPENDENT</td>
<td>INCREASINGLY SENSITIVE</td>
<td>INCREASINGLY ESTIMATABLE</td>
</tr>
<tr>
<td>CLASS B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 8.6: Contrasting Influences on Information Classification*

These three properties have contrasting influences on classification of information flows which are shown diagrammatically in Figure 8.6. Difficulties were found to arise when trying to decide which was the more dominant property for each information flow. For example, if a task is strongly dependent on a piece of information but it can be estimated to the required degree of accuracy, is it class A, B or C information? A flow chart was
developed to guide engineers and designers in the classification of information flows during the verification process. The flow chart, shown in Figure 8.7 is used by determining whether or not a task is highly dependent on a piece of information, and then following the shade of arrow relevant to the answer through the remainder of the chart to identify the relevant classification.

**Figure 8.7: Information Classification Flow Chart**

8.3.4 **Examples of varying information Classifications**
To clarify the differences between the various types of information classifications, the following examples have been drawn up:
Chapter 8

Class A: \textit{Strong Dependence, Unestimatable}. (Figure 8.8)
It would be impossible to guess the size, the position, the depth and the type of foundations when producing a detailed foundation GA drawing. This information must come some sort of calculation process and therefore is strongly dependent and 'unestimatable'.

![Figure 8.8: Foundation GA Drawing DFD](image)

Class A: \textit{Weak Dependence, Sensitive, Unestimatable}. (Figure 8.9)
The production of the civil engineering ground floor plan is not highly dependent on the position of rainwater pipes as they may or may not penetrate the floor slab. However, if they do, their position and perhaps their shape would be 'unestimatable' by the civil engineer. The task is sensitive to this information because changes in the position of these down pipes might affect the positioning of floor joints and the reinforcement in the slab.

![Figure 8.9: Ground Floor GA Drawing DFD](image)

Class B: \textit{Strong Dependence, Sensitive, Estimatable}. (Figure 8.10)
An electrical engineer will require an estimate of the electrical loadings produced by any pumps within the building. This information will be required at an early stage of the design process at which time the mechanical engineer will only have 'ball park' estimates for the size of his pumps. This figure will be passed on to the electrical engineer in the full knowledge that it will be updated later.
Class B: Weak Dependence, Sensitive, Estimatable. (Figure 8.11)
The production of the storm drainage layout is not strongly dependent on the position of the foundations. Where there may be the possibility of a clash the drainage layout becomes sensitive to the foundation positions. However, an accurate enough estimate may be made of the proximity of the foundations to ensure the drainage runs do not clash with the foundations.

Class C: Strong Dependence,Insensitive, Estimatable. (Figure 8.12)
The calculation of the total vertical loads is strongly dependent on all loads that may contribute. The self weight loads from the frame itself are usually a very small percentage of the total load and are 'estimatable'. If there is an error in the estimate the total vertical loads will not be greatly affected, i.e. the task is insensitive.
Class C: *Weak Dependence, Insensitive.*

Two or more drawings might portray the same piece of required information. If that piece of information, with the necessary classification, can be obtained from one of those sources it makes the information from the other source redundant and therefore the task is not dependent on that information being available.

8.3.5 Applying Information Classifications

Using the flow chart in Figure 8.7 in conjunction with the DFDs in the DPM, each information flow in the design of industrial unit was classified either A, B or C and was entered into the precedence matrix, see Figure 8.13. Partitioning the matrix with the classified dependencies ensured the most important dependencies (class A and to a lesser extent class B) were forced below or as close to the diagonal as possible, whereas class C information, being of less relative importance tended to exist mainly above the leading diagonal. These properties can be seen in Figure 8.14, which is the result of partitioning the precedence matrix in Figure 8.13. This matrix, called the Design Structure Matrix (DSM), shows the majority of class A and B information below the diagonal, whereas class C information, in the main, occurs above. It should be noted however, that partitioning matrices with classified dependencies still aims to ensure as many of the dependencies are positioned below or as close to the diagonal as possible.

It was found that classifying all the dependencies in the precedence matrix (Figure 8.13) and then partitioning, gave the same results as partitioning the initial precedence matrix containing all the dependencies (Figure 8.4), then classifying only those dependencies in the circuits and then re-partitioning the blocks. Therefore, classifying each information dependency in the precedence matrix and then partitioning the Design Structure Matrix could be achieved in one step. Classifying information dependencies in such a rigorous manner obviated the need to undergo the separate processes of partitioning and tearing as in traditional DSMA. It is envisaged that ultimately a fully automated version of ADePT would store the classification for each information flow in the relevant data dictionary entry in the DPM, thus allowing a classified precedence matrix to be generated automatically.
Figure 8.13: Classified Precedence Matrix for Design Example
Figure 8.14: Design Structure Matrix for Design Example
8.3.6 Robustness of Classified Information Dependencies

There were two indicators that justified the use of the information classification system:

- practising designers' opinions of the classified matrix representation; and
- the blocks of coupled tasks correlating with site co-ordination problems.

It was assumed that the least important information dependencies, i.e. class C information, did not require validation during the design process and therefore did not give rise to iterative cycles. This meant the initial circuit of tasks diagnosed in Figure 8.5 could be reduced into a number of much smaller blocks, shown in the Design Structure Matrix in Figure 8.14 as those blocks only containing either class A or B information. This was a more compact, less coupled representation of the building design process. In the majority of cases only class B information required estimating with only two class A dependencies being above the diagonal; the partitioning process not only minimised the size of the blocks, but ensured the most suitable places to make estimates were placed above the line. Designers shown both Figures 8.5 and 8.14 readily agreed that the smaller blocks of iteration was a closer representation of what they believed happened in the design of a building.

One particular feature of these smaller blocks that interested both designers and design managers was that certain blocks showed a close correlation to problems that blight both the building design and construction processes. Analysis of the large block in Figure 8.14 showed that all the tasks required in detailing the junctions between the structural frame, the secondary steelwork, the external and internal skins and the detailing of the door and window openings needed to be highly co-ordinated. This part of the building often causes problems during the construction process, mainly as a result of poor co-ordination between disciplines when producing working or detailed drawings. A lack of co-ordination often results in clashes and omissions which not only slows the construction programme but results in re-design. Studying the matrix can lead to a greater insight into the cross-discipline co-ordination needed for particular sections of the building design process.

Large blocks of coupled cross-disciplinary tasks raises the question of whether the traditional division of design teams into separate disciplines is the most suitable way of solving these processes; in fact because different disciplines are involved, it is more likely that co-ordination problems will arise. Analysis of the blocks in the DSM seems to suggest it may be more efficient for certain parts of the building to be designed by teams dedicated to the functional requirements of the building rather than the domain based aspects.
Further confirmation of the validity of our approach to the classification of information flows came from an internal MIT working paper by McCord and Eppinger (1993). They had independently developed a similar method of classification which seemed to have superseded their original percentile 'importance ratio' and used in its place three levels of dependence: high, medium and low, to differentiate between information flows. However, their method doesn't appear to be as rigorous as the method presented here or consider the separate effects of dependence, sensitivity and 'estimatability'.

The classification of information is a subjective process, which can lead to possible inaccuracies in the formation of the precedence matrix: one engineer's view on a classification may be different to another because of the way they perform a task or as a result of their own personal design experience. Although it was not possible to produce flawless definitions of task sensitivity, or how easily information can be estimated the classifications proposed are sufficiently rigorous to produce a fair representation of the building design process.

### 8.4 IMPROVEMENTS TO DESIGN PROGRAMMES

#### 8.4.1 Design Programmes: The Problem

Another perceived weakness of the prototype methodology discussed in the previous chapter was that the techniques and procedures for developing both logic networks and subsequent design programmes were vague.

The main problem encountered when converting the Design Structure Matrix to a logic network was how to represent the circuits of coupled tasks, reduced and simplified by the introduction of information dependency classifications. Figure 8.15 shows three circuits, each with different information dependencies above the diagonal; from left to right, the importance of the estimated information decreases and therefore if these blocks were to be solved in a truly iterative fashion with each estimate being gradually refined, the matrix on the left hand side would need to performed more times to validate the estimates made. The class of information above the diagonal affects the number of times the circuit must be worked and hence influences the ultimate design programme produced from the logic network. The following sections set out a more rigorous approach to develop effective design schedules from the fully ordered Design Structure Matrix.
8.4.2 Development of Logic Networks

Developing the argument put forward in the previous chapter in Section 7.3.4, on which iterative cycles should be included and which ignored, when forming the logic network, a decision was made to include only those circuits containing class A and B information; all circuits containing solely class C information were ignored. In the Design Structure Matrix shown in Figure 8.14, all class C information above the diagonal was ignored in the formation of the iterative loops, although strictly speaking tasks A1 through to C6 would be in a single circuit if all class C information were deemed important enough to validate.

The logic network developed from the Design Structure Matrix is shown in Figure 8.16. The circuits of coupled tasks are also highlighted on the logic network, clearly showing the importance of performing certain tasks by different disciplines within the same time span to facilitate the co-ordination of cross-disciplinary information. Two different types of circuits are shown: the darker shading indicates blocks that contain some class A information estimates and the lighter shading indicates blocks without class A information estimates. A network such as this clearly shows the exact interfaces and co-ordination required between different disciplines; this property of the design process is often very difficult to quantify when traditional techniques are used.

