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/
AN INTEGRATED PRODUCT AND PROCESS INFORMATION MODELLING SYSTEM FOR ON-SITE CONSTRUCTION

ANDREW GEORGE KIMMANCE
B.Sc. (Hons), M.Sc.Eng, CIOB

A Doctoral Thesis submitted in partial fulfilment of the requirements of the Loughborough University for the Degree of Doctor of Philosophy

November 2002
Abstract

The inadequate infrastructure that exists for seamless project team communications has its roots in the problems arising from fragmentation, and the lack of effective co-ordination between stages of the construction process. The use of disparate computer-aided engineering (CAE) systems by most disciplines is one of the enduring legacies of this problem and makes information exchange between construction team members difficult and, in some cases, impossible. The importance of integrating modelling techniques with a view to creating an integrated product and process model that is applicable to all stages of a construction project's life cycle, is being recognised by the Construction Industry. However, improved methods are still needed to assist the developer in the definition of information model structures, and current modelling methods and standards are only able to provide limited assistance at various stages of the information modelling process.

This research investigates the role of system integration by reviewing product and process information models, current modelling practices and modelling standards in the construction industry, and draws conclusions with similar practices from other industries, both in terms of product and process representation, and model content. It further reviews various application development tools and information system requirements to support a suitable integrated information structure, for developing an integrated product and process model for design and construction, based on concurrent engineering principles. The functional and information perspectives of the integrated model, which were represented using IDEF0 and the unified modelling language (UML), provided the basis for developing a prototype hyper-integrated product and process information modelling system (HIPPY). Details of the integrated conceptual model's implementation, practical application of the prototype system, using house-building as an example, and evaluation by industry practitioners are also presented.

It is concluded that the effective integration of product and process information models is a key component of the implementation of concurrent engineering in construction, and is a vital step towards providing richer information representation, better efficiency, and the flexibility to support life cycle information management during the construction stage of small to medium sized-building projects.
Dedication

To my loving mother, Carol.
my brother, Richard, and sisters, Nicola and Lindsay

my loving wife, Laura

and

To The Lord Jesus Christ,
who has made my life worth living,
and to whom honour and glory
for ever and ever, Amen!
Acknowledgements

I wish to express my sincere thanks to all those who have contributed to the successful completion of my PhD research project.

To the EPSRC and the Loughborough University’s studentship, I express my appreciation for the provision of funds for the research.

The contribution of my supervisors, Prof. Chimay J. Anumba, Dr. Dino Bouchlaghem, and director of research studies, Prof. Andrew N. Baldwin, in providing guidance and support, and various sources of information for the research is deeply appreciated. My special thanks to Prof. Chimay J. Anumba, and Dr. Dino Bouchlaghem, for being instrumental in securing the funding for the research, for being mentors, and for their constant inspiration and encouragement. I owe a deep dept of gratitude towards both of them.

I am also grateful to the following individuals who helped me in various aspects of my research through my project: Prof. Anne-Françoise. Cutting-Decelle (Paris University), Dr. Alistair Gibb, Mrs Joy Hull, Mrs Jo Brewin, Mrs Jackie Palmer, Mrs J. Robon, Mrs J. Burtan, Mrs C. Barton, Mrs C. Neale (Loughborough University, and Dr. John M. Kamara (Newcastle University).

The organisations, which participated in the industry surveys, and many individuals at conferences who were willing to discuss and clarify issues, relevant to the research, I am sincerely grateful. The contribution of my colleagues at the Department of Civil & Building Engineering, who provided encouragement and advice to me at various times, is highly appreciated.

I am very much indebted to my mother, Carol, my brother, Richard, my sisters Nicola and Lindsay, and especially my wife, Laura, for their patience, love, support, and encouragement.

And finally, my profound gratitude and praise go to God Almighty, who has enabled me to carry out this research, for sustaining me by His grace, and for using people to help me in times of need.

I thank them all Very Much.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xiii</td>
</tr>
<tr>
<td>ACRYNOMS</td>
<td>xiv</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 GENERAL INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.2 BACKGROUND</td>
<td>2</td>
</tr>
<tr>
<td>1.3 PROBLEM DOMAIN</td>
<td>3</td>
</tr>
<tr>
<td>1.4 INTEROPERABILITY PROBLEMS</td>
<td>8</td>
</tr>
<tr>
<td>1.5 POSSIBLE SOLUTIONS TO THE PROBLEMS</td>
<td>11</td>
</tr>
<tr>
<td>1.6 AIMS AND OBJECTIVES OF RESEARCH</td>
<td>15</td>
</tr>
<tr>
<td>1.7 SCOPE OF RESEARCH</td>
<td>16</td>
</tr>
<tr>
<td>1.7 JUSTIFICATION FOR THE RESEARCH</td>
<td>16</td>
</tr>
<tr>
<td>1.9 THESIS STRUCTURE</td>
<td>17</td>
</tr>
<tr>
<td>CHAPTER 2: RESEARCH METHODOLOGY</td>
<td>19</td>
</tr>
<tr>
<td>2.1 INTRODUCTION</td>
<td>19</td>
</tr>
<tr>
<td>2.2 THEORY OF RESEARCH</td>
<td>19</td>
</tr>
<tr>
<td>2.2.1 Generic Issues in Research</td>
<td>19</td>
</tr>
<tr>
<td>2.3 TYPES OF RESEARCH</td>
<td>22</td>
</tr>
<tr>
<td>2.3.1 Quantitative verses Qualitative Research</td>
<td>22</td>
</tr>
<tr>
<td>2.3.2 Qualitative Research Methodologies</td>
<td>24</td>
</tr>
<tr>
<td>2.3.3 Quantitative Research Methodologies</td>
<td>24</td>
</tr>
<tr>
<td>2.3.4 Hybrid Research Approaches (Triangulation)</td>
<td>30</td>
</tr>
<tr>
<td>2.4 METHODOLOGY ADOPTED</td>
<td>32</td>
</tr>
<tr>
<td>2.4.1 The HIPPY Research Model</td>
<td>32</td>
</tr>
<tr>
<td>2.4.2 Research Philosophy</td>
<td>33</td>
</tr>
<tr>
<td>2.4.3 Research Approach (information system development)</td>
<td>34</td>
</tr>
<tr>
<td>2.4.4 System Approach</td>
<td>35</td>
</tr>
<tr>
<td>2.4.5 System Analysis</td>
<td>36</td>
</tr>
<tr>
<td>2.4.6 Research Techniques</td>
<td>37</td>
</tr>
</tbody>
</table>
2.4.6.1 Literature Review ............................................... 37
2.4.6.2 Interviews with Domain Experts ................................ 37
2.4.6.3 Surveys ........................................................ 38
2.4.6.4 Triangulation Method ......................................... 38
2.4.6.5 Conceptual Modelling ...................................... 39
2.4.6.6 Prototype Development and Evaluation .................... 41

2.5 INTEGRATED PRODUCT AND PROCESS MODELLING .......... 41
2.6 CHOICE OF SOFTWARE TOOLS FOR MODELLING .......... 44
2.7 SUMMARY ........................................................... 44

CHAPTER 3: STATE OF THE ART REVIEW OF MODELLING METHODS

3.1 INTRODUCTION ...................................................... 46
3.2 EXISTING MODELLING TECHNIQUES ............................ 46
3.3 INFORMATION MODELLING ...................................... 47
  3.3.1 Product Modelling ............................................... 47
  3.3.2 Product Modelling Techniques and data Standardisation Efforts 50
  3.3.3 Product Model Based Standards .................................. 51
    3.3.3.1 STEP Techniques .......................................... 51
    3.3.3.2 EXPRESS and EXPRESS-G .................................... 52
    3.3.3.3 NIAM ........................................................ 53
    3.3.3.4 IDEF1 and IDEF1x .......................................... 53
    3.3.3.5 International Alliance for Interoperability ................. 54
    3.3.3.6 CIMsteel Integration Standards (CIS) .................... 55
  3.3.4 Object Model Based Standards .................................. 56
    3.3.4.1 Object-oriented Paradigm .................................. 56
    3.3.4.2 CORBA ...................................................... 58
    3.3.4.3 The Unified Modelling Method ................................ 59
  3.3.5 Document Model Based Standards ................................ 59
    3.3.5.1 Extensive Mark-Up Language (XML) .................... 60
    3.3.5.2 Alternative Methods ...................................... 61
  3.3.6 Process Models of the Construction Process .................... 62
  3.3.7 Process Modelling Techniques .................................. 64
    3.3.7.1 IDEFØ ....................................................... 65
    3.3.7.2 Data Flow Diagrams ...................................... 66
    3.4.7.3 Role Activity Diagrams .................................... 67
    3.3.7.4 Petri Nets ................................................... 68
    3.3.7.5 Scheduling ................................................ 69
  3.3.8 project Modelling (Integrated Product and Process Models) ....... 70
3.4 SELECTION OF MODELLING TECHNIQUES .......................... 72
  3.4.1 Comparison of Product Modelling Techniques ............... 72
  3.4.2 Comparison of Process Modelling Techniques ................ 73
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.3 A Detailed Description of IDEFØ Modelling</td>
<td>75</td>
</tr>
<tr>
<td>3.4.4 The Unified Modelling Language</td>
<td>77</td>
</tr>
<tr>
<td>3.5 COMPUTER INTEGRATED MODELLING ENVIRONMENT</td>
<td>80</td>
</tr>
<tr>
<td>3.5.1 Need for Integrated Project Models</td>
<td>80</td>
</tr>
<tr>
<td>3.5.2 Alternative Architectures for an Integrated Project Model</td>
<td>82</td>
</tr>
<tr>
<td>3.5.3 Related Integrated Projects</td>
<td>82</td>
</tr>
<tr>
<td>3.5.4 Evaluation of Existing Model Approaches</td>
<td>86</td>
</tr>
<tr>
<td>3.5.4.1 Criteria for Analysis</td>
<td>86</td>
</tr>
<tr>
<td>3.6 SOFTWARE SELECTION</td>
<td>87</td>
</tr>
<tr>
<td>3.6.1 Survey of Software Tools</td>
<td>90</td>
</tr>
<tr>
<td>3.6.2 Evaluation Criteria</td>
<td>90</td>
</tr>
<tr>
<td>3.6.4 Microsoft Visio 2000</td>
<td>93</td>
</tr>
<tr>
<td>3.7 DISCUSSION</td>
<td>94</td>
</tr>
<tr>
<td>3.8 SUMMARY</td>
<td>99</td>
</tr>
</tbody>
</table>

## CHAPTER 4: INTEGRATED CONCEPTUAL MODEL

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 INTRODUCTION</td>
<td>100</td>
</tr>
<tr>
<td>4.2 CONCEPTUAL DESCRIPTION OF THE BUILDING DELIVERY PROCESS</td>
<td>101</td>
</tr>
<tr>
<td>4.2.1 Background</td>
<td>101</td>
</tr>
<tr>
<td>4.2.2 Definition and Terminology</td>
<td>101</td>
</tr>
<tr>
<td>4.2.3 Factors Effecting the Productivity in the Building Delivery Process</td>
<td>104</td>
</tr>
<tr>
<td>4.2.4 Information Scope for House-building</td>
<td>107</td>
</tr>
<tr>
<td>4.2.4.1 Product Axis</td>
<td>107</td>
</tr>
<tr>
<td>4.2.4.2 The Process Axis</td>
<td>108</td>
</tr>
<tr>
<td>4.2.4.3 Actor Axis</td>
<td>109</td>
</tr>
<tr>
<td>4.2.4.4 Resources Axis</td>
<td>110</td>
</tr>
<tr>
<td>4.3 CONCEPTUAL INFORMATION MODEL</td>
<td>111</td>
</tr>
<tr>
<td>4.3.1 Functional Requirements</td>
<td>111</td>
</tr>
<tr>
<td>4.3.2 System Architecture to Support Information Integration</td>
<td>114</td>
</tr>
<tr>
<td>4.4 THE GENERIC CONSTRUCTION PROCESS MODEL (GCPM)</td>
<td>117</td>
</tr>
<tr>
<td>4.4.1 Development of the Process Information Model</td>
<td>117</td>
</tr>
<tr>
<td>4.4.1.1 Review of Modelling Tools and Techniques</td>
<td>117</td>
</tr>
<tr>
<td>4.4.1.2 Identification of Requirements</td>
<td>118</td>
</tr>
<tr>
<td>4.4.1.3 Identification of Construction Activities</td>
<td>119</td>
</tr>
<tr>
<td>4.4.2 Description of the Generic Construction Process Model</td>
<td>119</td>
</tr>
<tr>
<td>4.4.2.1 The IDEFØ Notation</td>
<td>120</td>
</tr>
<tr>
<td>4.4.2.2 The Contents of the Construction Process Model</td>
<td>121</td>
</tr>
<tr>
<td>4.4.2.3 The Top Level Process Activity Diagram</td>
<td>122</td>
</tr>
<tr>
<td>4.4.2.4 The Context Diagram: Construct House Facility</td>
<td>123</td>
</tr>
<tr>
<td>4.4.2.5 Third Level Diagrams</td>
<td>124</td>
</tr>
<tr>
<td>4.4.2.6 Four Level Diagrams</td>
<td>126</td>
</tr>
</tbody>
</table>
Contents

4.4.2.7 Information, ICOM Attribute and Hyperlink ........................................ 127
4.4.3 Discussions and Limitation ................................................................. 128

4.5 THE GENERIC BUILDING PRODUCT MODEL .............................................. 130
4.5.1 Development of the Generic Building Product Model ................................ 130
4.5.2 Modelling the System Requirements ..................................................... 131
4.5.3 Description of the UML Diagrams .......................................................... 132
4.5.3.1 Modelling Use Case Diagrams .......................................................... 132
4.5.3.2 Dynamic Modelling through Sequence Diagrams .................................. 135
4.5.3.3 Class Modelling ................................................................................. 137
4.5.4 Description of the Generic Building Product Model .................................. 139
4.5.4.1 Information Structure of the Generic Building Product Model .............. 139
4.5.4.2 Basic Entities of the Generic Building Product Model ...................... 141

4.6 THE INTEGRATED CONCEPTUAL MODEL ................................................. 145
4.6.1 Discussion of the Modelling Methods .................................................... 146

4.7 SUMMARY .................................................................................................. 148

CHAPTER 5: IMPLEMENTATION OF THE PROTOTYPE SYSTEM

5.1 INTRODUCTION .......................................................................................... 151

5.2 SCOPE, OBJECTIVE AND CONTEXT OF PROTOTYPE SYSTEM ............... 151
5.2.1 Scope of HIPPY ..................................................................................... 151
5.2.2 Objectives of HIPPY ............................................................................ 152

5.3 PROTOTYPE SOFTWARE DEVELOPMENT .............................................. 154
5.3.1 Approaches to Prototyping .................................................................... 154
5.3.2 Strategies for Prototype Software Development .................................. 155

5.4 IMPLEMENTATION ENVIRONMENT ....................................................... 156
5.4.1 Development of the Prototype System (HIPPY) .................................... 157
5.4.2 Implementation of the Prototype System (HIPPY) ............................... 160
5.4.3 Prototype System Components ............................................................ 160
5.4.3.1 Infrastructure Platform ..................................................................... 161
5.4.3.2 Database Storage Platform .............................................................. 162
5.4.3.3 User-Interface Platform .................................................................... 162
5.4.4 The Microsoft Windows Environment ................................................ 163
5.4.4.1 Microsoft Office 2000 ...................................................................... 164
5.4.4.2 Microsoft Visio 2000 ...................................................................... 164

5.5 POTENTIAL PRACTICAL APPLICATION OF HIPPY ................................. 167
5.5.1 Use of the Prototype System ................................................................. 168
5.5.2 On-site Construction Scenario ............................................................... 170

5.6 DESIGNING HIPPY IN MICROSOFT ACCESS .......................................... 171
5.6.1 Development Process of the Prototype System (HIPPY) ...................... 172
5.6.2 Operational System Architecture of HIPPY ....................................... 173
5.6.3 The Development of Database Objects for HIPPY .............................. 175
5.6.3.1 Designing Tables and Forms ............................................................ 177
### Chapter 5: Designing Reports and Macros

5.6.3.2 Designing Reports and Macros ........................................ 181
5.6.4 Transferring Data and Objects between Forms ....................... 185
5.6.5 Hippy and the Implemented Features for the Prototype System .... 186
5.6.6 Benefits of HIPPY ........................................................... 187

### Chapter 6: Operation and Evaluation

6.1 Introduction ............................................................................ 189
6.2 Operational Objectives .......................................................... 189
6.3 System Operation ................................................................. 190
  6.3.1 Starting the Prototype System (HIPPY) ............................ 190
  6.3.2 Defining Information Requirements ................................. 192
  6.3.3 Entering and Editing Information ..................................... 194
  6.3.4 Forms for Integrating Process and Product Information ...... 194
    6.3.4.1 Inputting Information in Top-Level Process Forms ....... 194
    6.3.4.2 Inputting Information in Top-Level Product Forms ...... 197
  6.3.5 Inputting Information in other Product and Process Forms .... 200
    6.3.5.1 Sub-process Activity Forms ..................................... 200
    6.3.5.2 Structuring Functions and Activity Durations ............ 201
    6.3.5.3 Structuring of ICOM Information .............................. 203
    6.3.5.4 Sub-product Element Forms .................................... 206
  6.3.6 Generating Product and Process Reports ............................ 209
  6.3.7 Integrating Product and Process Information Reports ............ 211
6.4 Discussion .............................................................................. 213
6.5 System Evaluation ............................................................... 214
  6.5.1 Evaluation Objectives ...................................................... 214
  6.5.2 Constraints on the Evaluation .......................................... 214
  6.5.3 Evaluation Approach ...................................................... 214
    6.5.3.1 Evaluation Procedure ............................................ 215
    6.5.3.2 Background of Evaluators ...................................... 216
  6.5.4 Test Case for Evaluation .................................................. 217
  6.5.5 Review of the Evaluation Questionnaire ............................ 218
  6.5.6 Overview of Results ...................................................... 219
  6.5.7 Suggestions for Improvement and other Comments .............. 222
  6.5.8 Strengths and Limitations .............................................. 224
6.6 Review of the Objectives for HIPPY ....................................... 224
6.7 Self-Appraisal of HIPPY ....................................................... 227
6.7 Summary .............................................................................. 230
CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

7.1 INTRODUCTION ................................................. 231
7.2 GENERAL SUMMARY ........................................... 231
7.3 BENEFITS .......................................................... 235
7.4 LIMITATIONS ...................................................... 237
7.5 CONCLUSIONS ...................................................... 238
   7.5.1 Prototype System (HIPPY) ................................. 238
   7.5.2 Integrated Product and Process Modelling ............... 239
   7.5.3 Integration in Construction ................................. 240
7.6 RECOMMENDATIONS FOR FURTHER WORK ................. 241
   7.6.1 Prototype System (HIPPY) ................................. 241
   7.6.2 Integrated Product and Process Modelling ............... 242
   7.6.3 Integration in Construction ................................. 243
7.7 CLOSING REMARKS .............................................. 244

REFERENCES .......................................................... 245
BIBLIOGRAPHY ......................................................... 261
APPENDIX 1 .......................................................... 262
APPENDIX 2 .......................................................... 283
APPENDIX 3 .......................................................... 286
# LIST OF FIGURES

**Chapter 2**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The HYPPY research model</td>
<td>33</td>
</tr>
<tr>
<td>2.2</td>
<td>The research process illustrating the steps for the development of HYPPY</td>
<td>42</td>
</tr>
</tbody>
</table>

**Chapter 3**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The framework of the RATAS model</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>IDEFØ Notation</td>
<td>65</td>
</tr>
<tr>
<td>3.3</td>
<td>An example of a data flow diagram for on-site construction</td>
<td>67</td>
</tr>
<tr>
<td>3.4</td>
<td>A basic RAD example of a construction house project</td>
<td>68</td>
</tr>
<tr>
<td>3.5</td>
<td>An example of a Gantt chart</td>
<td>70</td>
</tr>
<tr>
<td>3.6</td>
<td>Basic concepts representing the IRMA model with three levels of abstraction and the improved AEC process view model</td>
<td>71</td>
</tr>
<tr>
<td>3.7</td>
<td>Basic concept of the IDEFØ method illustrating the top-level process model</td>
<td>76</td>
</tr>
<tr>
<td>3.8</td>
<td>An example of the top level [A0] construction activity diagram</td>
<td>76</td>
</tr>
<tr>
<td>3.9</td>
<td>Package diagram showing input into relationship of the UML diagrams</td>
<td>79</td>
</tr>
<tr>
<td>3.10</td>
<td>Results of the software evaluation</td>
<td>93</td>
</tr>
<tr>
<td>3.11</td>
<td>Microsoft windows platform, illustrating the drawing facilities of Visio, and generic method and architecture of HIPPY</td>
<td>95</td>
</tr>
</tbody>
</table>

**Chapter 4**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>The construction management task</td>
<td>104</td>
</tr>
<tr>
<td>4.2</td>
<td>Work content for a typical housing project</td>
<td>105</td>
</tr>
<tr>
<td>4.3</td>
<td>View of structural entities required to construct a house facility</td>
<td>108</td>
</tr>
<tr>
<td>4.4</td>
<td>Information structure in the proposed Conceptual Information Model</td>
<td>113</td>
</tr>
<tr>
<td>4.5</td>
<td>Logical view of the integrated information modelling environment</td>
<td>115</td>
</tr>
<tr>
<td>4.6</td>
<td>Construct house facility hierarchy node tree</td>
<td>120</td>
</tr>
<tr>
<td>4.7</td>
<td>The top-level process activity diagram for the GCPM</td>
<td>122</td>
</tr>
<tr>
<td>4.8</td>
<td>Node A0 illustrating the five main activity stages of the GCPM</td>
<td>123</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Node A3 illustrating a third level diagram “Construct Substructure”</td>
<td>125</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Showing Node: A4 a third level diagram “Construct Superstructure”</td>
<td>125</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Fourth level diagram-illustrating Node: A33 “Construct Foundation”</td>
<td>126</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>The framework of the Process Activity Information Component</td>
<td>127</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Capturing and integrating the information for the generic process model</td>
<td>130</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>A top-level “Construct House” Use Case diagram</td>
<td>134</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>Sequence diagram for the “Construct Foundation” Use Case</td>
<td>135</td>
</tr>
<tr>
<td>Figure 4.16</td>
<td>Basic structure of a Class diagram</td>
<td>137</td>
</tr>
<tr>
<td>Figure 4.17</td>
<td>Breakdown of the building product data into classes and sub-classes</td>
<td>140</td>
</tr>
<tr>
<td>Figure 4.18</td>
<td>The “core” of the product model using UML class notation</td>
<td>141</td>
</tr>
<tr>
<td>Figure 4.19</td>
<td>Elaborated description of the generic building product information model</td>
<td>144</td>
</tr>
<tr>
<td>Figure 4.20</td>
<td>Integrated Conceptual Model of the building delivery process</td>
<td>147</td>
</tr>
<tr>
<td>Figure 4.21</td>
<td>The integration of IDEFØ and UML modelling approaches</td>
<td>149</td>
</tr>
</tbody>
</table>

**Chapter 5**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 5.1</td>
<td>System components for the (HIPPY) prototype</td>
<td>161</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Microsoft software development platform</td>
<td>163</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Microsoft Windows Platform, illustrating facilities of Visio, and generic method and architecture of HIPPY</td>
<td>166</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>The development lifecycle process</td>
<td>173</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Operational system architecture of HIPPY</td>
<td>174</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Foundation wall design</td>
<td>176</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>The datasheet view of an activity table</td>
<td>177</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Hierarchy form showing product element fields, controls, and property sheet</td>
<td>179</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Multiple form design showing process information and model fields</td>
<td>179</td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>Report ‘Design View’ for product and process information</td>
<td>182</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>Report ‘Print View’ of integrated product and process information</td>
<td>182</td>
</tr>
</tbody>
</table>
List of Figures

Figure 5.12 Report structure illustrating process activity modelling information 183

Chapter 6

Figure 6.1 Welcome Screen 191
Figure 6.2 Main Switchboard control form 192
Figure 6.3 Top-level process information form 195
Figure 6.4 Sub-activity form 197
Figure 6.5 Top-level product information form 198
Figure 6.6 Form for structuring sub-product information 199
Figure 6.7 Sub-activity form activated by clicking the 'next record' button 201
Figure 6.8 Function box information form 202
Figure 6.9 Form for ICOM attributes 204
Figure 6.10 Form for structuring activity ICOM arrows (connection relationships) 204
Figure 6.11 Form for structuring types of ICOM attributes 205
Figure 6.12 Form for structuring building element types 206
Figure 6.13 Decomposed building elements and their measurements 207
Figure 6.14 Forms for inputting building component information 208
Figure 6.15 Forms for structuring types of building components 209
Figure 6.16 Integrated product and process relationships report 210
Figure 6.17 Report structure for the top-level function box 210
Figure 6.18 Top-level product information report 211
Figure 6.19 Report displaying the integration of product and process information 212
# LIST OF TABLES

## Chapter 2

<table>
<thead>
<tr>
<th>Table 2.1</th>
<th>Purpose, Common Forms, and Examples of Research Techniques Used in Qualitative Research</th>
<th>27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.2</td>
<td>Techniques Used in Qualitative Research</td>
<td>30</td>
</tr>
</tbody>
</table>

## Chapter 3

<table>
<thead>
<tr>
<th>Table 3.1</th>
<th>Matrix of Features of Product Modelling Methods</th>
<th>73</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.2</td>
<td>Comparisons of Process Modelling Methods According to Design Criteria</td>
<td>74</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Comparison of Information Modelling Projects</td>
<td>88</td>
</tr>
<tr>
<td>Table 3.4</td>
<td>Comparison of Software Application Development Tools</td>
<td>91</td>
</tr>
</tbody>
</table>

## Chapter 4

<table>
<thead>
<tr>
<th>Table 4.1</th>
<th>House Project Examples Displaying the Hierarchical Breakdown of Terms in Construction Management</th>
<th>103</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4.2</td>
<td>General Sub/Superstructure Building Activity Sequence for a Housing Unit</td>
<td>109</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Actors Involved with Constructing the Sub/Superstructure Systems for a House Unit</td>
<td>110</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>Diagram Node Index and List of Activities for the Construction Process Model</td>
<td>121</td>
</tr>
<tr>
<td>Table 4.5</td>
<td>A Description of the “Concrete Foundation” Use Case</td>
<td>136</td>
</tr>
</tbody>
</table>

## Chapter 5

<table>
<thead>
<tr>
<th>Table 5.1</th>
<th>Information Details for Foundation Wall Design</th>
<th>177</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 5.2</td>
<td>Summary of how the Required Features for HIPPY were Implemented</td>
<td>186</td>
</tr>
</tbody>
</table>