The network was formed, from left to right, by analysing each task in the Design Structure Matrix to assess its dependency path. The procedure followed to form the initial sections of the logic network is outlined below:-

(i) Task A13, the first task in the Design Structure Matrix, was placed accordingly in the architectural section of the network.
Figure 8.16: Logic Network Developed from Design Structure Matrix
Figure 8.17: Simplified Logic Network Developed from Design Structure Matrix
(ii) Tasks C21 and C23, the next two tasks in the matrix were both dependent on A13. C23 was also dependent on C21 and therefore could be placed directly after C21 in the network. A direct link was not shown between C23 and A13 as a dependency path existed and could be traced via task C21.

(iii) The next tasks in the matrix were the circuit of tasks A1, A2, A3 and A5; the dark shading indicating that at least one class A information estimate was required. The only task this circuit was dependent upon was task A13 and therefore the block was placed directly after this task in the network.

(iv) The next two tasks in the DSM were S1 and S2 and as they were not dependent upon each other they could be performed in parallel. The dependency path of each task was clearly annotated in the network to indicate which of the tasks in the block A1, A2, A3 and A5, S1 and S2 were dependent on. Task A13 was not indicated as a dependency to S1 as it was clear the block was dependent upon it, and therefore it was not necessary to annotate it twice.

The remaining tasks in the Design Structure Matrix have been placed in the logic network following the procedure described above.

A simplified version of the logic network is shown in Figure 8.17, in which the total number of dependency paths have been reduced by collapsing the circuits of tasks. This version is perhaps more readable than the full network, whilst still highlighting the important cross-disciplinary information flow.

8.4.3 Linking the Logic Network to a Project Management Package

The next step would be to link the logic network into a standard project management package to produce full design programmes. Allocating tasks durations and iterations and applying CPM or PERT analysis would allow the development of design programmes using software already familiar to planners and project managers. Specific packages and the mechanisms to link to the network were not examined as part of this research, and it is appreciated that the package chosen may affect the way in which the logic network is developed and represented. The networks shown previously were developed principally for the manual production of design programmes.

Analysing the logic network, with PERT or CPM, would be almost identical to analysing a network drawn up using traditional network analysis except for one major difference: the circuits of coupled tasks have to be unravelled first. All tasks, in each circuit, need to be performed a certain number of times to produce a suitable solution; the exact number of times depends on the number and class of information estimates required. Although
choosing the number of iterations required would a subjective process and undoubtedly would differ from circuit to circuit, the different shadings of the blocks in Figure 8.16 would aid the planner to a certain extent, the darker shading indicating at least one class A estimate must be made. It is also safe to assume that each subsequent iteration of the circuit would require less time; again choosing the predicted duration of each task, for each iteration would be a decision the design planner would need to make before a design programme could be drawn up. For instance, consider the circuits of tasks C4, C1 and C5. If the planner believed that this block would take three iterations to solve and each task would take initially two days to perform, reducing by 50% for each subsequent iteration, the part of the design schedule shown in Figure 8.18 could be drawn up.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.18 : Unravelled Section of Logic Network

The remainder of the circuits can be unravelled in the same way and the process of adding tasks durations, dependent on the available resources, could turn the network into a design programme. As the design of the industrial unit was purely an example, used to refine the prototype, a true design programme based on task durations could not be drawn up. However, Figure 8.19 shows an indicative design programme, developed from the logic network that has been based on assumed task durations and only one architect, one civil and one structural engineer working on the project.

8.5 INCORPORATING INTERNAL AND EXTERNAL INFLUENCES

8.5.1 Internal and External Influences on Design : The Problem
The design programmes produced from the analysis of the DPM with the DSMA displays an idealised task order, based on purely on the optimum flow of design information; this very insular view takes no account of other influences. In contrast, traditional design planning is often performed by starting with the external influences, such as the construction programme, and scheduling design tasks to co-ordinate with the 'out to tender' or 'start on site' dates. However, this approach gives no consideration to the impact of performing design sub-optimally.
Figure 8.19: Design Programme Developed for Design Example
Chapter 8

The idealised task order in the Design Structure Matrix shown in Figure 8.14 produced an *unconstrained* design programme, which was evolved by considering purely design tasks and the transfer of design information. For many reasons this would not be workable: in scheme design the projected completion date and the influence of the client have the potential to disrupt performing design tasks in the optimal order; in detail design there are many influences, both internal and external to the design process that compromise the ideal task order. A *constrained* design programme, adapted to accommodate the influences of the construction and management processes can be studied in conjunction with the DSM and the impact of performing design in a sub-optimal order determined. Discussed in the following sections are the typical internal and external influences on detail design, which is followed by a specific example of how ADePT can be used to analyse these effects.

8.5.2 Internal Influences on the Design Process
The unconstrained design programme is based solely on a combination of the logic networks and times envisaged by a planner for a certain number of designers to perform each task. However, an essential part of a planner's job is the efficient use and control of human resources. The availability of resources, a finite entity, can drastically affect an optimised design programme. In Figure 8.19 large gaps are shown between various tasks indicating that designers may be idle for certain periods. This is clearly not practical and resources need to be levelled and smoothed to ensure that a steady demand is achieved rather than a widely fluctuating demand containing peaks and troughs. Also most design organisations are committed to more than one project at any one time and it is likely that designers switch frequently between projects. This makes the co-ordination of resources even more difficult and all the more likely to affect the unconstrained design programme.

8.5.3 External Influences on the Design Process
The majority of external influences on the design process are imposed by the client to ensure the building is constructed as quickly as possible. The three major influences at the construction interface are discussed below:

*Materials Procurement*
In the modern economic climate, where the emphasis is on speed of construction and where the procurement of certain materials may take a long while, it is becoming increasingly important to place orders early (during the design process) so that the construction process is not delayed. An example of this was the extreme demand for fabricated structural steel work in the mid 1980's, where large delays in its production made the ordering of the steel work a critical item in the success of the project. The use of fast-tracking was also winning favour with many clients at this time and therefore orders
had to be placed for steel work before design was complete. To finish an important package, such as structural steel work, at an early stage of design would mean that an unconstrained design programme would need to be compromised.

**Work Package Design**

Design programmes are usually adjusted to suit the needs of construction. The current practice of employing sub-contractors to construct discrete sections of the project has led to designs produced and released to site in a series of units or work packages. For example a concrete works package is common to most projects and contains foundations, ground beams, retaining walls. Imposing a date on the design team when these packages should be released to site may force some of these elements to be designed in a sub-optimal order simply because they have been grouped together with other elements solely to benefit the construction process.

**Procurement Route**

The start of the construction process, in relation to completion of the design process, is dictated by the procurement route. Under a traditional contractual arrangement design is almost complete before the construction process begins and therefore an unconstrained design programme is largely unaffected by procurement. However, many modern projects are now conducted by overlapping to some extent the construction and the design phases. Therefore, the order in which the information is required by the construction process changes the order in which design tasks need to be completed.

8.5.4 Using DSMA to Study the Impact of Construction Influences.

The impact of performing design to suit the construction process, rather than to suit the design process itself, has been difficult to appreciate and quantify in terms of project duration and cost. Design Structure Matrix Analysis, as well as being capable of optimising task orders can be used to study the consequences of performing design sub-optimally to complement a constrained rather than unconstrained design programme.

Consider the design of the foundations in the industrial unit example. Ideally this package would be designed towards the end of the design process when all the relevant information on the design loads and interfaces become available. This is borne out in the Design Structure Matrix in Figure 8.14, in which the foundation drawing tasks C10, C11, C13 and C14 are scheduled towards the end of the optimal task order. However, in many fast-track projects the construction of the foundations is a critical item and often occurs before the foundations would traditionally be designed. The impact of this interference on the optimal design order can be studied by examining the unconstrained design programme and the Design Structure Matrix.
Figure 8.20: Impact of Fast Tracking Foundation Design
For example, if the matrix (Figure 8.14) and the associated unconstrained design programme, reveal that the foundations had to be designed earlier (to fit in with the construction programme), e.g. prior to the design of task S12, new positions for the foundation tasks could be found in the Design Structure Matrix. This would involve moving all the tasks involved in the design of the foundations up the matrix so that they were ordered in front of task S12. Figure 8.20 shows the implications on the information flow of moving these tasks up the matrix to an earlier stage. It can be seen that two class A information dependencies now appear above the diagonal. Although this information would have to be estimated it cannot form part of an iterative cycle because these tasks need to be completed quickly so that construction can begin. Any errors in the estimates will have costly consequences that may involve the reconstruction of foundations plus subsequent delays in the construction programme. To ensure these problems do not arise the design estimates must be conservative, giving adequate flexibility to cope with any unexpected developments in remaining design work. This obviously has an economic implication which the designer will want to convey to the client.