## Chapter 6

<table>
<thead>
<tr>
<th>Table 6.1</th>
<th>List of Forms and Commands Accessible from the Switchboard</th>
<th>193</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 6.2</td>
<td>Details about Industrial Evaluators and their Organisations</td>
<td>217</td>
</tr>
<tr>
<td>Table 6.3</td>
<td>Details about Researchers and their Previous Organisation (if applicable)</td>
<td>217</td>
</tr>
<tr>
<td>Table 6.4</td>
<td>Responses to Questions</td>
<td>220</td>
</tr>
<tr>
<td>Table 6.5</td>
<td>General Comments on HIPPY</td>
<td>223</td>
</tr>
<tr>
<td>Table 6.6</td>
<td>Hippy and the Objectives for the Information Models</td>
<td>226</td>
</tr>
<tr>
<td>Acronyms</td>
<td>Full Name</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Activity-X Data Objects</td>
<td>ADO</td>
<td></td>
</tr>
<tr>
<td>American Institute of Steel Construction</td>
<td>AISC</td>
<td></td>
</tr>
<tr>
<td>Application Programming Interfaces</td>
<td>API</td>
<td></td>
</tr>
<tr>
<td>Architect, Engineering and Construction</td>
<td>AEC</td>
<td></td>
</tr>
<tr>
<td>Building Project Model</td>
<td>BPM</td>
<td></td>
</tr>
<tr>
<td>Business Process Re-engineering</td>
<td>BPR</td>
<td></td>
</tr>
<tr>
<td>CIMsteel Integrated Standards</td>
<td>CIS</td>
<td></td>
</tr>
<tr>
<td>Client Requirements Processing Model</td>
<td>CRPM</td>
<td></td>
</tr>
<tr>
<td>COMMIT Information Management Model</td>
<td>CIMM</td>
<td></td>
</tr>
<tr>
<td>Common Object Requested Broker Architecture</td>
<td>CORBA</td>
<td></td>
</tr>
<tr>
<td>Computer Aided Design</td>
<td>CAD</td>
<td></td>
</tr>
<tr>
<td>Computer Aided Engineering</td>
<td>CAE</td>
<td></td>
</tr>
<tr>
<td>Computer Aided Software Engineering</td>
<td>CASE</td>
<td></td>
</tr>
<tr>
<td>Computer Integrated Construction</td>
<td>CIC</td>
<td></td>
</tr>
<tr>
<td>Computer Integrated Manufacture</td>
<td>CIM</td>
<td></td>
</tr>
<tr>
<td>Computer Models for the Building Industry in Europe</td>
<td>COMBINE</td>
<td></td>
</tr>
<tr>
<td>Concurrent Engineering</td>
<td>CE</td>
<td></td>
</tr>
<tr>
<td>Construction Design and Management</td>
<td>CDM</td>
<td></td>
</tr>
<tr>
<td>Construction Modelling and Methodologies for Intelligent Information Integration</td>
<td>COMMIT</td>
<td></td>
</tr>
<tr>
<td>Data Access Objects</td>
<td>DAO</td>
<td></td>
</tr>
<tr>
<td>Data Exchange Protocols</td>
<td>DEPs</td>
<td></td>
</tr>
<tr>
<td>Data Flow Diagram</td>
<td>DFD</td>
<td></td>
</tr>
<tr>
<td>Document Object Model</td>
<td>DOM</td>
<td></td>
</tr>
<tr>
<td>Document Type Definitions</td>
<td>DTD’s</td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Data Exchange DDE
Electronic Data Interchange EDI
European Union EU
Extensive Markup Language XML
Facilities Management FM
General AEC Reference Model GARM
General Construction Object Model GenCOM
Generic Building Product Class Model GBPCM
Generic Building Product Information Model GBPIIM
Generic Building Product Model GBPM
Generic Construction Process Model GCPM
Generic Construction Process Modelling Method GEPM
Generic Design and Construction Process Protocol GDCPP
Graphical Reengineering Analysis Design Environment GRADES
Graphical System Modelling Language GRAPES
Graphical User Interface GUI
Gross Domestic Product GDP
Hyper-Integrated Product and Process Information Modelling System HIPPY
Hypertext Markup Language HTML
Industry Foundation Classes IFC
Information Integration for Construction ICON
Information Reference Model of AEC IRMA
Information Structure Diagram ISD
Information Systems IS
Information Technology IT
Information-Integration for Construction Model ICON
<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBPM</td>
<td>Integrated Building Process Model</td>
</tr>
<tr>
<td>ICAM</td>
<td>Integrated Computer Aided Manufacturing</td>
</tr>
<tr>
<td>ICE</td>
<td>Integrated Construction Environment</td>
</tr>
<tr>
<td>IDEFØ</td>
<td>Integrated Definition Ø Methodology</td>
</tr>
<tr>
<td>IEM</td>
<td>Integrated Enterprise Modelling</td>
</tr>
<tr>
<td>IAI</td>
<td>International Alliance for Interoperability</td>
</tr>
<tr>
<td>JIT</td>
<td>Just-In-Time</td>
</tr>
<tr>
<td>LPM</td>
<td>Logical Product Model</td>
</tr>
<tr>
<td>MIM</td>
<td>Management Information Model</td>
</tr>
<tr>
<td>MS</td>
<td>Microsoft</td>
</tr>
<tr>
<td>MOPO</td>
<td>Models of Construction Process</td>
</tr>
<tr>
<td>NIAM</td>
<td>Nijssen's Information Analysis Method</td>
</tr>
<tr>
<td>OLE</td>
<td>Object Linking and Embedding</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>OOA</td>
<td>Object Oriented Analysis</td>
</tr>
<tr>
<td>OODBMS</td>
<td>Object Oriented Database Management Systems</td>
</tr>
<tr>
<td>OOGUI</td>
<td>Object Oriented Graphical User Interfaces</td>
</tr>
<tr>
<td>OOPL</td>
<td>Object Oriented Programming Language</td>
</tr>
<tr>
<td>OOP</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>OO</td>
<td>Object Oriented</td>
</tr>
<tr>
<td>ORB</td>
<td>Object Requested Broker</td>
</tr>
<tr>
<td>ODBC</td>
<td>Open Database Connectivity</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PAIC</td>
<td>Process Activity Information Component</td>
</tr>
<tr>
<td>PBIA</td>
<td>Process Based Information Architecture</td>
</tr>
<tr>
<td>PP</td>
<td>Process Protocol</td>
</tr>
<tr>
<td>Acronyms</td>
<td>Full Forms</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Process Specification Language</td>
<td>PSL</td>
</tr>
<tr>
<td>Product Definition Units</td>
<td>PDU</td>
</tr>
<tr>
<td>Quality Function Deployment</td>
<td>QFD</td>
</tr>
<tr>
<td>Query By Example</td>
<td>QBE</td>
</tr>
<tr>
<td>Role Activity Diagram</td>
<td>RAD</td>
</tr>
<tr>
<td>Simultaneous Prototype for an Integrated Construction Environment</td>
<td>SPACE</td>
</tr>
<tr>
<td>Small to Medium sized Building Project</td>
<td>SMsBP</td>
</tr>
<tr>
<td>Small to Medium Sized Enterprises</td>
<td>SMEs</td>
</tr>
<tr>
<td>Standard Generalised Markup Language</td>
<td>SGML</td>
</tr>
<tr>
<td>Structured Analysis and Design Techniques</td>
<td>SADT</td>
</tr>
<tr>
<td>Total Quality Management</td>
<td>TQM</td>
</tr>
<tr>
<td>Unified Approach Model</td>
<td>UAM</td>
</tr>
<tr>
<td>Unified Modelling Language</td>
<td>UML</td>
</tr>
<tr>
<td>Virtual Reality</td>
<td>VR</td>
</tr>
<tr>
<td>Visual Basic for Application</td>
<td>VBA</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

This chapter introduces the research study by describing the current state and nature of the construction industry. It briefly describes some general trends of importance to the development of information technology (IT) in construction, and outlines current problems in construction by means of assertions, structured as a chain of reasoning. It also presents a number of management improvement philosophies, which seem to offer the solution to some of these problems. The final section outlines the research aim and objectives, and presents a guide to the thesis.

1.2 BACKGROUND

The construction industry is one of the largest sectors of the economy in any industrialised Country. The value added by the industry (including construction product manufacturing and related services) constitutes over one quarter of the industrial sectors, and contributes almost a tenth of the UK’s gross domestic product (GDP). The industry employs around 1.5 million people. This estimated 10-12% of the work force also accounts for 40% of the gross domestic fixed capital formation in building dwellings and other new building works, and is presently one of the strongest in the world, with output ranked in the global top ten (DTI, 2002). The construction industry is currently experiencing its best period of sustained economic growth since the late 1980s having apparently been unaffected by the September 2001 terrorist attacks. Figures recently released by Construction Forecasting and Research indicate that the first quarter of 2002 saw new building orders up by 18% quarter on quarter and 2% year on year, while output increased 3% on the previous quarter and 8% on the same period in 2001 (CFR, 2002). However, this buoyancy and strong indications of continued growth within the sector is putting considerable strain on construction management to cope with the ever-increasing building methods, skills and performance demands. Indeed, construction management capability presents a significant potential constraint on there ability to deliver projects over the next few years.
Nevertheless, there is a growing awareness of the need for changes within the construction industry in its current practices and processes of project development, which include design, procurement, construction, and project delivery (Khalfan, 2001). The following issues have mainly caused this:

1. The dramatically decreasing construction costs through standardisation of construction processes (CIRIA Report, 1999);
2. The rising requirements for project functionality through growing competition;
3. The rapid developments in communication and information technologies (IT); and

The increasing involvement of clients and the corporate management of those building organisations involved with the delivery process requires an assurance that organisational and project procedures are being clearly determined, monitored and followed. Consequently the efficiency and competitiveness of the construction industry are major concerns, and have led to the recognition of the importance of reductions in the project costs, duration and improved quality throughout the building delivery process. This combined with the growing complexity of buildings and their components, contractual arrangements, information management strategies, and the involvement of several contractors providing a variety of resources and services have resulted in new pressures on the management of building projects. These pressures are similar in nature to those experienced by other industries such as manufacturing where the reduction of product development time is critical to the success of the product. However, while other industries, such as the car industry, have been able to achieve significant improvements in productivity and quality over the last few decades, the construction industry seems to have been at a standstill.

This standstill in relation to improvements in productivity and quality is a significant concern given the Government’s pledges on improving the industry’s performance with regard to completion on time and to budget embodied within the recent *Rethinking Construction* (1998) and subsequent *Accelerating Change* (2002). These government-backed initiatives have called for radical improvements in the performance of the industry, that is, the way it performs its primary activity (i.e. the construction of building and civil engineering works) based on the adoption of techniques (such as Value Engineering, Total Quality Management, Lean Production, Quality Function Deployment, Business Process
Reengineering, etc.) and emerging information technologies (IT) developed within the manufacturing and other sectors. A succession of Government and institutional reports has examined this activity, particularly the practice of construction management, and have all commented upon the need for improvement and change (Emmerson, 1962; Banwell, 1964; British Property Federation, 1983), with each highlighting similar problems areas. Fundamentally, the need for improvement is related to the poor performance commonly associated with construction. Typically, this performance is measured in terms of cost, time and quality.

This was also reaffirmed in the results of a study carried out by Latham (1994), who reported on the fragmented nature of the industry as a major factor contributing to the poor communications between all parties working on a construction project. The results revealed that 30% of the total building costs could be saved when information problems such as reducing variations and confrontation (e.g. rework, overlapping work, false information, etc.) are solved. Improved data exchange and the overall management of the information will be key solutions to this, and thereby improving the performance of the construction delivery process. However, in the UK construction industry, there does not exist a means by which to do this. The current perception is that flexibility is difficult within the process of construction because the supply chain changes for every project, and relationships are dynamic (i.e. the client on one job may be a competitor on the next). Despite this, it can be argued that the underlying project delivery process remains broadly consistent. The capacity of the construction management function to fully understand and respond to these pressures and emerging technologies will affect the ability of the UK construction industry to continue to improve its performance and remain at the forefront of the world's best performing sectors (DTI, 2002). The need to provide an information framework that is capable of implementing those emerging computer information technologies and systems is therefore evident.

1.3 PROBLEM DOMAIN

A building evolves over a project's life cycle through several stages from design and construction to operation, maintenance and demolition. The parties involved during these stages, particularly with regard to design and construction, includes a variety of individuals such as clients, contractors, specialists, suppliers, and construction managers, all exchanging large quantities of complex information at different times and locations. The
type of information and extent of detail would clearly be related to the project and the parties involved but would typically include the management of time, cost, quality, health and safety, environmental impact, and the exchange of information and communications (i.e. administrative, technical, financial or legal information, and the systems and procedures that they give rise to).

The information management methods used in current construction processes are inadequate to cope with these demands (Atkin, 1995). In particular, the traditional almost 'water-tight' separation of design and construction causes duplication of work, inconsistent documentation, etc. Product and process models can provide the necessary management information framework for implementing computer systems for the architecture, engineering and construction (AEC) industry. Although the focus of these models is slightly different, both are needed to provide a foundation for managing project information during the design and construction phases. Design 'product' information based on the building elements and components, needs to be integrated with the construction management tasks, the 'process' information necessary to build the components.

However, modelling construction projects requires the application of a diverse palette of object entities. For example, the physical or logical entities which make up the final construction product such as the overall finished facility itself, its systems and components; the various required resources, including materials and equipment; people and organisations, and process information such as contracts, schedules, standards, etc. The organisation and application of these entities can be viewed in terms of the level at which decisions are being made; that is, there is a construction hierarchy that dictates the way in which construction projects are organised and performed. This basically means that different managers use different approaches or techniques (strategies) to realise a facility, instigating further problems in construction management, and provoking a response to which approach or technique can be seen as being the correct choice. Therefore, one of the most fundamental questions confronted by construction managers is 'what construction methods should be selected'.

The scale and magnitude of available IT and information systems for construction are diverse. New modelling methods (or languages) are continuously being developed for use in the building sector and contribute to an ever-expanding diversity of construction techniques. Many of the methods currently being used did not exist 30 years ago. The post-
war era has seen the development of a staggering variety of models for representing product or process information that are changing the way in which facilities such as buildings, bridges, and roads, are being modelled. What is needed is the development of modelling techniques and the adoption of the new and advanced information technologies in the construction arena, thus leading to a greater concentration on improving methods for the information management of integrated product and process models.

However, bewilderment and fragmentation is still the main characteristic on the integration front in the construction industry, and the lack of integration between different stages in the construction process has been widely recognised as a major problem, with a typical small-to medium sized building project (SMsBP) involving up to six or more different professional disciplines (Betts, et al. 1995). As a result, the productivity and quality of the industry is being adversely affected and the ability to fully satisfy the requirements of its clients is being hampered. This has led to various problems including 'inter alia': an adversarial culture, the lack of integration, communication and collaboration between the various disciplines, the fragmentation of design and construction information (with data generated at one stage not being automatically available for re-use downstream), and the lack of real life cycle analysis of projects, including costing, safety assessment, risk management, maintenance, etc., (Anumba & Evbuomwan, 1997). The availability of timely, accurate and complete production information is critical to the construction site decision-making process and its long-term decision effectiveness. It is now recognised that the use of computing technology and the adoption of new business processes based on Concurrent Engineering (CE) principles will provide a valuable platform for overcoming these problems, and improve the competitiveness of the construction industry. This has led to various efforts to develop tools and techniques for its implementation (Kamara, 1999; Khalfan, 2001). Amongst other things, this requires the development of an integrated product and process modelling system that offers the potential for improved collaborative working, with all members of the project team working together at the early stages of a construction project to resolve key 'downstream' issues, with shared ownership of design rationale and decision making (Anumba, et al. 1999).

Previous studies have focused on modelling either the product or the process, without adequate consideration of the implications of one on the other (Anumba, et al. 1998; Kimmance, et al. 2002). Indeed many research projects (some based on European initiatives) have been devoted to the description of the product to be designed or achieved
with the aim of providing an *automated way* of designing, archiving and exchanging data (Dubois, et al. 1995). However, this description has proved inadequate since, although enabling a good implementation of software tools, it neither provides visibility of the roles of the actors involved in the construction process nor improves the information exchanges. Furthermore, the processes necessary to generate the product are largely ignored.

The inadequate infrastructure that exists for seamless project team communications has its roots in the fragmentation of the construction industry. The use of disparate computer-aided engineering (CAE) systems by most disciplines is one of the enduring legacies of this problem and makes information sharing and exchange between project team members difficult and, in some cases, impossible (Anumba, et al. 1999). One of the main characteristics of concurrent engineering is the intensive information interchange in the early stages of design and construction. This includes sharing information about products and processes and resources required in their production. In concurrent engineering, where communication is even more important, flawless information logistics (getting the right information to the right place in the right format at the right time) is becoming mandatory. The integration of product and process modelling will help to overcome this problem by enabling construction project teams to collaborate on the basis of a shared integrated project model. This is in line with the recommendations emanating from recent reviews of the UK construction industry, notably the Latham Report (Latham, 1994), the DOE/BT Report on Construction IT (DOE, 1995), the Egan Task Force Report (DETR, 1998), Rethinking Construction (1998) and Accelerating Change (2002) Reports.