Forcing the position of tasks within the matrix may also affect the order of the tasks below the relocated tasks, especially if these lower tasks are heavily dependant upon information from them. In this example it would not be necessary to re-order below the repositioned foundation tasks, because the remaining order can be seen to be still optimal. However, if a task is removed from a large iterative block and forced up the matrix it will be necessary to re-partition all the tasks below the re-positioned one, as the circuit of coupled tasks could be affected significantly by the removal of a single task.

Each internal and external influence on the unconstrained design programme can be studied in turn by going through the loop of checking the impact of repositioning the affected task in the DSM. ADePT has not only the capability of producing effective and workable design programmes but allows the impact of performing design tasks in a sub-optimal order to be studied.

8.6 OTHER APPLICATIONS OF ADePT

8.6.1 Studying the Effect of Change
A design process, however well planned, is subject to variations and change, whether initiated by the design team or by the client. The management of variations is becoming increasingly important in respect to consultants' profit margins. However, predicting, costing and monitoring the effect of change is an extremely difficult process, hindered by the lack of appropriate tools available to the design manager.
The Design Structure Matrix can be used in a simple manner to predict the affects variations have on the design process. A study of the DSM can show the propagation of effects which occur as a consequence of a change to a design task. In a matrix a study of the dependencies in a row shows what information is required to perform that task, whereas going down a column shows what information is required from that task, i.e. what other tasks are dependent of the results of the task in question. This latter property can be used to study which tasks are affected when a change is initiated in a single task. For instance, in Figure 8.14 if it is assumed that task A11 changes, it can be seen by going down the relevant column in the DSM that the tasks A1, A5, A4, and A10 are dependent on task A11 and therefore it is possible that these four tasks might need altering as well.

The class of an information dependency obviously influences the likelihood of a change to one task affecting another: the more important the information to a task, the more likely the change will have a knock on effect. If, to be consistent with previous analysis, class C information is ignored, the dependency checking becomes easier. This simplifies the dependencies of task A11 to just tasks A4 and A10; these dependencies are called primary dependencies. Further changes may occur because it is reasonable to assume if the tasks A4 and A10 are altered, resulting changes may occur in the tasks requiring information from them. It is therefore necessary to determine what other tasks may be affected from these secondary dependencies; in this case a change to task A10 might affect task A1 and C4. Task A4 is also part of an iterative loop and any change in A4 is likely to affect other tasks in the same loop. This process of tracing the knock on effects of secondary and iterative dependencies could carry on almost indefinitely, but the likely effect of a variation will diminish with each check performed.

While it was not possible to fully explore the full consequences and effectiveness of tracing variations through the DSM a preliminary procedure was developed to produce a practical checklist which can be used by a design manager as a guide to the primary and secondary effects of change. The procedure is as follows:-

**Step One** : Determine the primary dependencies, ignoring all the type C information.

**Step Two** : Use the primary dependencies to examine the secondary and iterative dependencies including the type C information.

If the primary, secondary and iterative dependencies were traced for a change to task A11, a list as shown in Table 8.2 could be drawn up. It lists the primary dependencies, which ignore the type C information and then uses them to determine the secondary and iterative dependencies. It can be seen that class C information dependencies have been included because a task may appear more than once on the variation list and therefore the cumulative effect of two or more insignificant type C information changes may result in a
noticeable variation. It is suggested to use this checklist in the most effective way it is
drawn up by an experienced design manager, who can make an informed judgement about
the cumulative effects of the changes to secondary and iterative dependencies.

<table>
<thead>
<tr>
<th>Altered Task</th>
<th>A11 - Toilet Floor Layout Dwgs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Dependencies</strong></td>
<td><strong>Secondary Dependencies</strong></td>
</tr>
<tr>
<td>A4  External Wall Layout Dwgs (A)</td>
<td>A1  General Gnd Floor Layout Dwgs (A)</td>
</tr>
<tr>
<td></td>
<td>C6  Gnd Floor Layout Dwgs (A)</td>
</tr>
<tr>
<td></td>
<td>C8  Foundation Preliminary Calcs (A)</td>
</tr>
<tr>
<td></td>
<td>C10  Foundation Layout Dwgs (C)</td>
</tr>
<tr>
<td></td>
<td>C20  Column Casing Dwgs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iterative Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>A6  Roof Sections Dwgs</td>
</tr>
<tr>
<td>A7  External Wall Section Dwgs</td>
</tr>
<tr>
<td>A8  Window &amp; Door Details Dwgs</td>
</tr>
<tr>
<td>S9  Gnd Floor Level SWK Dwgs</td>
</tr>
<tr>
<td>S9  Roof Level SWK Dwgs</td>
</tr>
<tr>
<td>S10  SWK Sections Dwgs</td>
</tr>
<tr>
<td>S13  Lifting Beam Calcs</td>
</tr>
<tr>
<td>S14  Lifting Beam Dwgs</td>
</tr>
<tr>
<td>S15  Window &amp; Door SWK Dwgs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Dependencies</th>
<th>Secondary Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>A10  Floor Finishes Dwgs (A)</td>
<td>A1  General Gnd Floor Layout Dwgs (C)</td>
</tr>
<tr>
<td></td>
<td>C4  Gnd Floor Preliminary Calcs (C)</td>
</tr>
</tbody>
</table>

Table 8.2 : Variation Effect Trace.

It is believed that the approach put forward, although in its infancy has the potential to
become a useful technique for helping a design manager to predict the implications of
variations within the design process.

8.6.2 Using the Logic Networks to Programme Design Reviews

Design reviews, used to verify and co-ordinate the proceeding design work, usually take
one of two forms: external design reviews, involving the client, and internal design team
reviews. Traditionally design reviews are scheduled arbitrarily at regular intervals
throughout the design process, usually fortnightly or monthly. However, scheduling
reviews in such a fashion takes no account of the structure of the design process or the type
of design work occurring between two consecutive meetings. A study of the logic network
in Figure 8.16 suggests that design reviews could be programmed to complement the
structure of the design process rather than be scheduled arbitrarily.

External Design Reviews

External design reviews involving the client are often used as a means of checking and
approving the progression of the design. One result of such meetings is that the design
team has to adjust or refine its solutions. This gives rise to a repetitive cycle, (see Section
6.2.4) where certain design tasks need to be re-worked to suit the client's demands and requirements. This research suggests that external design reviews should be performed directly before large circuits of coupled tasks are performed, e.g. those in the large block indicated on Figure 8.16. Performing these tasks with unapproved information increases the chance of having to re-work or adjust the solution of one of tasks in the block. If this happens, all the tasks in the circuit may need to be adjusted because of their interdependence. Conversely, ensuring that all the information required for the coupled tasks is available, correct and approved by the client at this stage should reduce the chances of having to re-design.

**Internal Design Reviews**

Internal design reviews usually involve the members of the design team, their purpose being to review and co-ordinate the design solutions produced by the various disciplines. This research clearly indicates that internal reviews should be programmed to occur directly after a large block of iterative tasks have been completed to ensure the co-ordination and integration of highly coupled tasks have been achieved successfully.

### 8.7 PARTITIONING ALGORITHMS

It was found that manipulating and partitioning small matrices of up to 15-20 tasks could be managed successfully by hand. However, for larger design problems with more tasks and dependencies (such as the example used in this chapter) this became impractical without a computer solution. Two algorithms, capable of partitioning large matrices, were procured and their performance and philosophies are examined and compared below.

#### 8.7.1 Partitioning Theory

The general theory of partitioning is to systematically determine the earliest possible time a task can be performed in the design process. Both partitioning algorithms use the same basic steps which are repeated in a loop until the matrix is partitioned; they were:

**Step 1:** Schedule non-interdependent tasks which are tasks not dependent on information, and tasks not providing information.

**Step 2:** Identify tasks in loops (circuits) of coupled tasks.

Step 1 was performed identically for both algorithms, whereas step 2 was be carried out either by Boolean algebra or by Path Searching.
8.7.2 Step 1: Scheduling Non-Interdependent Tasks

Consider the system of tasks and information flows shown in Figure 8.21(a) and the matrix drawn up to represent this system in Figure 8.21(b). All non-interdependent tasks in this matrix have either an empty row or an empty column: tasks with empty rows are not dependent on any information and can be scheduled at the earliest possible time; tasks with empty columns do not provide information to other tasks and therefore can be scheduled at the latest possible point. Once identified these tasks can be ignored, along with their corresponding information flows, for the remainder of the analysis. This may or may not highlight all the non-interdependent tasks and this procedure continues until all the non-interdependent tasks are discovered.

From Figure 8.21(b) task C can be scheduled last as it has an empty column, and task M can be scheduled first as it has an empty row. Ignoring both these tasks and their information dependencies means that task G now has an empty row and therefore, can be scheduled directly after task M. After the removal of task G no further tasks have empty rows nor columns, meaning that the remaining tasks are either in circuits or are dependent on information from a task in a circuit.