Nevertheless, many of the existing computer-aided tools and domain-specific applications are automating specific industry processes and generate data structures in a proprietary format, which reinforces the fragmented nature of the construction industry. Furthermore, there is no established systematic approach for data collection during the construction phase of projects. However, despite the fact that much of the information generated during the design phase of a construction project is useful for the project delivery process (i.e. on-site construction stage), experience has shown that construction managers typically receive this information as incomplete and often inaccurate, or is available, but not accessible. In addition, the information is limited to computer-aided design (CAD) drawings, and other supporting information such as specifications, schedules, etc., and is largely in paper format. As a result, the problem encountered is two-fold:
Chapter 1

- Inefficiency in handling information due to the following reasons:
  - either too little or too much information;
  - redundant or inaccurate information (e.g. is not relevant to the task at hand); and,
  - incomplete and inconsistent information.

- Acquired information is in non-electronic format, which requires re-entering.

The main consequences of this inefficiency within the design and construction phases are that decisions are taken without the right information support, resulting in delays and increased costs.

This was also reaffirmed by a study undertaken at the Construct IT Centre of Excellence (Construct IT, 1996), in which the current use of IT and the importance attached to that use in managing the construction site process by many of the leading construction companies in the UK was compared. The results of this study revealed that there is widespread use of IT on the construction site. However, this was mainly used to automate the traditional paper-based information systems. Word processors, spreadsheets, databases and to some extent CAD systems, are the most commonly used computer tools on the construction site. Applications that supported activities in one specific process area are also used. For example, planning and scheduling software can be used for monitoring progress, resource levelling, process simulation, and for determining the consequences of a delay. Although, the study displayed the use of such advanced systems in 'best practice IT', the advanced features of such systems are often not applied (Froese, et al.1996).

Several other studies on IT have described the inefficiency of applications during the construction phase of projects (Betts, et al. 1997; Anumba, 1998; Jägbeck, 1998; Svensson, 1998; Dado & Tolman, 1998). The main conclusion from these studies reaffirms that IT is often used only to automate the paper-based information systems, and that electronic communication within one, or between two or more, process-areas is supported, but that more often communication is performed verbally or by paper-based means using telephones and fax machines. Other conclusions reveal that:

- Companies exchange data electronically on some projects, mostly by e-mail or on disk, but there is no significant use of electronic communication for direct data exchange within the whole project team.
Most companies discerned the importance of 'integration' for better communication, and that some companies demonstrated some degree of integration, current solutions only support communication between applications in one specific process area. As there are no neutral standards in this area, the integration achieved has been provided by one software vendor or based on a specific company development.

It is appropriate to mention that not only the construction industry, but also other industries (such as manufacturing) are increasingly challenged with growing competition, reduced product life cycle, and changing market. To meet these demands, organisations within these industries are forced to develop new products, which have to be cheaper, delivered faster and provide greater functionality (Khalfan, 2001). The product development process in many organisations, however, presents many problems with respect to the required product quality, time-to-market, and cost. To achieve a better performance level, a new or better configuration of processes and technology, consisting of people and means, is needed. The overall purpose of this study is to develop new approaches and techniques that will enhance the performance of on-site construction operations.

1.4 INTEROPERABILITY PROBLEMS

What is happening is the construction industry as a whole is also happening on the construction sites. Most operations and tasks contain the basic work processes in construction and are supported by IT, but most supporting systems are not integrated. Communication therefore is largely done by traditional means, mainly formalised on paper. In order to understand the underlying reality of poor information management during the construction stage of projects, it is important to recognise some characteristics of IT on the construction site. The following section outlines some of the current problems associated with IT on the construction site (Construct IT, 1996; Dado & Tolman, 1998; Svensson, 1998).

A probable cause for the inefficient use of IT on the construction site is the lack of 'integration', which is technically caused in a number of ways. For example, a primary concern is that downstream IT applications (occurring later in the construction process) cannot reuse the information produced by earlier applications in a digital format; however, system operators and/or end users need to interpret the information manually so as to input it again at a later stage. Another related feature is that information produced by an earlier
application may be structured differently in comparison to a later application. As a result, the cost of recreating the information is potentially high. A further concern is that many current IT applications for the management of information during the construction phase of a project have been developed from existing IT systems, and are often created as a second application, to sit on top of an existing system. The problem here is that they have inherited many specific features from the original system. Furthermore, the few systems that have been developed from inception are often created on an ad hoc basis without a thorough investigation or co-ordinating phase at the beginning of the system's development, so as to secure a solid foundation for appropriate functionality. Also, developed IT systems in design and construction have generally been carried out within a local environment, based on a specific company's needs, without any real consideration or ambition to work for standardisation and open systems. Finally, many of these information system developments were carried out, or at least started (about 10 years ago), before modern, open and standardised IT solutions existed. The lack of integration in construction, in particular, on-site construction can be summarised as follows:

**Between different construction management applications:**
A way of illustrating this is through different systems using information concerning products and processes (i.e. objects, classes, activities, functions, tasks, etc.), project information such as planning and scheduling, time dependencies, durations, etc., construction, design and management (CDM) regulations, health and safety, cost management etc., all use information relating to the building delivery process, but they seldom all use the same source for this information.

**Between applications used in different areas of a construction organisation:**
This relates to the lack of integration between IT systems within construction organisations and the different domains or parts of the enterprise. For example, information used on different levels of management (e.g. operational, tactical or strategic) or the information used for different types of building functions (e.g. design, production, estimating, project management, etc.) within the organisation.

**Between different generations of construction applications (integration of old and new):**
A significant current problem is the poor interoperability of new applications with existing ones (i.e. integration between computer aided engineering systems). Ideally, the new IT systems for construction site application integration should be flexible and adaptable, and
be able to co-exist with minimal effort with earlier applications. The existing discontinuity between different information systems generates extra cost to both the client and construction management/or contractors.

**Between different types of information used on the construction site:**
A lack of integration between the different types of information used during the construction phase of a project, such as:

- paper-based (non-digital) information and data in digital format;
- design information (e.g. drawing plans, specifications, etc) and on-site activity information (e.g. operation sequence and scheduling) always seem to conflict;
- graphical and non-graphical (e.g. alpha-numerical information); and
- between the different classifications standards (e.g. mixing different classification classes, such as SMM7, CI/SfB, IFCs, etc.).

**Between design and construction applications and applications for other domains such as facility management:**
The lack of integration between IT systems could also include areas outside the management of information during the construction stages of a project. When evaluating and discussing construction management, a major concern is the lack of integration with systems used during the design and construction phases of the life cycle of the building. Ideally, information from the design and construction stages of the building delivery process should be transferred in digital form to downstream domains, such as phases of the facility lifecycle. What is more, a substantial amount of knowledge can be gained during the facility management processes, this information will be of considerable interest when designing, planning and constructing new buildings. Therefore, information should be exchanged in both directions and between several processes, such as briefing, design, construction, and facility management (i.e. the integration of construction site applications).

In summary, the lack of integration generates problems such as: multiple systems needing to be operated in parallel, information redundancy, deleting and updating information (poor data distribution to IT systems), poor analysis of information and limited feedback functions. The need for changes within the industry in its current practices and project processes is therefore evident. As a result, there is an increasing awareness among
construction professionals at every level of the need for efficient, coordinated information management methods, planning procedures, and integrated modelling techniques that will effectively deal with the complexity, and the demanding responsibilities that are assumed for any construction project. Therefore, what is effectively needed is an integrated information system capable of managing both product: objects, components, etc., and process: site operations; activities, tasks, functions, etc., during the design and construction stages of construction projects.

1.5. POSSIBLE SOLUTIONS TO THE PROBLEMS

In order to tackle some of the problems outlined in the preceding sections, a few pioneering companies in the manufacturing and construction industries have, during the last few decades, explored possible ways to improve performance, productivity, and the efficiency of the product development process. However, within many organisations, these improvements in project development process, productivity, etc., present many problems with respect to the required product quality, time-to-market, and cost. To achieve better performance levels, new and better configuration of processes and technology, is needed. A number of general management philosophies, often using IT as an enabling technology, have been devised to solve the problems encountered (Koskela, 1992; Miyatake, et al. 1992; Anumba & Evbuomwan, 1997; Kamara, et al. 1997; Kimmance, 1999; 2000a).

Examples of general management philosophies and techniques, which in principle could be applied to construction, are Value Engineering, Quality Control Management, including Quality Assurance, Total Quality Management (TQM) and Quality Function Deployment (QFD), Just-In-Time (JIT) Production, Business Process Re-engineering (BPR), Supply-Chain Management and Lean Production. Other techniques for process improvement, which are more specific to the construction industry, are Performance Driven Construction, Critical Path Analysis and Work-Study approaches, and systems for the management of design and construction (e.g. Turnkey contracts, Partnering, Design and Build, etc). Although many of these approaches represent useful and pragmatic techniques for improving the project development process, they do not however, appear to be producing the desired effects to meet the growing demands of industry. Consequently, further work has been undertaken to examine the integration of the life cycle phases in the project development process to reduce product lead-times, changing market and customer demands (de Graaf, et al. 1996; Tummala, et al. 1996; Anumba, et al. 1998). This has led
to many construction companies responding to the increasing importance of project development processes by incorporating Concurrent Engineering (CE) techniques to improve their project development capacity (Khalfan, 2001).

CE is a management philosophy, which embodies several other methodologies such as multi-disciplinary teams, parallel scheduling of activities and cross-functional problem solving (Kamara, 1999). It is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. The approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from concept through disposal, including quality control, cost scheduling, and user requirements (Winner, et al. 1988). According to Evbuomwan & Anumba (1996), where CE is applied to construction projects, it seeks to optimise the design of a facility and its construction process to achieve reduced lead times, and improved quality and cost by integrating design, fabrication; construction and erection activities; and by maximising concurrency and collaboration in working practices. Aspects of this approach are increasingly common and are known by such terms as fast tracking or design and build depending on the way it is implemented. A vital aspect of the adoption of CE principles in construction is the need to structure and standardise information at the semantic level; that is, an effective communications infrastructure that facilitates seamless inter-working between the disparate actors involved in construction projects. Such an infrastructure needs to be based on the latest information and communications technologies and should facilitate information interchange between members of the project team and across stages in the project lifecycle (Anumba, et al. 1998; 2000). A more detailed discuss on CE has been reported by Kimmance, (2000a). It has been suggested that a generic integrated modelling framework should inhabit the central focal point of this communications infrastructure (Anumba, et al. 1998; Kimmance, et al. 2002a).

In order to identify a proposed remedy to the lack of information integration in construction projects, the following assertions, which have been structured as a chain of reasoning, and bear a significant relevance for the research work in this thesis, are presented.
Chapter 1

Assertion One: The need to structure and standardise information at the semantic level is necessary for data exchange between IT applications.

A possible solution is the development of higher-level information models (e.g. conceptual product and process models, reference or application models, domain or aspect models, etc.), with a view to identifying, formally defining, and structuring the semantic concepts intrinsic to the capture and exchange of product and process information related to information management during the design and construction stages of a project. A suitable standardised description of the project delivery process using a neutral modelling language such as the Unified Modelling Language (UML) or the emerging Process Specification Language (PSL), which could function as an interchange language to integrate IT, and product and process information throughout the construction process would facilitate the integration and communication. Another important reason for proposing standardised approaches, as a solution to the integration problems is that on-site construction is primarily concerned with the many functions (e.g. activities, operations, tasks, etc.) related to constructing the actual facility (e.g. a house). The different functions have in common that they all exist to support the top-level context function (e.g. construct house). Each function has relationships and links to information that describe the purpose of on-site construction, such as resources (materials, products, elements, etc) equipment, plant, and environmental issues. This implies the following assertions:

- On-site construction is generally concerned with the construction processes, and the utilisation of resources needed to carry out these processes. These resources, such as products and sub-product (foundation, floor, roof, etc.), elements and components (doors, windows, masonry materials, etc.) and services (water, gas, phone systems, etc.) can for example be classified or grouped into building systems, parts or classes.
- The site construction process is closely linked with other construction process phases, such as design; production, and facility management processes of buildings, which also have to contend with the building systems or parts described above.
- The information used in design and construction phases, is to a certain degree, of multimedia type with complex interrelationships (alphanumerical data, drawings, standards and specifications, graphical representations, photographs, video, etc.), and a standardised digital model of the delivery process (constructed building from start to finish) could act as a mechanism to integrate such heterogeneous data.
Assertion Two: A proposed remedy for the lack of integration is the consistent use of appropriate data structures:

There are several developments, which can contribute to solving the interoperability problems discussed above. Their features make it easier for end users of information systems to contribute in the overall development of new systems, provide facilities for learning new systems and how to use them (Svensson, 1998). These developments include:

- Industry-wide standardisation efforts, such as STEP and the International Alliance for Interoperability (IAI) with their Industry Foundation Classes (IFC).
- Standardised computer hardware (PCs, laptops, etc.) and computer-aided software engineering (CASE) tools.
- Distributed object techniques, like CORBA, object linking and embedding (OLE) and hyperlinking, local (LAN) and global (WAN) networks and WEB technologies.
- System analysis, prototyping and object-oriented technology, such as object client-server network techniques, object databases and standardised graphical user interfaces.

However, while many of these developments (described in greater detail in Chapter 3) promise to simplify the design and implementation of IT applications, as well as facilitate the user interfaces for the end user, many focus on specific technology or application domains, such as client requirements (briefing), design, planning, and facility management phases. Therefore, to a certain degree are only part of the solution for data exchange and data sharing between applications.

Assertion Three: Object-oriented approaches can constructively be applied to the definition of the standardised building process description (i.e. a generic building product class model is an appropriate solution).

The basic concept of object-oriented systems is that they model real world entities as a collection of objects in a data-centred oriented way, rather than a procedure or function-oriented way. Objects can be easily defined, designed, implemented and maintained. This basically means that the system development process becomes more intuitive and controllable. Object-oriented approaches offer benefits not just to the system development process, but also to the utility and flexibility of the resulting software systems. Recent
developments in building product modelling (reviewed in Chapter 3) indicate a broad consensus among researchers who applied an object-oriented methodology to product modelling; thus, offering a promising approach for defining standardised information structures for supporting the exchange and sharing of data. Practitioners and software developers are also increasingly sharing this view. One additional benefit of applying an object-oriented approach is that the traditional technique for classifying information in the construction industry with respect to design and construction processes could be used throughout the building delivery process.