8.7.3 Step 2: The Identification of Loops in the Matrix

Two methods, Boolean algebra and path searching, were examined to identify and order loops of the remaining tasks in the original matrix; in the example used above these tasks are A, B, D, E, F and H to L. The following sections describe and compare the two methods.
Loop Identification by Boolean Algebra

Boolean algebra is a well established mathematical technique for manipulating matrices (Steward 1962, Ledet & Himmelblau 1970). The initial step in the identification of loops using this technique is to formulate the Adjacency Matrix, 'A' of all the unscheduled tasks, in which all information dependencies within the system are represented with a non-zero entry, the remaining entries being zero. The Adjacency Matrix for the unscheduled tasks from the previous step is shown in Figure 8.22(a). The deduction of the loops to which tasks belongs is achieved by calculating and studying the Reachability Matrix, $R^*$. This is done in the following steps:

(i) Calculate the Boolean union of the Adjacency Matrix and its Identity Matrix. The resulting matrix, $R$, is simply the Adjacency Matrix with all its leading diagonal entries changed to non-zero elements. This calculation can be expressed thus:

\[ R = (A \cup I) \]

(ii) Calculate $z$ which is a power of two that exceeds the number of tasks in the Adjacency Matrix. In the example the number of tasks is ten, and as two to the power of four is greater than ten $z$ must be sixteen.

(iii) Calculate the Reachability Matrix, $R^*$, by raising the result of step (i) by the power of step (ii), i.e.

\[ R^* = R^z = (A \cup I)^z \]

The Reachability Matrix for the example is shown in Figure 8.22(b). In all the cases where $r_{ij} = r_{ji} = 1$ in the Reachability Matrix, task $i$ will be in the same loop as task $j$. A search of the Reachability Matrix must be made to check where this rule holds true and the tasks sorted into their relevant loops. Figure 23(a) highlights which tasks are within which loop in the Reachability Matrix. It can be seen that tasks A, D, J and K are in a loop (indicated by circles) so are E, F, I and L (indicated by squares). The remaining tasks B and H are singular loops which indicate they are not interdependent tasks but tasks dependent on interdependent tasks.
The order in which these loops should be performed can be determined by collapsing all the tasks within a loop into a single quasi-task; the dependencies for this new quasi-task are calculated by the Boolean union of the dependencies of the individual tasks. This allows the procedure discussed in Section 8.7.2 for identifying non-interdependent tasks to be used to determine the priority of these loops. If quasi-task X is equivalent to the loop of tasks A, D, K and J, and quasi-task Y is equivalent to the loop of tasks E, F, I and L, the matrix shown in Figure 8.23(b) can be formed. From this matrix it is possible to order quasi-task Y first, followed by tasks H, B and quasi-task X. The order of individual tasks within a loop is determined by referring back to the Adjacency Matrix and scheduling at the earliest point, the tasks having the minimum number of non-zero entries in the columns corresponding to the tasks within the loop. For instance tasks I and E have the minimum number of non-zero entries and therefore should be scheduled ahead of tasks L and F in the circuit E, F, I and L. Using the tasks orders determined from Step One and from the Boolean algebra manipulation the fully partitioned matrix is obtained and is shown in Figure 8.24.

Loop Identification by Path Searching

Path Searching was developed by Steward (1981a) as a means of identifying and ordering loops within a system. Using the same example as in the Boolean method and ignoring
the tasks with blank rows and columns deduced by Step One (Section 8.7.3) the matrix shown in Figure 8.25(a) can be formulated. Path searching examines the tasks within a matrix by tracing information flow either forward or backwards until a task is encountered twice. All other tasks between the two encounters are then deemed to be in the loop. Path searching begins at any arbitrary point; from Figure 8.25(a) we can begin at task A. Task A is dependent on information from task B, i.e. task B is a predecessor to task A. A path is now searched along row B in which task F is found to be a predecessor. The following path is followed until a loop is encountered: task E is a predecessor to task F; task L is a predecessor to task E; task I is a predecessor to task L; and task L is a predecessor to task I. As task L has been encountered twice, the tasks between these two encounters are therefore in a single loop within it; in this case just task I.

![Figure 25(a) & 25(b): Identifying Loops Using Path Searching](image)

The next step is to collapse the tasks discovered in the loop into a single \textit{quasi-task}, task L and Task I into the \textit{quasi-task} LI, and reform the matrix as shown in Figure 8.25(b). As this matrix contains no blank rows or columns the path searching continues: task B is a predecessor to task A; task F is a predecessor to task B; task E is a predecessor to task F; task LI is a predecessor to task E; and task F is a predecessor to task LI. Between the two encounters of task F are tasks E and the \textit{quasi-task} LI. Therefore, these tasks are all in a single loop which is then combined into another \textit{quasi-task}, termed Y, and another matrix, shown in Figure 8.26(a), formed. In this matrix a blank row from \textit{quasi-task} Y is found and therefore can be scheduled at the earliest point in the matrix. By eliminating this task from the matrix it can be seen that tasks H and B can also be ordered leaving just the tasks A, D, J and K to undergo a path search. This reduced matrix is shown in Figure 8.26(b) and following the path around from task K, via task A and task J it can be seen that tasks A, J and K are all in a loop. Collapsing these tasks into a single \textit{quasi-task}, Z, the new matrix in Figure 8.27(a) clearly shows that tasks A, J and K are in a loop with task D.
Chapter 8

Figure 26(a) & 26(b) : Identifying Loops Using Path Searching

The order of collapsed tasks deduced from the path searching is shown in Figure 8.27(b), which is clearly lower triangular, showing the partitioning has been successful. Determining the order of tasks within each loops can be performed as in the Boolean method or by shunt diagrams used by Steward (1981a, 1991) for large loops. The fully partitioned matrix using the path searching method, shown in Figure 8.28, is exactly the same the partitioned matrix produced by the Boolean method shown in Figure 8.24.

Figure 27(a) & 27(b) : Identifying Loops Using Path Searching

Figure 28 : Fully Partitioned Matrix Using Path Searching
8.7.4 Partitioning Classified Information Dependencies

Any partitioning algorithm must be capable of differentiating between and coping with classified information dependencies to ensure the most important dependencies are below, or as close to, the diagonal as possible. Both partitioning techniques discussed above are capable of this and deal with the problem in the same way in the following steps:

- partition the whole matrix in the appropriate way;
- examine each block individually and eliminate the least important dependency, i.e. in the first instance class C information;
- re-partition each block;
- examine the new blocks containing only class A and B and eliminate class B information; and
- re-partition these smaller blocks.

8.7.5 Software to Perform the Partitioning the Algorithms

Software was procured for both types of partitioning algorithm: the software for the Boolean method was written from first principles at an early stage of this research, whereas the path searching software was obtained from personal communication with Donald Steward. This allowed a comparison of both methods.

Software for the Boolean Algebra Method

A specification for the Boolean method of partitioning were drawn up for an experienced programmer to develop a prototype piece of software. The resultant software was strictly functional and not particularly user friendly, but it allowed data to be entered via a precedence table which was deemed an important attribute if it were to be ultimately linked to a database (see Figure 8.3).

The partitioning or re-ordering capabilities of the software was tested on several matrices of up to 20 tasks and also on the matrix with 51 tasks used earlier in this chapter (Figure 8.4). The software performed the partitioning quickly and accurately on all matrices with less than 20 tasks with the results produced matching those calculated manually. However, the 51 task matrix caused problems as the memory handling capabilities of the computer were exceeded, causing certain information dependencies to be lost. Although much time and effort was spent trying to discover whether the fault was a bug in the code or in the computer itself, the exact problem was never identified. This meant the software developed was only useful for partitioning matrices of less than approximately 25 tasks.

Software for the Path Searching Method

From a private correspondence with Donald Steward we were able to obtain the TERABL programme, described in Steward (1981a), which utilises the Path Searching approach to
### Design Structure Matrix Developed with TIRAIL

<table>
<thead>
<tr>
<th>Figure 8.29: Design Structure Matrix Developed with TIRAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>a13</td>
</tr>
<tr>
<td>c22</td>
</tr>
<tr>
<td>c23</td>
</tr>
<tr>
<td>a1</td>
</tr>
<tr>
<td>a2</td>
</tr>
<tr>
<td>a3</td>
</tr>
<tr>
<td>a5</td>
</tr>
<tr>
<td>c1</td>
</tr>
<tr>
<td>c4</td>
</tr>
<tr>
<td>c5</td>
</tr>
<tr>
<td>c7</td>
</tr>
<tr>
<td>c2</td>
</tr>
<tr>
<td>s1</td>
</tr>
<tr>
<td>s2</td>
</tr>
<tr>
<td>s3</td>
</tr>
<tr>
<td>s4</td>
</tr>
<tr>
<td>s5</td>
</tr>
<tr>
<td>s6</td>
</tr>
<tr>
<td>s7</td>
</tr>
<tr>
<td>c8</td>
</tr>
<tr>
<td>c12</td>
</tr>
<tr>
<td>c9</td>
</tr>
<tr>
<td>s12</td>
</tr>
<tr>
<td>a4</td>
</tr>
<tr>
<td>a5</td>
</tr>
<tr>
<td>a7</td>
</tr>
<tr>
<td>a8</td>
</tr>
<tr>
<td>a11</td>
</tr>
<tr>
<td>a9</td>
</tr>
<tr>
<td>s9</td>
</tr>
<tr>
<td>s10</td>
</tr>
<tr>
<td>a13</td>
</tr>
<tr>
<td>a14</td>
</tr>
<tr>
<td>a15</td>
</tr>
<tr>
<td>a10</td>
</tr>
<tr>
<td>c10</td>
</tr>
<tr>
<td>s11</td>
</tr>
<tr>
<td>c13</td>
</tr>
<tr>
<td>c14</td>
</tr>
<tr>
<td>c16</td>
</tr>
<tr>
<td>c15</td>
</tr>
<tr>
<td>c16</td>
</tr>
<tr>
<td>c17</td>
</tr>
<tr>
<td>c18</td>
</tr>
<tr>
<td>c22</td>
</tr>
<tr>
<td>c21</td>
</tr>
<tr>
<td>c20</td>
</tr>
<tr>
<td>a9</td>
</tr>
<tr>
<td>a12</td>
</tr>
<tr>
<td>c3</td>
</tr>
<tr>
<td>c19</td>
</tr>
</tbody>
</table>
partitioning. The software was found to be easy to manipulate and allowed data to be entered easily via a precedence table.