Assertion Four: Suitable process descriptions provide a foundation for the development of integrated information systems for on-site construction.

The on-site construction phase is carried out through many processes. These processes are usually represented by a series of activities, operations, tasks, etc., which add value to the overall project managed by the construction manager. The information requirements and constraints of the construction processes are generally described through process modelling, and many different techniques for modelling processes exist. Examples of techniques for analysing and describing processes are the data flow diagram (DFD), role activity diagram (RAD), Petri Net, and IDEFO diagrams, etc., presented in Chapter 3. A further example and possible solution to this is also presented in Chapter 4.

1.6 AIM AND OBJECTIVES OF RESEARCH

Against the background discussed in this chapter, the aim of this research was to develop a practical and easily adaptable integrated product and process information modelling system (HIPPY), for on-site construction. The aim was to be achieved through the following specific objectives:

- to investigate the role of system integration by reviewing product and process information;
- to review and evaluate the use of product and process models in the construction industry;
- to investigate current modelling practices and standards in the construction industry, in order to draw comparisons with similar practices from other industries, both in terms of product and process representation and model content;
• to review and evaluate application development tools and information system requirements to support a suitable integrated information structure;
• to develop a conceptual integrated product and process model for on-site construction,
• to implement a prototype information system capable of managing product and process information essential to construction managers, and to evaluate the developed prototype system.

1.7 SCOPE OF RESEARCH

It is not possible to model all the information needs of the construction industry within the life of a single research project. Therefore, in order to be able to develop modelling systems, which are robust and detailed enough to support the proposed applications it is necessary to establish the scope of the modelling activity. This research focuses on the activities and information flows between the design and construction stages of small-to-medium-sized building projects (SMsBP), with particular focus on the on-site construction stage, using house-building as a test bed. The approach adopted in this research and the resulting integrated information modelling system could be extended to cover the lifecycle of a construction project. In the light of this, the scope of the thesis is limited primarily to construction information models (product and process modelling) and to those methods related to and used or proposed for construction process modelling, although the concept or principles could be used in other areas of modelling as well.

1.8 JUSTIFICATION FOR THE RESEARCH

The implementation of a conceptual integrated product and process model for construction, which directly addresses the problems of fragmentation in construction, is of major significance to the construction sector. This is because of the problems associated with fragmentation, such as: lack of IT and integration, communication and collaboration between the various project disciplines, lack of true life-cycle analysis of projects (including costing, safety assessments, maintenance, etc.), loss of design information, variations, claims and disputes, an adversarial culture, etc., which lead to dissatisfied clients whose requirements are less likely to be met (Anumba & Evbuomwan, 1997; Kamara, 1999). An integrated product and process information modelling system has the potential to make construction projects less fragmented, improve productivity and quality,
thus improving the performance with regards to completion on time and to budget. It is therefore important to provide an integrated information model to bridge the gap between product and process information during the on-site construction stage. The development of a integrated conceptual model for processing both product and process information to support the implementation of a prototype system, will not only encourage those involved with the construction to use and add to design information, but also provide richer information representation, better efficiency and data consistency, and the flexibility to support information management during the on-site construction stage of SMsBP.

1.9 THESIS STRUCTURE

This thesis comprises of seven chapters covering the main stages in the research. A summary of the content and purpose of each chapter is presented below.

Chapter One: Introduction

This chapter provides the general background to the research study by describing the nature and current state of the construction industry. It then briefly describes some general trends of importance to the development of information technology (IT) in construction, and outlines current problems in construction by means of assertions and identifies a number of management improvement philosophies, which seem to offer the solution to solve some of these problems. The final section outlines the research aim and objectives, and presents a guide to the thesis.

Chapter Two: Research Methodology

This chapter describes the research methodology, both in terms of an argument for and an explanation of the chosen method, the 'system analysis approach'. The first part reviews current research methods and examines the underlying theory of the adopted research process, described in part two. The Chapter ends with a description of the research process and how it satisfies and reflects the established theory of research.

Chapter Three: State of the Art Review of Modelling Techniques

This chapter discusses the contemporary perception of the design and constriction process, as a basis for its development and improvement. It builds on the previous chapters by reviewing the state-of-the-art in IT development methodologies, product and process
modelling techniques, integrated project models and object-oriented approaches to system integration. Enabling software tools and technologies are surveyed and other research related to this study is reviewed.

**Chapter Four: Integrated Conceptual Model**

Chapter Four provides a description of the conceptual integrated product and process model. The features and characteristics are also discussed along with a description of the building delivery process, using the principle of 'general to specific' for describing the information requirements of the integrated information system.

**Chapter Five: Implementation of the HIPPY Prototype System**

The means of integrating product and process information is addressed in this chapter by discussing the implementation of a prototype system (HIPPY) for managing the processing of product and process information essential to the construction manager during the on-site construction stage of a small-to-medium-sized building project (using a house as an example), in order to establish automated and controlled information flow. The chapter also presents an overall description of the scope and objectives of HIPPY, as well as the resource constraints and principles of software prototyping. A brief description of the potential practical application of the system is also discussed.

**Chapter Six: Operation and Evaluation**

This chapter demonstrates the application of HIPPY to on-site construction operations, and provides an evaluation of its practical application. The actual requirements for a house-building are used, as an example to demonstrate the use of the prototype system. The approach adopted for the evaluation is also reported.

**Chapter Seven: Conclusions and Recommendations for Further Work**

Chapter Seven presents the conclusions of this research project. It discusses the contributions made by this research, and makes recommendations for practical applications and future research.
CHAPTER 2

RESEARCH METHODOLOGY

2.1 INTRODUCTION

The primary aim of this chapter is to provide an overview of the research methodology used in the research described in this thesis. The first part presents a brief review of current research methods and examines the underlying principles and theory of the research process. The second part of this chapter presents and justifies the research method adopted in this study. The chapter ends with a description of the research process followed and demonstrates how it satisfies and reflects the established theory of research.

2.2 THEORY OF RESEARCH

All research is considered to be a process of enquiry or examination designed to discover information or relationships, which is usually influenced by its environment, and is 'historic' since it is impossible to collect and analyse data simultaneously (Easterby-Smith, et al. 1994). Research is simply a method or tool for investigating or collecting information (getting information from point A to point B). A research method is a strategy of inquiry which moves from the underlying philosophical assumptions to research design and data collection. The choice of research method influences the way in which the researcher collects data. Specific research methods also imply different skills, assumptions and research practices. The objective of this section is to briefly discuss generic research issues and identify research approaches, so as to enable informed decisions to be made at a later date.

2.2.1 Generic Issues in Research

Research can be undertaken in many different ways. A research approach is based on a set of basic beliefs (or metaphysics) derived from philosophical assumptions in the field of ontology, the study of being; epistemology, the study of knowledge and the acquisition of valid knowledge; methodology, the study of research method; and axiology, the theory of
value (Guba & Lincoln, 1994; Fitzgerald & Howcroft, 1998). Each academic or professional domain typically makes wider use of some methods than others; develops local ground rules for dealing with concerns about ‘reliability’ and ‘validity’; and, quite often, invents technical jargon for the use of insiders (investigators). Nevertheless, many of the basic problems in conducting good research remain the same everywhere. It is the human inventiveness in response to those fundamental difficulties that works constantly to alter the face of how research is conducted. In the past two decades, the number and complexity of research methods has increased sharply, particularly with the advent of information technologies (IT) within the construction domain. Where once only a few forms of inquiry were available (and acceptable within the scientific domain), many options now exist. This proliferation offers more than just a larger choice for the researcher; it also makes possible a better matching of research tools to the demands of each particular question, and this is an enormous advantage for all researchers.

Research is often judged by the degree to which the results might be generalised. Generalisation is closely linked to validity and reliability; this is, the extent to which researchers are able to use their method to study what they had sought to investigate rather than something else, and also, how well the research has been carried out. The following are some important issues that epitomise research study:

Validity: This should be central (depending on what approach is used) to the whole issue of the cohesion in relation to the research work between conceptual framework methods (e.g. system analysis and modelling approaches), questions and findings. If these methods, approaches and techniques generally fit and measure the issues, which have been investigated then the findings are likely to be valid.

Reliability: Research is considered ‘reliable’ if another researcher carrying out the same research activities with the same kind of group or methods would be likely to replicate the findings, although these finding need not be identical.

Generalisation: One of the main generic issues (elements), which distinguish between good levels of research, particularly PhD research, is the generalisability of the research findings. If generalisation of the findings is required, a choice between statistical or analytic generalisation needs to be made.
• **Statistical generalisation** is possible when a deduction can be made about a population on the basis of empirical data collected from a representative sample. The application of statistical laws and formulae can be used to establish the confidence level.

• **Analytic generalisation** is possible when the test results support the theory but the phenomena tested do not claim to be a representative sample of the whole population. Normann (1970), for example, states that using an approach such as case studies, it is possible to generalise even from a single case provided that it is founded in the comprehensiveness of the measurements, which make it possible to research a fundamental understanding of the structure, process and driving forces rather than a superficial establishment of correlation or cause-effect relationships.

Therefore, generalisation has two dimensions. One is **quantitative**, based on a large array of observations, surveys or measurements to determine how much, how often and how many. The other is **qualitative** and is based on exhaustive investigations and analysis to identify certain phenomena that are suspected to exist in similar situations.

Other important issues when planning and undertaking research include: **Reasoning** (deductive, inductive or abductive), and **Bias** and **Rigour**.

**Deductive reasoning** begins with a general principle or rule (considered by most as fact), and assesses a specific case that seems to fit the rule or principle. This process entails reasoning from the general to the particular (specific) and is useful in determining cause and effect relationships.

**Inductive reasoning** requires that theoretical constructs be formulated after and on the basis of, observation and fieldwork. Theory is grounded in the empirical world. Delbridge and Kirkpatrick (1994), for example, mention that central to the idea of grounded theory is the establishment of rigorous *ideal type* models, which are closely aligned with the empirical world and not deductively imposed upon it. The theory emerges from the data rather than being imposed upon it.

**Abductive reasoning** is the process of generating the best ‘explanation’ from a collection of observations, the explanation being the relation between one or more ‘hypotheses’ and the datum or data the hypotheses account for (Peng & Reggia, 1990). Therefore, **abductive reasoning** is a process of both generating hypotheses and selecting the best one (or some) for further evaluation.
Bias has been described as 'seeing what we want to see, misperception, mis-interpretation and making too much of ambiguous data' (Miles & Huberman, 1994). Bias can be classified as holistic fallacy (e.g. putting more logic and coherence into events than is justified), or elite bias (e.g. allocating greater importance to data provided from articulate, high status informants, and less value to data from lower status informants). Bias can arise at any stage of the research process, although academic research is not necessarily negated by bias, the researcher must be aware of the sources and reduce it as much as possible. The inevitable distortion in the collection, analysis and interpretation of data are controlled through the application of rigour, and are given different priorities by different research philosophies (Kitchener, 1994).

Rigour is the term used to define the value and extent to which the research method and techniques employed strictly adheres to the fundamental requirements of research analysis and design (Bennett, 1991). The research is generally characterised by hypothetical-deductive testing according to the positivist paradigm, with emphasis on internal validity through strict experimental control and quantitative techniques.

2.3 TYPES OF RESEARCH

2.3.1 Quantitative versus Qualitative Research

Research methods can be classified in various ways, although traditionally research followed the natural sciences approach; confidently predicting events on the basis of research. In contrast to this scientific approach is the recently developed naturalistic or hermeneutic (from the Greek 'to interpret') approach, which encourages a broader, thinking about the possibilities, assumptions and values that underlie the research work (Easterby-Smith, et al. 1994). These two approaches are generally dichotomised into either realistic/quantitative (involving or relating to considerations of amount or size) or idealistic/qualitative (relating to distinctions based on quality) research methods. However, because qualitative research is often defined by how it differs from quantitative research, it may be helpful to compare the two approaches. A major difference lies in their fundamentally different assumptions about the goals of research. Babbie (1983), for example, has defined qualitative analysis as “the non numerical examination and interpretation of observation for the purpose of discovering underlying meanings and patterns of relationships”, as opposed to quantitative research, “the numerical
representation and manipulation of observations for the purpose of describing and
explaining the phenomena that those observations reflect". Qualitative methods lend
themselves to discovering meanings and patterns while quantitative methods seek causes
and relationships demonstrated statistically, a theoretical perspective, positivism that is
concerned with facts, prediction, and causation and not the subjective nature of the groups
or individuals of interest. Researchers in the qualitative mode seek understanding through
inductive analysis, moving from specific observation to the general. Quantitative analysis,
on the other hand, employs deductive logic, moving from the general to the specific, i.e.,
from theory to experience. Taking these definitions a step further, the difference might be
summarised by saying that quantitative research is structured, logical, measured, and wide.
In contrast, qualitative research can be considered to be more intuitive, subjective,
unstructured, and deep.

The differences between the two approaches, then, result because each is defining
problems differently and each is looking for different solutions or answers. This implies
that some areas are best investigated using quantitative methods; whilst for others,
qualitative approaches will generate better results. However, the majority of research
methods can accommodate a qualitative and quantitative approach, depending on the
subject and research procedures adopted. In some cases, both methods using a hybrid
approach called 'triangulation' can be used to obtain the best results (Markus, 1994;

As well as the methodological level of qualitative and quantitative distinction, there are
other distinctions, which are commonly made on the epistemological level. Research
methods have variously been classified as Objective Vs. Subjective (Burrell & Morgan,
1979), as being concerned with the discovery of general ‘nomothetic’ laws Vs. being
concerned with the uniqueness of each particular ‘idiographic’ situation, as aimed at
prediction and control Vs. aimed at explanation and understanding, or as taking an outsider
(Etic) Vs. taking an insider (Emic) perspective. Considerable controversy continues to
surround the use of these terms, and indeed on the axiological level; for example external
validity of action research questions and its relevance to practice (Relevance) Vs. internal
validity through tight experimental control and quantitative methods (Rigour). However, a
discussion of these distinctions is beyond the scope of this thesis. For a more detailed
discussion see (Luthans & Davis, 1982; Morey & Luthans, 1984; Fitzgerald & Howcroft,
1998; Myers & Avison, 2002).
2.3.2 Qualitative Research Methodologies

Qualitative research is conceived to be entirely different from quantitative approaches, and was initially developed in the social sciences to enable researchers to study social and cultural phenomena (Coombes, 2001). In a sense, qualitative research is also defined by the research methodologies or procedures employed to obtain the subjective data that form the basis for analysis and further understanding. Adoption of this method produces ‘soft’ research in which the subject of investigation is not independent of the research process and enquiry is hermeneutic or a never-ending process. Qualitative research (or idealistic research) typically entails in-depth analysis of relatively few subjects (small-scale research) for which a rich set of data is collected and organized. It is a process, with guidelines rather than rules, of reasoning, of matching descriptions and choosing to accept or reject any observation, condition or statement on the basis of the researcher’s own interest.