The results of partitioning the 51 task matrix can be seen in Figure 8.29. TERABL allows information to be classified in up to ten different categories, with class A being represented by an 'X', class B by a '1' and class C by a '2'. If the partitioned matrix produced by TERABL is compared to the hand calculated version in Figure 8.8 it can be seen that two are almost identical. TERABL has identified the same loops as the hand partitioned version but has not ordered them, or some other tasks, in exactly the same way. However, this is unimportant because on closer inspection these loops or individual tasks are independent of each other and can be performed in parallel, so their exact position in the matrix is unimportant and both matrices would produce identical logic networks.

8.7.6 Conclusions on the Partitioning Algorithms
Both methods examined, Boolean algebra and path searching, worked successfully when applied manually, but we were unsuccessful in automating the Boolean technique. Although only limited testing occurred with the Boolean method software, the problems encountered did not encourage further development, especially as the TERABL software was found to so effective.

TERABL was very user friendly and allowed accurate results to be obtained quickly. The only criticism of the software was the difficulty in obtaining an easily readable hard copy of the output. However, at the time of writing this thesis the TERABL software was being rewritten by Steward, in a Windows format, addressing this and other minor problems.

8.8 VERIFICATION AND VALIDATION OF ADePT

8.8.1 Verification
ADePT is largely dependent on the validity of the Design Process Model, and the resultant design programme will only be as accurate as the data it is based upon. Great efforts were made to ensure the data in the DPM and in the DFDs used in the design example were stringently checked (Section 6.4.2) and this process verified the DPM. The matrix analysis software was also verified by comparing the results with the hand analysis. Taken together these two activities therefore verify the ADePT methodology i.e. confirm its integrity and correctness.
8.8.2 Validation

The initial intention was to verify ADePT by analysing a completed project (described in Section 6.3.1) and compare the programmes used against those produced by ADePT: it was believed that this was the most appropriate way of assessing the practicalities and weaknesses of ADePT. Although a great deal of data was collected for one particular project, it became apparent that this approach was an impractical way of validating the technique.

ADePT initially produces an unconstrained design programme based on the optimal flow of information. What makes individual design programmes unique are the internal and external constraints placed upon them. Although it was difficult to assess, and retrospectively model, the impact of internal influences (staff changes, illness etc.) and external influences (procurement, construction programme etc.), the single most important reason why this method of verification was unfeasible was the effect change had on the design process. A design programme is a living, changing document which needs to be constantly updated to reflect changes, such as development of design solutions or a client induced change. As detailed records of these variations and changes were not kept, it was not possible to contrast actual and predicted design programmes because the comparison would not be like for like.

However, the work performed on the design example validated ADePT to some degree. The large blocks of interdependent tasks highlighted in the Design Structure Matrix were confirmed by practising designers as areas where co-ordination problems are known to occur. This finding strongly supports the approach taken in this methodology and partially validates ADePT.

It is believed that the only true test of ADePT's practicality is to use it on an ongoing project, where the influences discussed in the previous section can be identified and incorporated in the design programme as they happen.

It was beyond the scope and time scale of this project to validate ADePT in this manner because the main objective of the research was purely to consider fresh approaches to building design programming. In order to validate ADePT properly on an ongoing project the following work would be required:

- completion of the remaining parts of the Design Process Model; and
- further automation and linking of the separate procedures within ADePT.
8.9 CONCLUSIONS

The aim of this research was to develop an improved approach to the planning of multi-disciplinary building design work and it is believed that ADePT meets this objective. It is appreciated that although ADePT has potential it does have some drawbacks in its present form because it is disjointed, cumbersome and involves the manual manipulation of many procedures and diagrams. If it is to become a practical management tool these issues need to be addressed. It is not proposed to enlarge on these points here because the discussion in the following chapter covers them in greater detail and also reflects on the research project as a whole and gives recommendations for future work.
Chapter 9
Chapter Nine

CONCLUSIONS, REFLECTIONS AND FUTURE WORK

9.1 INTRODUCTION

This final chapter completes this thesis by reviewing the project and comparing the outcome with the objectives set out in the earlier chapters. These conclusions are then followed by the author's reflections on various aspects of the research and a description of future work required to refine and further validate ADePT.

9.2 CONCLUSIONS

9.2.1 Overview & Problem Formulation

This research project set out to examine the building design process and resolve some of the conflicts that make multi-disciplinary building design such a problematic and difficult process to manage. This aspiration was refined by making a detailed study of the factors that have a detrimental influence on a smooth progression through the design process. The study, based on a review of current literature, discussions with design professionals and the author's own personal experience, examined the challenges facing the management of the design of a modern building and their causes. The research also drew on the experience of other design professions, such as manufacturing and production design, because while many of the problems investigated were specific to a particular domain, the underlying reasons were not. From this initial work the following conclusions were drawn. These helped narrow the objectives of the research and form the central hypotheses upon which the work was based.

Problems associated with building design can be categorised into one of five groups. The five inter-related categories to which building design problems can be attributed are:-

(i) Increasingly sophisticated clients or employers.
(ii) Fast-tracking pressures on design.
(iii) Increasing building complexity.
(iv) Insufficient information management.
(v) Difficulty in planning design work.

Although these problems are highly inter-related, a clear distinction can be made between the first three, which cannot be controlled by the design team, and the last two that can.
Poor information management and design planning are inextricably linked and it was argued that an improvement in design planning would facilitate the management of information.

Building design planning is performed poorly for two important reasons. Firstly, modern buildings are so technically complex that all the processes involved in a multi-discipline design project can't easily be assimilated by an individual. This results in a fragmented approach to design planning that does not promote, nor duly consider, cross-discipline co-ordination, resulting in unrepresentative and unworkable design programmes. Secondly, planning techniques used widely within all domains of design, such as network analysis, are not able to represent complex issues unique to design, such as iteration and choice.

From these two findings the overall aim of the research was evolved, namely:-

The development of a more sophisticated approach to the planning and co-ordination of multi-disciplinary building design.

To achieve this aim the research was split into two distinct phases each based on overcoming one of the two weaknesses identified with current design planning procedures. Objectives were set to resolve these two problems and their combination resulted in the development of a prototype design planning methodology, called ADePT (Analytical Design Planning Technique). ADePT has been developed by combining two separate techniques, from different engineering domains, to produce a methodology that has the potential to organise and co-ordinate multi-disciplinary building design projects more effectively than current planning techniques. The findings of these two phases shaped ADePT's development and are outlined in the following section.

9.2.2 Development of the Design Process Model
It was proposed that a clearer understanding of the building design process would be achievable if its features were recorded in a graphical model. The models would represent the knowledge and 'know how' of designers, and capture it in such a way that it could be assimilated by designers and managers alike. Fully validated models would cover the design process from a multi-disciplinary viewpoint and could be analysed repeatedly, to form the basis of a new design planning methodology.

In the course of the work undertaken to meet these objectives the following conclusions were drawn.
Chapter 9

Existing design models are unsuitable to represent the building design process. Initially it was hoped to utilise existing models of the design process and adapt them to represent the building design process. Models from both the manufacturing domain (prescriptive, descriptive and consensus models), and from the construction domain (product and process models) were examined. It was found that neither were adaptable nor sufficiently detailed to represent building design in the manner required to gain a clearer understanding of the whole building design process.

Data flow diagrams are a most suitable technique for modelling building design. Of the many graphical modelling techniques examined, data flow diagrams (DFDs), a technique used in software development, were found to be the most appropriate. DFDs were the most flexible and appropriate technique because, as Fisher (1990) neatly sums up, they have the following properties:

- they are graphical;
- they can be partitioned;
- they are multi-dimensional;
- they emphasise the flow of data rather than control; and
- they represent a situation from the viewpoint of the data rather from the viewpoint of a person or an organisation.

Data flow diagrams were used to form a graphical model of a generic building design process. To ensure the development occurred in a consistent manner several simplifications had to be made to the model. These were chosen to ensure the validity of the model would not be compromised. The simplifications meant that the model should only represent:

- detail design;
- the tasks performed by designers and not those conducted as functions of management;
- the information links that give rise to iterative, but not repetitive, loops; and
- the flow of pre-emptive, recordable information and not informal or verbal queries or requests.

The objective of this phase of the work was to gain a clearer understanding of the building design process and to form a model that can be an integral part of an alternative approach to planning design work. The Design Process Model maps the flow of information between tasks and disciplines in a generic way to achieve this objective.
Chapter 9

As will be discussed in Section 9.3, although the Design Process Model was developed to be an integral part of ADePT it has been found, through practical application within the design environment, to be a useful tool in its own right; it can be used to identify:

- the key tasks and activities performed by each discipline;
- the information flowing between disciplines and tasks at all levels;
- the critical interfaces between disciplines;
- the critical tasks within the design process;
- the size and nature of iterative processes in building design work;
- the stages of the design process that would benefit most from a more efficient approach to planning;
- the different forms of output from design; and
- the aspects of the building design process not of primary importance to the function of producing design output (drawings, calculations etc.).

Although these findings are not associated with the primary objectives of the research, they are of particular interest and form useful applications of the Design Process Model.

9.2.3 Development of ADePT

The objective of the second phase of the research was the development of a new planning technique that could cope with the inherent complexities of design. To achieve this it was proposed to adapt or develop a technique that would analyse the information stored in the Design Process Model (DPM) and produce workable design programmes.

*Design Structure Matrix Analysis is suitable for analysing multi-disciplinary design.*

Design Structure Matrix Analysis (DSMA), a technique originally developed in the 1960's and 70's and used sporadically in production engineering, was found to be capable of analysing the complexities of design. Although the technique can identify and minimise iterative loops, it is not directly applicable to building design. Forming the precedence matrix requires the user to define each task and dependency before the analysis can begin. This is clearly impractical when a large number of tasks from differing disciplines require co-ordinating.

ADePT was formed by combining the Design Process Model with Design Structure Matrix Analysis to overcome the problems outlined above. In addition DSMA was refined to suit the specific needs of building design. While limited validation work was performed on ADePT (see section 9.4), the following conclusions were made about the technique.
Chapter 9

ADePT has many advantages over traditional planning techniques.

- Design tasks and dependencies are pre-defined in ADePT, which makes the production of design programmes quicker and more consistent;
- ADePT allows different design options or alternatives to be examined;
- ADePT classifies information flows, which in turn allows design programmes to be produced that are more representative of the way designers actually perform design;
- ADePT can handle interdependent tasks and schedules them to firstly reduce the number of iterative cycles and, secondly to reduce the number of tasks within each iterative cycle;
- ADePT indicates to designers which tasks require the most cross-disciplinary coordination and integration; and
- ADePT allows a design programme to be developed based on the optimal flow of design information, which can then be adapted and compared to design programmes constrained by external influences such as procurement or the construction programme.
- ADePT has the potential to be used as a tool to investigate the effect of variations on the design process.

ADePT is not a panacea for all design planners' problems; the work performed during this research has only produced a prototype methodology that exhibits some potential. Its major drawback is rather than simplifying the building design planning process, ADePT is more complicated than traditional network analysis, especially when performed manually. However, it is concluded that to plan and co-ordinate design properly the analysis needs to be detailed and this is due to the very nature of the problem rather than the technique itself. ADePT, in its present form, is cumbersome to use and although many of the procedures can be performed by computer, the linking has been performed manually. However, this is seen as a software development issue rather than a research problem.

Whilst the aim of this project has been met in the course of the research, it is appropriate to reflect on the work performed and consider how, with hindsight, any of it could have been performed differently and then look to the possible future of ADePT as a practical design management tool. This is done in the following sections.

9.3 REFLECTIONS

The development of ADePT was neither a smooth nor sequential process and, as with most research work, many dead ends were encountered and hours spent examining ideas that had little or no bearing on the final outline. It is hoped that this reflection on the research methodology may be of help when considering the further development of ADePT.
Chapter 9

The single most important output from the research is the Design Process Model. In the short time between the completion of the research and the submission of this thesis I have personally found the DPM to be a useful tool in its own right when used within the design environment. As well as acting as a prompt when performing individual design activities, the DPM affords a more structured view of the whole design process. In addition, the hierarchical breakdowns of civil and structural engineering activities (Appendix I) have been used as a framework from which fee estimates for various projects have been formulated and it is hoped that in future this hierarchy could provide a template for feedback to compare actual fee with forecasted spend.

In the practical application of the Design Process Model it has become possible to query the validity of many of the decisions made during its development. One characteristic questioned is whether the DPM would be more useful if it represented scheme rather than detail design. In a series of seminars given to introduce ADePT to practitioners this question has raised a great deal of debate and it was interesting to see that while many practising designers appreciated the need for a more structured approach to detail design, many senior managers felt the requirements were more pressing during scheme design. While it could be argued that scheme design would benefit more from a structured approach, the concepts were proven for the more systematic detail design process. It may be possible, with modifications, to apply ADePT to the more nebulous scheme design.

The DFDs were not completed in full detail for the electrical and mechanical services' design processes. However, these processes have been modelled, albeit in much more detail, by Hanby et al. (1993) in a parallel project running at LUT. Also the development of any data flow diagram is greatly facilitated by a detailed understanding of the processes being modelled. For these reasons a decision was taken at an early stage to concentrate the research effort in developing detailed DFDs of only the civil and structural design processes.

Whilst the number of projects examined during the research period was adequate to derive the concepts of the Design Process Model, it was insufficient to be confident of representing all the tasks and dependencies performed by a single discipline in the course of any detail design process. However, it was always recognised that the model produced would not be totally comprehensive nor universally accepted because design is an individual activity, with different designers having different philosophies and adopting differing approaches to achieve the same goal. This raises the question whether the DPM is applicable to the design work of design team organisations other than those based in large multi-disciplinary organisations. While the DPM was developed with this in mind,
and although it is believed it would be applicable to teams of small discrete design practices, this has not been proven; if ADePT is to be widely applicable the DPM needs to be validated with these types of design teams, which are still more prevalent within the building industry than multi-disciplinary teams.

One of the cornerstones to the practicality of ADePT is that the DPM is sufficiently generic to minimise the need to define the information requirements of each task before each design programme is produced. This is a valid argument, however it has become increasingly apparent that whilst the information requirements do not change from one project to the next, the relative importance of the information might. A typical example of this, given by Hedges and Murray (1994), is that of the contrasting importance of solar heat gains in the design of the air conditioning in two types of buildings: a laboratory block with only four windows and an office development with complete curtain walling. While a small change in the rotation of the latter building will have a dramatic effect on the solar gain calculations, a rotation of 180 degrees to the former will have negligible effect. On the basis of this evidence it is hypothesised that different types of building require similar sets of information requirements but with different classifications. The classification of the relative importance of information flows is seen as a crucial to the future success and acceptance of ADePT and it is here that the input of experienced engineers and designers is seen as essential. The system put forward for classifying information has had to strike a compromise between being rigorous and practical. Introducing more groupings, such as Eppinger's (1991) percentile range, whilst arguably more rigorous, is inappropriate for what is in effect a subjective judgement.

It is appreciated, but not anticipated, that although ADePT may be developed and refined further the conclusion may be that ADePT is not practical enough to be used by design managers and that no real benefit can be gained. To reach this decision much work is required and the following section outlines the work required to refine and validate ADePT.

9.4 FUTURE WORK

The procedures developed for ADePT are still at the prototype stage and more work is required to turn ADePT into a usable tool. The work required to achieve this falls into three categories: validation, investigation and refinement. While much of this has been discussed in detail in the main body of the thesis a summary is given below. It should be noted that industry has seen sufficient potential in this work to pursue further research, via a LINK proposal, to achieve some of the suggestions made below.
Validation
As discussed in Sections 6.4.3 and 8.8.2, only limited attempts have been made to validate both the Design Process Model and ADePT throughout this project. The best way of validating both the DPM and ADePT is to use them to produce design programmes for on-going projects and the co-ordination of the design work. The DPM should also be tested to see how applicable it is for design work performed by single discipline practices rather than large multi-disciplinary design organisations.

Investigation
The following issues need to be investigated to analyse the anticipated benefits of ADePT:

(i) Generic nature of the Design Process Model
   • investigate how generic the DPM can be made; and
   • are different sets of information classifications required for different types of building or does each classification need defining for each job?

(ii) Using the Design Process Model for scheme design
   • can scheme design be modelled to the degree of accuracy required to plan it:
   • can the same modelling techniques be used to model scheme design:
   • is it possible to use parts of the DPM of the detail design process to represent scheme design; and
   • can the same model be used but with different information classifications?

(iii) The simplification of ADePT.
   • can ADePT be simplified;
   • can class 'C' type information be removed from the DPM without any adverse affect on the resultant programmes; and
   • can ADePT be used to produce hierarchical programmes with the level of detail represented appropriate to the person using it, i.e. general design activities scheduled for project managers; individual design tasks scheduled for engineers and designers?