Nevertheless, qualitative research is now deployed in many fields of study, and its influence in the natural sciences has been growing steadily. It also has undergone a recent period of rapid diversification, with the creation of a number of distinctive research traditions, or sub-categories. While social scientists differ widely about the categorization of qualitative research and the terminology used to describe it, there are three philosophies that appear to be the principal categories currently employed: positivist, interpretive, or critical (Guba & Lincoln, 1994). However it needs to be said that, while these three research epistemologies are philosophically distinct (as ideal types), in the practice of qualitative research these distinctions are not always so clear-cut (Lee, 1989). There is considerable disagreement as to whether these research ‘paradigms’ or underlying epistemologies are necessarily opposed or can be accommodated within one study. It should be clear from the above that the word ‘qualitative’ is not a synonym for ‘interpretive’ since qualitative research may or may not be interpretive, depending upon the underlying philosophical assumptions of the researcher. It follows from this that the choice of a specific qualitative research method (such as the case study method) is independent of the underlying philosophical position adopted. These underlying philosophical approaches are briefly described below:

**Positivist Research:** In this kind of study the researcher generally assumes that reality is objectively given and can be described by measurable properties, which are independent of the researchers and their instruments. Positivist studies generally attempt to test theory, in
an attempt to increase the predictive understanding of the problem. In line with this approach, Orlikowski & Baroudi (1991) classified Information Systems (IS) research as positivist if there is evidence of formal propositions, quantifiable measures of variables, hypothesis testing, and the drawing of inferences about a phenomenon from the sample to a stated population. This method is usually used with 'hard systems thinking' and is concerned with the 'how' of the problem (e.g. to develop or modify a system in some way so that the assumed goal is achieved in the most efficient way).

**Interpretive Research:** In this kind of 'soft' study approach, the researcher builds an extensive collection of detailed records concerning context, human activities, people, actions, and the perceptions of participants, as the basis for inductive generation of explanatory theory (Seymour, et al. 1997). Walsham (1993) perceived interpretive research as the attempt to understand events through the meanings that people assign to them (no universal truth), and also states that interpretive methods of research in information systems are “aimed at producing an understanding of the context of the information system, and the process whereby the information system influences and is influenced by the context”. Klein & Myers, (1999) suggest a set of principles for the conduct and evaluation of interpretive research for developing information systems.

**Critical Research:** This begins with a number of assumptions that differ sharply from those made by people working within other qualitative research traditions. Researchers assume that social reality is historically constituted and that it is produced and reproduced by people. Although people can consciously act to change their social and economic circumstances, critical researchers recognize that their ability to do so is constrained by various forms of social, cultural and political domination. The main task of critical research is seen as being one of social critique, whereby the restrictive and alienating conditions of the status quo are brought to light. Critical research focuses on the oppositions, conflicts and contradictions in contemporary society, and seeks to be emancipatory; that is, it should help to eliminate the causes of alienation and domination.

However, just as there are various philosophical perspectives, which can inform qualitative research, there are, various qualitative research approaches (forms) associated with the interpretive and critical science perspectives. These include: naturalistic, hermeneutics, ethnographic, ethnmethodological, phenomenological, post-positivist, subjective, artistic, case study, grounded theory, humanistic, ecological, action research, participatory,
Each research method discussed above uses one or more techniques for collecting 'empirical' data (many qualitative researchers prefer the term 'materials' to the word 'data' since most qualitative data is non-numeric). These techniques range from interviews and questionnaires, from unstructured to structured, observational techniques such as participant observation and fieldwork (direct), through to archival research, and unobtrusive research, including the examination (study) of documents, and the researcher's impressions and reactions. Written data sources can include published and unpublished documents, company reports, memos, letters, reports, email messages, faxes, newspaper articles and so forth. However, data is said to be qualitative if the observations belonging to it are separate and distinct (e.g. the number of entities stored in 'a' table or the number of object controls in 'a' interface form). All qualitative data is inherently discrete, in that there are a finite number of possible categories into which they may fall, and can be further classified as nominal if there is no natural order between the classifications or categories.

Although a clear distinction between data gathering and data analysis is commonly made in quantitative research, such a distinction is problematic for many qualitative researchers. For example, from a hermeneutic perspective it is assumed that the researcher's presuppositions affect the gathering of the data – the questions posed to informants largely determine what one is trying to find out. The analysis affects the data and the data affect the analysis in significant ways. Therefore, it is perhaps more accurate to use the phrase 'modes of analysis' rather than 'data analysis' in qualitative research. These modes of analysis are different approaches to gathering, analysing and interpreting qualitative data. The common thread is that all qualitative modes of analysis are concerned primarily with textual analysis, whether verbal or written. Although there are many different modes of analysis in qualitative research, the main ones include construct analysis and grounded theory. Others include: hermeneutics, semiotics, and approaches, which focus on narrative and metaphor (Strauss & Corbin, 1990; Hirschheim & Newman, 1991; Myers, 1994).

Table 2.1 presents an overview of these traditions, or sub-categories of qualitative research. It describes the purposes, commonly used nomenclature (forms), and specific techniques associated with qualitative research. A more detailed discussion of these
philosophies and methods can be found in (Orlikowski & Baroudi, 1991; Miles & Huberman, 1994; Yin, 1994; Wass, 1994; Guba & Lincoln, 1994; Baskerville & Wood-Harper, 1996; Fitzgerald & Howcroft, 1998; Coombes, 2001; Myers & Avison, 2002).

Table 2.1: Purpose, Common Forms, and Examples of Research Techniques used in Qualitative Research

<table>
<thead>
<tr>
<th>Epistemological Level</th>
<th>Positivist or Interpretive (fixed laws vs. no universal true)</th>
<th>Critical Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose:</td>
<td>Understanding a situation from the perspective of the participant</td>
<td>The understanding and critique of power within society</td>
</tr>
<tr>
<td>Common Forms:</td>
<td>• Ethnography • Constructivism • Phenomenology • Participant Observation • Interpretive Interactions • Hermeneutics • Case Study • Action Study</td>
<td>• Feminist • Marxist • Critical Ethnography • Deconstruction • Postmodernism • Post Structuralism • Foucaultian • Emancipatory</td>
</tr>
<tr>
<td>Examples of Research Methods:</td>
<td>• Observations and use of field notes • Examinations of documents • Interviews</td>
<td>• Analysis of print materials, popular culture, and social structures • Documentation of empowerment activities, often using interpretive research techniques</td>
</tr>
</tbody>
</table>

2.3.3 Quantitative Research Methodologies

As the name implies, quantitative research deals with things that can be counted, as it often entails the proper application of statistics; that is, the statistical manipulation of numbers to process data and summarise results to typically a large number of subjects. At the methodological level, quantitative methods are associated with 'hard' research methods, whereby they help to make generalisations to larger groups and follow a well-established and respected set of statistical procedures of which the properties are usually well understood. When properly applied, quantitative research is arguably much more statistically powerful than its (qualitative research) counterpart. Quantitative research methods were originally developed in the natural sciences to study natural phenomena. The deeper assumptions that underpin this research tradition lie in the particular form of
philosophy called positivism; the belief that the world conforms to fixed laws of causation, where complexity can be tackled by reductionism, and emphasises objectivity, measurability and repeatability of variables (Lock, et al. 1998). The basis of the scientific, or quantitative approach is that the researcher should remain independent and distant from the research process; that is, stands apart from the subject and observes an independently existing reality. Quantitative enquiry aspires to certitude, to the idea of describing conditions perfectly thereby enabling acceptance or rejection on the basis of a perfect fit between those conditions. This is achieved by adherence to a strict set of rules and requires the ability to control circumstances, conditions and external influences (Wass, et al. 1994).

Quantitative research is by far the oldest type of research, and its capacity to describe, predict, and explain multiple disciplines (such as mathematical, biological, social, and psychological) has provided a significant part of the foundation on which the natural sciences have been erected. According to Locke, et al. (1998), the main branches of the quantitative family include: descriptive; correlational/predictive; quasi-experimental; single-subjects; and the recent addition of meta-analysis. These approaches can be further broken down to include survey methods (term used to generalise the techniques of sampling, interviewing and questionnaire); laboratory experiments; questionnaires; structured observations; numerical methods such as statistics; mathematical modelling (graphical pictures or diagrams), and formal methods (e.g. econometrics).

When quantitative data is collected through questionnaires or observations using analytical data techniques (such as numerical formats, coding, variables, etc.), the data is said to be discrete, that is, the data must be able to be counted, for example the number of ✓ (yes) and ✗ (no) for a given question, or the number of companies who use Microsoft Office in a given time period (a week). The data is also said to be continuous if the measurements take on value often within the same range (e.g. the number companies with less than 10 employees (small to median size firms) who use Microsoft Office in a week).

Table 2.2 presents the names and purpose of some commonly encountered formats associated with three broad categories of quantitative research: correlational, descriptive, and quasi-experimental/experimental. It is acknowledged that single subject and meta-analysis (multiple cases) research are not included in the table, as these represent special cases, and are only briefly described along with the other categories in the following subsections:
**Single Case Research (design):** This type of research is often categorised as experimental by researchers who are usually developing a new theory, instead of testing, developing or proving an existing theory nor to establish statistical generalisation.

**Meta-Analysis Design:** This is a relatively new approach, which enables researchers to combine studies that have the same focus so as to derive a single result. Multiple case studies rely on replication to create theory: the same results are predicted from each case. This approach normally requires a level of environmental control usually only achieved in laboratory conditions.

**Descriptive Research:** This form of research captures and displays a graphical representation or picture of some aspect(s) of a situation, usually expressed in numbers. For example “what is the level or amount of data that can be stored in a relational database table?” “How long does it take to develop a table?” “What kind of data is stored in object-oriented databases?” “What kind of journals/reports is read by students studying IT/software engineering courses?” These are the type of questions, which call for descriptive studies.

**Correlational Research:** This type of research examines the nature of the relationship between variables such as simple (statistical methods that yield a single number), predictive (improves capacity to anticipate events), and modelling (path analysis, graphical maps, structural analysis using boxes with connecting arrows, etc) (Locke, et al. 1998).

**Quasi-Experimental or Experimental Research:** This category includes a very large variety of related research ‘design’ methods. The feature that ties them together is the inspection of data to determine whether two or more groups differ on some variables, involving conditions under which the investigator does not control one or more of the critical variables and may not be able to use random procedures to select or assign all subjects and treatment. Although true experiments, researchers not only choose the treatment, but they also select subjects, assign subjects to groups and, finally, assign treatments to groups, which are controlled by random procedures (allowing chance to control selection and assignment).
Table 2.2 Techniques used in Quantitative Research (Adapted from Locke, et al. 1998)

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Descriptive</th>
<th>Correlational</th>
<th>Quasi-Experimental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>The description of a sample on a specific variable.</td>
<td>Describing relationships among variables;</td>
<td>Testing of standard deviation (differences between) group averages and means for one or more independent variables</td>
</tr>
<tr>
<td>May also describe sub-samples on the same variable</td>
<td>Predicting a criterion variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Testing a model of the interrelationships among variables used to predict a variable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commonly used Research Formats</th>
<th>Descriptive</th>
<th>Correlational</th>
<th>Quasi-Experimental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey Research</td>
<td>Survey Research</td>
<td>Predictive Methods</td>
<td>Casual Comparative</td>
</tr>
<tr>
<td>Political Polling</td>
<td></td>
<td>Multiple Regression</td>
<td>Repeated Measures Design</td>
</tr>
<tr>
<td>Delphi Survey</td>
<td></td>
<td>Casual Modelling</td>
<td>Within and Between Design</td>
</tr>
<tr>
<td>Statistical Analysis</td>
<td></td>
<td>Path Analysis</td>
<td>Analysis of variance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System Analysis</td>
<td>Analysis of Covariance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Facture Analysis</td>
<td>Randomised Block Design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bivariate Correlation</td>
<td>Multivariate Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples of Research Approaches</th>
<th>Descriptive</th>
<th>Correlational</th>
<th>Quasi-Experimental</th>
</tr>
</thead>
<tbody>
<tr>
<td>[For all Three Research Types]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data collected with instruments</td>
<td>Data collected with instruments</td>
<td>Data collected with instruments</td>
<td></td>
</tr>
<tr>
<td>Surveys</td>
<td>Surveys</td>
<td>Surveys</td>
<td></td>
</tr>
<tr>
<td>Attitude measurements</td>
<td>Attitude measurements</td>
<td>Attitude measurements</td>
<td></td>
</tr>
<tr>
<td>Use of statistics</td>
<td>Use of statistics</td>
<td>Use of statistics</td>
<td></td>
</tr>
</tbody>
</table>

2.3.4 Hybrid Research Approaches (Triangulation)

Although most researchers carry out either quantitative or qualitative research work, some researchers have suggested integrating one or more research methods in the same study, creating hybrids or more commonly ‘triangulation’ methods (Mingers, 2001). These terms are used in a research context to describe the use of a variety of data sources or methods to examine a specific phenomenon either simultaneously or sequentially in order to produce a more accurate account of the phenomenon under investigation. Hybrid basically means approaches, which do in themselves, constitute a combination of qualitative and quantitative elements. These elements may be so closely ‘packed’ as to be practically indistinguishable (e.g. a systematic content analysis, which combines the qualitative coding of text with the quantitative calculation of numbers or coefficients of variants). Triangulation is an approach that combines independent yet complimentary research methods. It is a term borrowed from the study of experimental methods (instrumentation).
and refers to the application and combination of several research methods, in the attempt to investigate the same phenomenon (research topic). Triangulation, also known as 'mixed methods' was developed to counteract the inherent threats to validity, generalisability and reliability that each experimental research method contained (Gable, 1994). Also, by combining multiple methods, it is used to overcome the weakness or intrinsic 'biases' and the problems, which may arise from single research methods, such as single-observer and single-theory studies. There are two principal categories: simultaneous triangulation and sequential triangulation:

- **Simultaneous triangulation** refers to using both qualitative and quantitative methods at the same time. An example of this is using both a survey methods and a case study.

- **Sequential triangulation** (sequencing) refers to using the results of one method for planning the next method; that is, both quantitative and qualitative methods are employed within one and the same study, although in different phases of the research process (Bowen, 1996). The most common example would be a qualitative phase of data collection, which is followed by a quantitative phase of data analysis (e.g. using an exploratory pilot study before beginning an experimental design), as in the case of interviews which are coded and for which coding frequencies are determined. Alternatively, data analysis might involve the construction of types by means of cluster analysis, the reduction of categories to a smaller number of dimensions by means of multiple correspondence analysis, etc.