(iv) The study of external pressures on the design process.
   • study and produce guidance of the effect the various external influences have on the optimal design task order.
The study of the effects of change and variations

- investigate the ways ADePT can be used to study the effect of individual changes;
- investigate the ways ADePT can be used to study the effect of cumulative changes; and
- develop rules and procedures that will allow project managers to quickly deduce the financial implications of change.

Refinement

The refinement of ADePT has been covered in some depth in Chapter 8, specifically Sections 8.3.1 and 8.4.3, and reference should be made to Figure 8.3. In summary the refinement process will make ADePT a more user friendly and practical tool for managing design and includes the development of the following software:

(i) A front end module that allows users to decide which design options they require for each new project and gives them the ability to change the information classifications within the Design Process Model.

(ii) Linking the DPM and the DSMA so that the analysis can be performed automatically on data within the model. It is envisaged this will involve a neutral database and possibly the conversion of data relating to tasks and their information dependencies into a precedence table before the matrix analysis takes place.

(iii) Software that generates a logic network that can be linked directly to commercially available project management software to produce easily readable hierarchical design programmes.

(iv) Software that can link the resultant design programme back to the DSMA to allow the position of tasks to be manipulated to study the impact of external influences on the design programme.

(v) Routines to allow variations and changes to be analysed.
References


References


References


References


References

References


References


References


Appendix I
The Generic Design Process Model Overall Hierarchy
Civil Design Process Hierarchy

(ii)
Options for Non-Ground Floor Design:

Precast General Floor

- Precast General Floor Outline Drawings
- Precast General Floor GA Design
- Precast General Floor GA Detail Drawings
- Precast General Floor Calc

Options for Non-Ground Floor Design:

Holorib General Floor

- Holorib General Floor Calc
- Holorib General Floor GA Design
- Holorib General Floor GA Detail Drawings
- Holorib General Floor Penetration Co-ordination

Options for Non-Ground Floor Design:

Insitu General Floor

- Insitu General Floor Outline Drawings
- Insitu General Floor GA Design
- Insitu General Floor GA Detail Drawings
- Insitu General Floor Penetration Co-ordination
- Insitu General Floor Calc

(v)
Options for Non-Ground Floor Design:
- Precast Plant Room Floor
- Holonib Plant Room Floor
- Insitu Plant Room Floor

(vi)
Options for Non-Ground Floor Design:

- Precast Walk On Ceiling
  - Precast Ceiling Outline Drawings
    - Precast Ceiling GA Layout Drawings
  - Precast Ceiling Holes Detail Drawings
  - Precast Ceiling Penetration Coordination

- Hololit Walk On Ceiling
  - Hololit Ceiling Outline Drawings
    - Hololit Ceiling GA Drawings
    - Hololit Ceiling Penetration Coordination
  - Hololit Ceiling Holes Detail Drawings
  - Hololit Ceiling GA Detail Drawings

- Insitu Walk On Ceiling
  - Insitu Ceiling Outline Drawings
    - Insitu Ceiling GA Drawings
    - Insitu Ceiling RC Layout Drawings
    - Insitu Ceiling Penetration Coordination
  - Insitu Ceiling Holes Detail Drawings
  - Insitu Ceiling GA Detail Drawings
  - Insitu Ceiling RC Detail Drawings
Drainage Design

- Foul Drainage Design
  - Foul Drainage Layout Details
  - Foul Internal Setting Out Details
  - Foul Site Layout Details

- Storm Drainage Design
  - Storm Drainage Calculations
  - Storm Site Layout Details
  - Manhole Schedule Details

- Drainage Details
  - Internal Detail Draw
  - External Detail Draw

Underground Co-ordinated Services Design

- Co-ordinated Services Calculations
  - Co-ordinated Services Site Layout Details

- Cross Sections thru Service Runs
  - Co-ordinated Services Above Ground Details
  - Co-ordinated Services Below Ground Details

- Access Pit Details Details

Enabling & Site Work Design

- Site Establishment Details
  - Man Setting Out Details

Survey Work

- Drainage Survey Design
  - Site Survey Details
  - Ground Investigation Reports
External Works
- External Works
  - GA Design
- External Works
  - Setting Out Design
- External Works
  - Details Design

Retaining Wall Design
- Retaining Wall
  - Calc
- Retaining Wall
  - Layout Design
- Retaining Wall
  - GA Design
- Retaining Wall
  - RC Design

Earthworks Design
- Cut & Fill
  - Calc
- Earthworks
  - GA Design
- Earthworks
  - Sections Design

Pit & Basement Design
- Pit & Basement
  - Calc
- Pit & Basement
  - GA Design
- Pit & Basement
  - Section & Detail Design
- Pit & Basement
  - RC Design

External Works Design
Retaining Wall Design
Earthworks Design
Pit & Basement Design
ARCHITECTURAL DESIGN

General Drawing Package

Roof Drawing Package

General Builders Work Package

Floor Finishes Package

Ceiling Drawing Package

Wall Cladding Drawing Package

Miscellaneous Drawing Packages

External Works Drawing Package

See Page (xi) of Appendix I

See Page (xii) of Appendix I

See Page (xiv) of Appendix I

See Page (xv) of Appendix I

See Page (xvi) of Appendix I

Architectural Design Process Hierarchy

(x)
Mechanical Services Design Process Hierarchy

(xvii)
High Pressure Hot Water Design

HPHW Calculations
- HPHW Loading Calcs
- Pipework Expansion Calcs
- Pipework Pressure Loss Calcs
- Relief Stream Calcs
- HPHW Pipework Selection
- Pump Selection
- Pipework Equipment Selection
- Primary Plant Selection
- HPHW Schematic Dwg
- HPHW Control Schematic Dwg
- HPHW Section & Detail Dwg
- HPHW Isometric Dwg
- HPHW Pipework Layout Dwg
- Modified Pipework Layout Dwg

HPHW Equipment Selection
- HPHW Work
- HPHW Work
- HPHW Work
- HPHW Work
- HPHW Work

HPHW Drawings
- Plant Room HPHW Pipework Layout Dwg
- High Level GA Pipework Dwg
- High Level Detail Pipework Layout Dwg
- Low Level GA Pipework Dwg
- Low Level Detail Pipework Layout Dwg

Low Level GA Pipework Dwg

Low Level Detail Pipework Layout Dwg
Chilled Water Design

- Chilled Water Calculations
  - CHW Load Calc
  - CHW Pipework Pressure Loss Calc
- CHW Equipment Selection
  - Pump Selection
  - CHW Equipment Selection
  - Chiller Selection
  - CHW Schematic Dwgs
- CHW Water Drawings
  - CHW Schematic Dwgs
  - CHW Section & Detail Dwgs
  - CHW Pipework Layout Dwgs
  - Plant Room CHW Pipework Layout Dwgs
  - Walk On Ceiling CHW Pipework Layout Dwgs
  - Susp Ceiling Void CHW Pipework Layout Dwgs

High Level GA Pipework Dwgs
High Level Detail Pipework Layout Dwgs
Low Level GA Pipework Dwgs
Low Level Detail Pipework Layout Dwgs
General Arrangement Pipework Dwgs
Detail Layout Pipework Dwgs
General Arrangement Pipework Dwgs
Detail Layout Pipework Dwgs
Cold Water Design

- Cold Water Calculations
  - Cold Water Usage Cabs
  - Cold Water Storage Cabs
  - Cold Water Sizing Cabs
  - Pipework Selection

- Cold Water Equipment Selection
  - Pipework Equipment Selection
  - Water Tank Selection

- Cold Water Equipment
  - Cold Water Schematic Dwg
  - Cold Water Section & Detail Dwg
  - Cold Water Control Schematic Dwg

- Cold Water Drawings
  - Cold Water Pipework Layout Dwg
    - Plant Room Cold Water Pipework Layout Dwg
    - W.O. Ceiling Cold Water Pipework Layout Dwg
    - Susp. Ceiling Void Cold Water Pwtr layout Dwg

- Cold Water Pipework Layout
  - High Level GA Pipework Dwg
  - High Level Detail Pipework Layout Dwg
  - Low Level GA Pipework Dwg
  - Low Level Detail Pipework Layout Dwg

- Plant Room Cold Water Pipework Layout Dwg
  - General Arrangement Pipework Dwg
  - Detail Layout Pipework Dwg

- Susp. Ceiling Void Cold Water Pwtr layout Dwg
  - General Arrangement Pipework Dwg
  - Detail Layout Pipework Dwg

- Incoming MCW External Dwg

(xxiv)
Structural Design Process Hierarchy

(xxvii)
Loading Design

- Wind Loading Design
  - Wind Loading Calculations
  - Wind Loading Design
  - Roof Loading Calculations
  - Snow Loading Design
  - Walk On Ceiling Loading Calcs
  - Non Ground Floor Room Floor Loading Calcs
  - Column Loading Calculations

- Vertical Loading Design

- Load Case Design
Structural Steel Work Frame Design:

(xxix)
Structural Steel Work Frame Design:
Primary Steel Work Calculations

- Layout of Primary Structural of Members
- Roof Section Calcs
- Walk On Ceiling Section Calcs
- Non Ground Plan Room Floor Section Calcs
- Column Section Calcs
- Bracing Section Calcs