There are also five basic techniques of triangulation, which can be used simultaneously or sequentially. These include:

1. **Data Triangulation**, involving time, space, and persons;
2. **Investigator Triangulation**, consist of the use of multiple, rather than single observers;
3. **Theory Triangulation**, which consists of using more than one theoretical scheme in the interpretation of the phenomenon;
4. **Methodological Triangulation**, which involves using more than one method and may consist of within-method or between-method strategies;
5. **Multiple Triangulation**, when the researcher combines in one investigation multiple observers, theoretical perspectives, sources of data, and methodologies.
There are many benefits to using triangulation, such as minimising the inadequacies of each method by using different methods, and using methods to complement each other, in addition to providing stronger and more reliable findings. Full discussions of triangulation can be found in (Ragin, 1987; Gable, 1994; Markus, 1994; Mingers, 2001).

2.4 METHODOLOGY ADOPTED

The overall research methodology was adapted to address the aim and objectives described in Chapter 1. It has also attempted to emphasise the human activity aspects of developing an information system, while utilising a scientific approach for developing a pragmatic computer information system. Both aspects; that is, the human and computer elements are inter-related. The research undertaken endeavoured to use an approach that was sympathetic to the issues being investigated.

A methodology for the domain of information systems (IS) covers a number of research aspects, although coverage varies from one to another. Avison & Fitzgerald (2001), describe a methodology as a collection of many components. Typically, each methodology has procedures (e.g. research approaches such as object-oriented analysis or structured system analysis and design), a set of research techniques and notations (e.g. the unified modelling language (UML) or IDEF0) that support the research approach, tools for modelling to structure the development process (e.g. a life cycle model–spiral incremental) and documentation aids (repositories and databases) that facilitate the development of an information system. There is usually also some type of underlying philosophy that captures a particular view of the meaning and purpose of information systems development. To provide the necessary contingency-based, but integrated, research methodology to accommodate these different demands in a coherent and consistent way (such as the one described above), an overall research model was developed. This section discusses the research approaches and techniques adopted in this project.

2.4.1 The HIPPY Research Model

The overall research model is shown in Figure 2.1. The first box represents the unifying research philosophy, which guides the research approaches and techniques. The research approach consists of the dominant theory generation and testing methods, while the research techniques comprise documentation, data, and information collecting forms and
modelling tools. The research model’s elements generated a framework, which provided an interactive portfolio of research approaches and techniques used in this study, and which benefited from meta-level direction and cohesion. Each of the model’s elements will be briefly discussed.

Figure 2.1: The HIPPY research model

2.4.2 Research Philosophy

The philosophical view of this study was based on an open system of thought where scientific data was critically examined and generalised, in order to specify the limits of the generalisations. According to Phillips & Pugh (1994), there are four important elements to structuring a scientific research PhD thesis: the background stage, focal stage, data stage...
and the contribution stage. The research described in this thesis has sought to comply with the following guidelines.

- **The Background Stage**: Reviews the literature in the field of study.
- **The Focal Stage**: Describes what the research is about and why it is done.
- **The Data Stage**: Justifies the relevance and validity of the proposed solution of the research.
- **The Contribution Stage**: Evaluates the importance of the contribution made to the discipline by the research.

The strategic approach of this study followed a 'hybrid' approach, whereby several different methods were used together to make a sound strategy (Bowen, 1996). Since the aim of the project was not only to develop a practical and easily adaptable integrated product and process information modelling system (HIPPY) for on-site construction, but also to develop a conceptual integrated model with CE implementation strategies for the industry. Therefore, triangulation was the most appropriate method for this purpose because it combines both qualitative and quantitative methods within the same study.

### 2.4.3 Research Approach (information system development)

In analysing and developing information systems, much will depend on the organisation, its legacy systems, the extent of existing manual and computerised systems, and future system requirements. The analysis and design of systems can be carried out in a variety of ways, and several methods have evolved over the years (Avison & Fitzgerald, 2001). Many of these methods are well documented (Yourdon, 1989; Graham, 1994;) and are covered in numerous texts (DeMarco, 1979; Jackson, 1983; Chung, 1989, Martin, 1989; Booch, 1991; Coad & Yourdon, 1991; Jenkins, 1994; Yeates, et al. 1994). In developing software systems, many previous and current systems have been developed for construction using a variety of techniques such as, the traditional systems development life cycle approach (Lee, 1979), and also partly used in this thesis, the structured rapid-prototyping approach (Isensee & Rudd, 1996), and the object-oriented approach for software development (Ambler, 1998; Coad & Yourdon, 1991). For example, in practice, each method incorporates particular tools and techniques for information collection (e.g. feasibility study & systems investigation), documentation and diagrammatic representation (e.g. systems analysis & model systems design), implementation and review, in addition to
automated tools to assist in the processes involved. Nevertheless, while no one method can
be employed for modelling every problem domain, this research has adopted a *systems
approach* and employed useful characteristics (properties and regulations) from different
methods to develop the integrated information system (HIPPY). These methods, which
include the systems analysis and design and object-oriented analysis, will briefly be
discussed.

### 2.4.4 Systems Approach

An important part of scientific research involves the methodology of how to approach and
solve a problem in a certain domain or application area (i.e. the problem of developing a
new prototype information system). One of the main targets in problem solving is to obtain
a general view of the problem area, not simply one single part. This statement stems from
Aristotle’s dictum that the whole is greater than the sum of the parts, and forms the base to
the approach adopted in this thesis. The systems approach supplements the traditional
analytical approach (Lee, 1979), whereby it integrates the analytic and the synthetic
method, encompassing both holism and reductionism (Bertalanffy, 1968). It was first
proposed under the name of ‘General System Theory’, which handles the description of
systems, their properties and regulations (characteristics) by using formal, mainly
mathematical, concepts and modelling methods. According to Ackoff (1971), a system is a
set of interrelated elements, which have a set of inputs going into it, a set of outputs going
out of it, and a set of processes that convert the inputs to the outputs. The transformation
of input into output by the system is usually called throughput.

In order to utilise a validated approach that is iterative in practice (occur concurrently), this
project has adopted the concepts of the systems development life cycle (Lee, 1979) and the
procedure for scientific work described by (Ackoff, 1971), and applied them in
establishing the systems requirements and to provide an interface to designing and
implementing an information system’s prototype. In this research, the term ‘systems
analysis’ is used to cover both the analysis and design aspects of the proposed integrated
information system.
2.4.5 Systems Analysis

Systems analysis provides an approach for describing, analysing, and planning complex systems, and is primarily based on two concepts: the *system* and the *model* (Ackoff, 1971; Svensson, 1998; Karhu, 2001; Avison & Fitzgerald, 2001). According to Schoderbek, *et al.* (1990), a system could here be defined as "a set of objects together with relationships between the objects and their attributes related to each other and to their environment so as to form a whole". This is universal and could be applied to almost any field of reality.

The systems analysis and design phases were iterative by nature, and followed a similar procedure for scientific work proposed by Ackoff, (1962). This included the following stages:

- Problem Perception and Formulation
- System Design and Modelling
- Problem Solving and Validating the Model Solution
- Implementation and Evaluation of the Solution

In the work described in this project, the problem perception and formulation, i.e. the background or data gathering stage, including the recognition of the problem area, was performed in parallel with the development of the conceptual product and process models and studying the existing product and process modelling techniques. An extensive review of the existing literature, written on modelling techniques, software development applications, and management philosophies, such as CE, and their implementation within construction was based on examination of documents (personal document analysis method) and surveys. A comparison of these product and process models, and software development tools were carried out through the qualitative method. The methodology was also infused with an *action* research element, in as much as the researcher participated in the process under study, in order to identify, promote, compare, and evaluate problems and potential solutions. According to Fellows & Lin (1997), action research is typically directly aimed at influencing practice. The action research method was used to identify problems in existing modelling systems, techniques, and software tools, which prevent them to be used in the construction industry. The method is then used to develop a solution in the form of conceptual product and process models, which is applicable to the industry through the modification and evaluation of existing modelling techniques, tools, and project systems.
The research techniques that are often used in systems analysis and design research projects, and also in this thesis, for gathering the background theory and developing an integrated product and process information modelling system are briefly discussed in the next section.

2.4.6 Research Techniques

The research methods used to identify the use of IT applications for on-site construction and to analyse various improvement strategies, which were used to develop an integrated information system included: action research, literature studies, interviews (including observation), surveys and evaluation, questionnaires, and a multi-faceted approach (triangulation).

2.4.6.1 Literature Review

This entailed an extensive literature study of primary, secondary and tertiary sources, to appraise existing works on various product and process modelling techniques, the use of IT/IS applications in construction, software development applications, associated research project, business improvement, and formal management philosophies in order to obtain a greater understanding of the research area. The specific aim of the review was:

1. To obtain a greater understanding of the research area, in order to establish the system requirements needed to develop product and process models.
2. To enable the research thesis to build on and use the work and experiences of others.
3. To demonstrate to third parties that the research project is aware of and, where appropriate, firmly grounded in previous work in this area.

2.4.6.2 Interviews with Domain Experts

Interviews with construction industry practitioners have played an important role during all phases of this research project, especially the interviews with software companies and other personnel involved in manufacturing and construction, combined with specific input from other research projects, process observations, work studies and workshop meetings. The interviews were semi-structured in nature to allow the interviews to have an overall
Chapter 2

The purpose, but were sufficiently flexible to explore issues as they arose during the interview sessions.

2.4.6.3 Surveys

Four surveys on integrated project databases; product and process modelling techniques and available software development tools (e.g. Microsoft Visio, Rational Rose, System Architect, etc.) have been undertaken. The surveys and analysis (evaluations) cover the main approaches to the development of product, process and integrated project models for construction, related research projects, and software development tools, which have been specifically developed to support the various types of information modelling techniques, including product and process, static and dynamic, class and object, and many other modelling types. The synthesis of the various surveys is based on a pre-defined set of criteria. The criteria were intended to provide a common platform for comparison and, while every effort has been made to obtain the relevant information related to each survey evaluation, there are still gaps in the information available. The surveys and analysis aimed to identify the main projects that have addressed the development of integrated product and process information systems, and explore the use of IT in construction, in addition to providing a comparison of modelling techniques and IT applications tools used in the construction industry. More specifically, the aim of the surveys was to:

- identify the appropriate modelling techniques, software tools, and to evaluation their capabilities;
- establish project/system specific criteria; and
- to prioritise the criteria, and if possible, to map the criteria to the techniques and tools' capabilities.

2.4.6.4 Triangulation Method

This research project has adopted the use of a multi-faceted approach, generally referred to as triangulation (Bowen, 1996). The aim of the research project was to develop an integrated product and process information modelling system, based on CE principles, and compatible with existing information systems in construction. In order to develop the product and process models, qualitative ‘interpretive’ research methods are used for building an extensive collection of detailed information relating the research domain.
These methods included a review of existing literature (examinations of documents) on the research (theory) domain, and semi-structured interviews with domain experts. Use was also made of established ‘descriptive’ survey techniques in the form of four surveys (previously mentioned in the last section). A comparison of these product and process modelling techniques and software development tools is also carried out through the qualitative method, and is covered in Chapter 3. The action research method is used to identify problems in the modelling techniques and existing tools, and is also used to develop possible solutions in order to create information models, which are applicable to the construction industry through the selection of the appropriate software. The results from the surveys also generated solutions for the development and implementation of the information models (described in Chapter 4).

Research theory generation and trialling have been conducted in an iterative manner throughout the life cycle of this research and a quantitative correlational (hard) approach has been used to examine the relationships between the anticipated events (variables) and research theories (analysis of data). Prototype development and implementation involved the design and development of a prototype information system for industry; to be tested (validated) in a selected heuristic way (e.g. by using a test panel of practitioners (experts) and questionnaires), so as to allow subsequent modification and recommendations in light of any findings (Wroe, 1986). Recommendations on using and evaluating questionnaires and presentation of results are carried out through qualitative review of literature on these topics and through document analysis method. Microsoft software applications were used to develop information models and associated software for the models. Information modelling and surveys, their evaluation and presentation of data were carried out through quantitative research methods. The development of information models and CE implementation strategies would be based on the results from the surveys, prototype software, questionnaire, and a qualitative review of existing literature.

Its underlying premise of triangulation is that the weakness of a given research strategy can be compensated by counter-balancing strengths of another. Thus, triangulation attempts to exploit the strengths and offset, rather than compound, the weakness of different methodologies. The results from such an approach can then be triangulated, reducing bias, increasing validity and revealing a more comprehensive picture. Triangulation achieves this by cross-referencing between different sources of data, such as documentation
evidence, observation, surveys and questionnaires and ways of interpreting it. The active pursuit of validation through triangulation is demonstrated by combining multiple research approaches and techniques in the overall investigation, such as carrying out interviews, undertaking surveys and developing product and process models, prototyping, presenting and debating the findings at conferences, workshop meetings, generating reports, and publishing journals.

2.4.6.5 Conceptual Modelling

Conceptual modelling has played an important role in this project. Conceptual modelling is a means of conceptualising some well-defined part or entities of the real world. A conceptual model should show the structure of information in these 'mini-worlds', and also show how the various activities in an activity system or model relate to each other. In order to define the structure for the conceptual models, there are basically two approaches that can be followed (Young & Dorador, 2000): top-down and bottom-up approaches. In the work described in this phase, both the top-down and bottom-up approaches were used in parallel to develop the integrated conceptual model.

The utilisation of a structured analysis and design technique IDEFØ, a top-down modelling approach was adopted, in order to determine the information process levels necessary for the development of a construction process modelling system, and was iterative in nature, to ensure its relevance to, and practical application in, the construction industry. This involved three distinct stages: review of modelling techniques and tools, identification of requirements (scope and purpose), and identification of on-site construction activities.

After identifying the context and scope (i.e. the information captured and modelled in the IDEFØ construction process model), a bottom-up modelling approach was applied utilising the features offered by the unified modelling language (UML), an object-oriented modelling methodology that enables individuals to model the conceptual entities with a powerful modelling concept: the object. The approach adopted is to break down the conceptual entities (e.g. products, actors, resources, etc) into objects and classes of objects with information attributes. As well as, completing the basic structure of the information models and architecture, with the aim of developing an integrated conceptual model to support the management of information during the on-site construction stage. The
development of the generic building product model also involved an iterative process, to ensure its relevance to, and practical application in, the construction industry.