Structural Steel Work Frame Design:
Primary Steel Work Drawings

- Primary Steel Work Sections Dwg
- Primary Steel Work Roof Plan Dwg
- Primary Steel Work Ceiling Plan Dwg
- Primary Steel Work Non-Ground Floor Plan Dwg
- Primary Steel Work Ground Floor Plan Dwg

Structural Steel Work Frame Design:
Primary Steel Work Details

- Base Plate Steel Work Details Dwg
- Roof Steel Work Details Dwg
- Non Ground Floor/Ceiling Steel Work Detail Dwg
- Load Correction Dwg
Precast Concrete Frame Design:

(XXXI)
Precast Concrete Frame Design: Precast Section Size Calculations

Precast RC Calculations

Precast Members Plan Dwg

Ground Floor Members Plan Dwg

Non-Ground Floor Members Plan Dwg

Ceiling Members Plan Dwg

Roof Members Plan Dwg

Primary Precast Members Layout Dwg

Precast Members Plan Dwg

Precast Members Elevation Dwg

Roof Member RC Calcs

Walk On Ceiling Member RC Calcs

Non-Ground Floor Members RC Calcs

Column Member RC Calcs

Shear Wall RC Calcs

Calculation of Section Sizes

Section Size Calculations

Roof Member Section Calcs

Calculation of Section Sizes

Primary Precast Members Layout Dwg

Precast Members Layout Dwg

Precast Members Plan Dwg

Precast Section Size Calculations

Precast Concrete Frame Design: Precast Section Size Calculations

Primary Precast Members Layout Dwg

Precast Concrete Frame Design: Primary Precast Members Layout Dwg

(33ii)
### Loading Design

#### Walk On Ceiling Loading Calculations

(S5)

<table>
<thead>
<tr>
<th>Information Flow</th>
<th>Source of Information</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• approx details of gen mech plant</td>
<td>M</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• approx details of gen elec plant</td>
<td>E</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• loading from mech suspended services</td>
<td>M</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• loading from elec suspended services</td>
<td>E</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• approx ceiling constn make-up details</td>
<td>C22 Walk On Ceiling Calcs</td>
<td></td>
</tr>
<tr>
<td>• proposed ceiling uses</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• approx partitioning details</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• approx finishes details</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• self weight of support swk</td>
<td>S12 Walk On Ceiling Section Calcs</td>
<td></td>
</tr>
<tr>
<td>• construction loadings</td>
<td></td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>• soffit profile</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

Including method of slab construction

(XXXV)
## Primary Steel Work Drawings

### Primary Steel Work Sections Drawings (S19)

<table>
<thead>
<tr>
<th>Information Flow</th>
<th>Source of Information</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary roof swk layout</td>
<td>S18 Prim Steel Work Roof Plan Dwgs</td>
<td></td>
</tr>
<tr>
<td>primary plant room floor swk layout</td>
<td>S16 Prim SWK Plt Rm Flr Plan Dwgs</td>
<td></td>
</tr>
<tr>
<td>primary ground floor swk layout</td>
<td>S15 Prim SWK Gnd Floor Plan Dwgs</td>
<td></td>
</tr>
<tr>
<td>primary walk on ceiling swk layout</td>
<td>S17 Prim SWK Ceiling Plan Dwgs</td>
<td></td>
</tr>
<tr>
<td>layout of primary bracing members</td>
<td>S9 Layout of Primary Struct Members</td>
<td></td>
</tr>
<tr>
<td>details of bracing section sizes</td>
<td>S14 Bracing Section Calcs</td>
<td></td>
</tr>
<tr>
<td>levels of steel work</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>building sections</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>architectural elevation details</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>external cladding support details</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>service riser section details</td>
<td>S34 Vertical Service Riser Calcs</td>
<td></td>
</tr>
<tr>
<td>movement riser section details</td>
<td>S35 Vertical Stairwell Riser Calcs</td>
<td></td>
</tr>
</tbody>
</table>

(*xxxvi*)
## Primary Steel Work Drawings

### Primary Steel Work Sections Drawings (S19)

<table>
<thead>
<tr>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>roof purlin details</td>
<td>S25 Cladding Support Calcs</td>
</tr>
<tr>
<td>wall purlin details</td>
<td>S25 Cladding Support Calcs</td>
</tr>
<tr>
<td>foundation depths</td>
<td>C9 Pile Cap Calcs</td>
</tr>
<tr>
<td>accurate details of main duct routes</td>
<td>M</td>
</tr>
<tr>
<td>steel work elevation details</td>
<td>S20 Prim SWK Elevation Dwgs</td>
</tr>
</tbody>
</table>

*To be confirmed at later date*
# Piled Foundation Design

## Pile Cap Layout Drawings (C10)

<table>
<thead>
<tr>
<th>Information Flow</th>
<th>Source of Information</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>setting out dimensions</td>
<td>C9 Pile Cap Calcs</td>
<td></td>
</tr>
<tr>
<td>pile cap dimensions</td>
<td>C9 Pile Cap Calcs</td>
<td></td>
</tr>
<tr>
<td>accurate ground beam positions</td>
<td>C4 Ground Beam GA Dwgs</td>
<td></td>
</tr>
<tr>
<td>accurate pile layout</td>
<td>C7 Pile Layout Dwgs</td>
<td></td>
</tr>
<tr>
<td>levels to top of piles</td>
<td>C6 Piling Calcs</td>
<td></td>
</tr>
</tbody>
</table>
## Drainage Design

### Storm Drainage Calculations (C17)

<table>
<thead>
<tr>
<th>Information Flow</th>
<th>Source of Information</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>existing drainage layouts</td>
<td>C42 Drainage Survey Dwgs</td>
<td></td>
</tr>
<tr>
<td>existing underground services</td>
<td>C43 Site Survey Dwgs</td>
<td></td>
</tr>
<tr>
<td>approx roof layouts</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>approx RWP positions</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>hard surface layouts</td>
<td>A</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>approx foundation positions</td>
<td>C9 Pile Cap Calcs</td>
<td></td>
</tr>
<tr>
<td>approx u/g services position</td>
<td>C35 Co-ordinated Services Calcs</td>
<td></td>
</tr>
<tr>
<td>proposed external ground levels</td>
<td>A</td>
<td>land drainage</td>
</tr>
</tbody>
</table>

* (xxxix)*
<table>
<thead>
<tr>
<th>Source of Information</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>C29 Plant Room Floor Calcs</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>C29 Plant Room Floor Calcs</td>
<td>To be confirmed at later date</td>
</tr>
<tr>
<td>C31 Plant Room Floor GA Dwgs</td>
<td>M/E</td>
</tr>
<tr>
<td>S19 Primary SWK Section Dwgs</td>
<td>A</td>
</tr>
<tr>
<td>S22 Non Gnd &amp; Ceiling SWK Dots Dwgs</td>
<td>A</td>
</tr>
<tr>
<td>S16 Prim SWK Plant Floor Plan Dwgs</td>
<td>A</td>
</tr>
</tbody>
</table>

### Information Flow
- bundling & drainage details
- plant floor construction details
- plant floor concret details
- swk sections details
- interface with floor boundaries
- floor/swk connection details
- accurate primary steelwork positions
- accurate plinth/recess details
- fire stopping details
Appendix III
Diagram 0: The Detail Design Process
Diagram 3.2: Loading Design

Diagram 3.2.1: Wind Loading Design

Diagram 3.2.2: Vertical Loading Design

Diagram 3.2.3: Load Case Combination Cals
Diagram 3.2.2: Vertical Loading Design

Diagram 3.3: Structural Steel Work Design
Diagram 3.3.2: Primary Steel Work Drawings

Diagram 3.3.2.1: Primary Steel Work Plan Drawings

(xlvi)
Diagram 3.4.1: Cladding Support Design

Diagram 3.4.2: Roof Safety Design
Diagram 3.3.3: Primary Steel Work Details Drawings

Diagram 3.4: Secondary Steel Work Design
Diagram 3.4.3: Trimmer Support Design

Diagram 3.4.3.1: Trimmer Support Calculations
Diagram 3.4.3.2: Trimmer Support Drawings

Diagram 3.5: Vertical Shaft Design
Diagram 4.2.2: Pile Cap Design

Diagram 4.3: Suspended Ground Floor Slab Design
Diagram 4.3.2: Suspended Ground Floor Drawings

Diagram 4.3.3: Suspended Ground Beam Drawings
Diagram 4.4: Non Ground Floor Design

Diagram 4.4.1: Insitu Walk On Ceiling Design
Diagram 4.4.1.2 : Walk On Ceiling Outline Drawings

Diagram 4.4.1.3 : Walk On Ceiling Detail Drawings
Diagram 4.4.2.1: Holorib Plant Room Floor Design

Diagram 4.4.2.2: Holorib Plant Room Outline Drawings
Diagram 4.5.2: Storm Drainage Design

Diagram 4.5.3: Drainage Details Drawings
Diagram 4.6: Underground Co-ordinated Services Design

Diagram 4.6.3: Cross Sections Through Service Runs

(lxi)
Diagram 4.7: Enabling Works Design

Diagram 4.8: Survey Work
Diagram 6: Document Control