2.4.6.6 Prototype Development and Evaluation

As part of the research objectives, the development of the prototype software for the product and process models was required to demonstrate the implementation of the models in a computer environment, based on concurrent engineering principles, as well as provide a framework for future integration with other computer-based construction activities. Software prototyping concepts (Avison & Fitzgerald, 2001), and the resource constraints of the research (e.g. time, finance) therefore influenced the development of the prototype software for the models. Use was made of a general-purpose package (Microsoft Access, 2000), as it facilitated a relatively inexpensive and quicker development of the prototype.

The strategy adopted for the evaluation of the prototype was to invite a selection of researchers and industry practitioners, who represent some of the potential users of the prototype software, to participate in a demonstration of the software using the actual requirements for a building project, and also represented in the information models. Evaluators were then requested to assess the effectiveness of the prototype by completing a questionnaire.

2.5 INTEGRATED PRODUCT AND PROCESS MODELLING

For the integrated product and process modelling aspects of the research, the following approaches were used: systems analysis, conceptual modelling techniques, such as the structured analysis and design technique, object-oriented modelling, and prototyping. The research work reported in this thesis contained five major phases: domain analysis and research theory, product and process methodology review, development of theory and construction of product and process models, implement of HIPPY prototype, and research evaluation and conclusions. The research process is schematically described as an IDEF0-model (Figure 2.2), and shows how the systems analysis approach and the procedure for scientific work, described above, compare with the research process in this thesis, and described below. Figure 2.2 also illustrates the steps that have been taken during this research, which has led to the development of the hyper-integrated product and process information modelling system. Each of the five phases includes a number of activities
undertaken in an iterative manner, which basically meant that normally a pair of consecutive activities was run through twice. The research process also provides the logical structure for the thesis, and is described in the following sub-sections:

**Figure 2.2: The research process illustrating the steps for the development of HIPPY**

**Domain Analysis and Research Theory:** The activities within this phase involved observing and analysing the area of interest; that is, the management of information during the on-site construction phase of small to medium sized building projects. This included an extensive review of the construction industry through the use of structured assertions (chain of reasoning) instead of a regular hypothesis. The research requirements were formulated on the basis of this information (introduction and assertions) and the aim and objectives were determined, (see Chapter 1). The problem domain was further analysed through a review of product and process techniques and software.

**Product and Process Methodology Review:** This phase involved reviewing the state-of-the-art in product and process modelling techniques, integrated project databases, conceptual modelling, information systems development, and object-oriented approaches (see Chapter 3). It also includes the information gathered from surveys of industry and evaluation of the basic software tools to create solutions to the research problem.
**Develop Theory and Construct Product and Process Models:** A building product model and a construction process model using UML and IDEFØ were developed (see Chapter 4) from a study of literature on related subjects, and the project modelling/software application tools survey undertaken. An information framework and integrated conceptual model, combining both the conceptual product and process models developed in Chapter 4 were the primary outcomes of this phase.

**Implement HIPPY Prototype:** This stage of the research meant implementing the developed information modelling framework and the integrated conceptual model. In order to develop an information system prototype, the validity in practice of the chosen theories was studied using surveys and extensive literature reviews of related integrated database projects, product and process modelling techniques (reported in Chapter 3). By comparing the results and solutions, one may deduce which ways have been successful and satisfy the requirements. The prototype system's architecture was formulated and the prototype implemented utilising the Microsoft Windows environment. The implementation activities are described in Chapter 5, followed by a brief description of the potential practical application of the system.

**Research Evaluation:** Since the information was generated within a research environment, it would be necessary to validate the practical application of HIPPY on a real test case of a construction project, in order to assess the definition and effectiveness of HIPPY. However, as this was not possible, due to time and financial constraints, a selection of researchers and industry practitioners, who represent some of the potential end-users of the prototype software were invited to evaluate the prototype system through the use of a hands-on demonstration of its use. Testing the validity of the software was mostly carried out by a heuristic's research method. The participants were requested to complete a questionnaire, which was developed to allow them to indicate their opinion on the various aspects of the software being evaluated.

The purpose of the validation is to check that it gives a sufficient description of the system so that the problems can be solved. This means that it is essential to be assured that the prototype system is adequate for its purpose, not necessarily that the system is true or false as such. The research evaluation meant reviewing the results of the prototype demonstration and questionnaire, enabling refinements and conclusions to be made, with
Chapter 2

recommendations for further research. The research evaluation and questionnaire is described in the concluding Chapters of the thesis.

2.6 CHOICE OF SOFTWARE TOOLS FOR MODELLING

At this stage, it is appropriate to mention the particular software tools that have been used in this research project. Following an extensive evaluation, the software tools for process modelling using the IDEFØ method, have been System Architect 2001 and, later, Microsoft (MS) Visio 2002, a general purpose graphical tool. Product modelling using the UML method has been Rational Rose 98 and, later, MS Visio 2002. Process modelling in the form of tasks was carried out using MS Visio 2002, and the conceptual modelling in the latter phase were also carried out using MS Visio 2002. The development and implementation of the prototype was carried out using MS Access 2000, a software application for creating databases and programmable user interfaces. The main reason for using Visio is that it provided the facility to create custom shapes and stencils to support all the modelling standards needed to generate the information models presented in this thesis, in addition to providing the facilities for programmable user interfaces and language/code regeneration. A more detailed description of the software applications and the HIPPY prototype is found Chapters 3 & 5 of this thesis and in the following papers (Kimmance, et al. 2000; Kimmance, et al. 2001; Kimmance, et al. 2002a; Kimmance, et al. 2002b).

2.7 SUMMARY

This chapter has been presented in two parts. The first part briefly reviewed existing research theories and methodologies, in order to provide background understanding for analysis the adopted research approach. The second stage identified important aspects of research and presented the methodology applied in this thesis. The chosen methods were intended to meet the needs of the nature of the research project and its aim and objectives. Comparing and contrasting these methods was used to formulate and validate empirically well-founded theories about managing product and process information in the design and construction stages of SMsBP, using house building as a test bed.

The choice of method(s) depends largely upon the questions or objectives the research is seeking to answer, and can usually be categorised by who, what, where, how and why type questions. At different stages of the research a variety of methods and techniques have
been used to address these research questions, and at each stage care has been taken to ensure the validity of the outcome. A contextual (qualitative) approach is taken in the early stages through the use of assertions and literature reviews, instead of an oversimplified normative (hypothesis) statement being produced in the absence of context. In the latter stages (e.g. modelling and system development) a correlational positivist (quantitative) approach is taken, instead of the less structured non-contemporary subjectivist description, which emphasises the nature of research (e.g. access to limited or dead research sources), rather than normative, objectivity and measurability of the research (e.g. follows a set or given procedure). Both approaches are complementary and minimise the inadequacies of each other, in addition to providing stronger and more reliable findings. The trustworthiness and authenticity of the outcome of the research have been carefully controlled using many of the tactics suggest by Maxwell (1996). Comparison is used at all stages and triangulation of the data from the different sources has been used to verify the theoretical outcome (results) and conclusions of this thesis. Searching for discrepant evidence and negative cases has been an ongoing process at all levels of the investigation and the data has been continuously compared with background theory, surveys and similar cases.

A significant input to the overall formulation and structuring of the research process was based on detailed literature reviews of the manufacturing and construction industries, as described in Section 2.4.6. The next chapter presents the state-of-the-art review of literature related to this thesis.
CHAPTER 3

STATE OF THE ART REVIEW OF MODELLING TECHNIQUES

3.1 INTRODUCTION

One of the major requirements for integration is a standard paradigm for representing and communicating information; that is, a general conceptual model of the data to be represented. This chapter describes the research domain from an information handling perspective; by presenting a review of literature related to the state-of-the-art in information modelling standards, including object and document model based standards, data, product and process model based techniques, and discusses object-oriented approaches for addressing integration in construction. It then explores the advantages of combining these modelling methodologies in the definition of the structure for integrating information models that will support computer-aided IT applications throughout the life-cycle phases of on-site construction operations. The chapter also reviews several research projects within the domain of this research; that is, presents product and process models of construction processes, and integrated construction project (databases) models. The aim of the review is to present the contemporary perception of the construction process, as a basis for it's development and improvement.

3.2 EXISTING MODELLING TECHNIQUES

Modelling is an essential step for understanding and improving the efficiency of construction performance. In addition, the development of suitable credible models is a logical precursor to automate and improve construction processes throughout the whole life cycle of a construction project (Kimmance, 2000c). It is these principles that underline the value of developing valid models. In order to appreciate the underlying reality of modelling the on-site construction process, it is important to examine product and process modelling methods, information models of construction and processes, and models that have been developed using these methods. This section presents a description of the current practices in information modelling, object, data, product and process model based standards/techniques, and integrated project environments within the Architecture, Engineering, and Construction (AEC) industry. It presents several high-level conceptual
(core) product and activity process models developed from several Computer Integrated Construction (CIC) and Concurrent Engineering (CE) projects, including some that are not intended to be core process models but are relevant to this discussion.

### 3.3 INFORMATION MODELLING

The literature within information modelling is comprehensive and has been widely studied by the international research community. This has lead to several views on modelling approaches (Anumba, et al. 1998; Avison & Fitzgerald, 2001; Kimmance, et al. 2002a), resulting in a number of research projects attempting to develop coherent and integrated information models, which facilitate continuous and interdisciplinary data sharing (DETR, 1999; Kimmance, 2000c; Kimmance, et al. 2000). Despite this there is no universally agreed definition of the scope and nature of each approach. There are various techniques, which can be used for information modelling within the (AEC) domain. These techniques can be divided into three principal categories namely; product models, process models and project models (sometimes referred to as integrated product and process models or integrated project database). This section looks at the three principle modelling efforts that are relevant to this research project.

#### 3.3.1 Product Modelling

The literature on product modelling is comprehensive; a lot of this reports on different attempts and approaches in developing product models (Turner, 1990; Froese, 1992; Eastman & Fereshetian, 1994; Baxter, et al. 1994; Eastman, 1994; Luiten, 1994; Björk, 1995; Anumba, et al. 1998; Eastman & Augenbroe, 1998). Despite this there is no universally agreed definition of the scope and nature of a product model. Therefore, in the work described here, a **product model** is a representation of the totality of data elements that define a product over its expected life cycle, and should facilitate the unambiguous exchange of such data between project participants. A product model needs to be structured in such a way that any software application can access and store information within it. Thus supporting applications used in product development that deal with information relating to elements and the relationships between them (e.g. objects, components, classes, entities, etc.), in the overall building facility.
Many product models for construction have been developed in recent years, amongst these are:

- Perhaps the most influential models, based on later research, include the (AEC) Building System Model that is built on a framework called the AEC Global Model, which establishes a layered structure between two different types of product models (Turner, 1990). Firstly, models that are specific, but general (e.g. describing a building project, site, general systems definition, etc.). Secondly, ones that are independent called ‘Common Technology Models’, which include models describing property, attributes, direct network, and enclosed area. Together, these two groups establish a general information framework on which AEC System Models could be built.

- The General (Global) AEC Reference Model (GARM), which organises construction information at a high level of abstraction was proposed by Gielingh (1988). GARM specifies the structure of a product model, based on the idea that the product information is clustered around stages or types of Product Definition Units (PDU), given as a collection of characteristic of the product or part (functional units), and their technical solutions.

- Björk (1989) proposed the RATAS model, an entity-relationship model enhanced with inheritance as a framework. A basic feature of the RATAS modelling framework is an abstraction hierarchy with five levels: building, system, subsystem, part, and detail, all structured to define product data about a specific building, based on compositions and/or decompositions relationships of building components, as illustrated in Figure 3.1.

One important feature of these models was the ambition to provide a framework for a single building product model. Another feature is that they all claim to support all phases of the life cycle aspect of construction projects, with buildings as a focus point.

Recently, a working group of ISO/TC59/SC13 has issued a draft International Standard called ‘Framework for Classification of Information’ (ISO, 1997). The aim is to investigate the feasibility of establishing a conceptual Schema for improving constructions works by applying semantics and ontological theories to define the concepts within classification systems (e.g. objects, things, classes, parts, elements, components, etc.), and to build the conceptual framework for construction works (Ekholm, 1996).
A number of other researchers have focused their proposals on a more refined field.

- Hannus (1990) discusses CAD-Systems based on product modelling and pre-cast concrete structures;
- Serén, et al. (1993) proposed an Object-Oriented CAD Model (OODAC), which is a generic data model based on a composition of objects with part-of relationships;
- Luiten (1994) used pre-cast concrete structures to test product models of beams, columns, hollow core slabs and connections;
- Watson (1995) uses the CIMSTEEL product model for structural steel framing data models; and
- Karhu (1997) proposed a product model based design of Finish pre-cast facades.

Many of these approaches, and indeed much of the current work on product modelling has been carried out by VTT in Finland. They have also been a prime research area for a number of years, and are currently receiving attention from international standards, such as ISO STEP: 10303 (ISO, 1994), the IAI International Foundation Classes (IAI, 1997) and CORBA (OMG, 1995), which are attempting to standardise computer-interpretable representations and exchange of product data.

However, recent work in the area of product modelling suggests that localised product models utilising object-oriented approaches are now perceived as the solution to the
difficulties of representing building components and the relationships between them. This move away from the traditional global product model results from the inability to address the differing information needs of the various disciplines involved in the construction industry. The models at the detailed level are not representatives of reality (inaccurate), because compromises have had to be made (Aouad, et al. 1998). Several efforts have been undertaken in the area of the modelling of spaces and their boundaries (Björk, 1992; Froese, 1996; Svensson, 1998). However, again, these approaches appear to have either modified previous efforts or been produced quickly and do not address issues at a higher level of abstraction. Also, many have been developed for integration purposes between specific project design disciplines, and as such are inadequate to form the basis for a building product model for on-site construction purposes.

3.3.2 Product Modelling Techniques and Data Standardisation Efforts

There are various techniques, which can be used for information modelling including data, product and process modelling. In data modelling, items of interest to the industry are identified and defined (Kagioglou, et al. 1998). A data model provides the basic data structuring mechanisms for describing data, relationships and constraints of the information stored in any information system (Björk, 1992). Using a specific data model, conceptual information models can be developed. A conceptual model specifies the categories of information used in a specific domain or database. Tools used can generally be grouped to include graphical schema languages (e.g. entity-relationship diagrams (ERD), NIAM or IDEF1x diagrams) and data definition languages, using for example EXPRESS, which adds several ideas, such as rules and methods, which cannot be modelled in standard graphical tools. Several methodologies (or languages) can be used for product data modelling, amongst which include:

- Product Model based Standards, such as STEP and its description methods (e.g. EXPRESS, EXPRESS-G), IAI-International Foundation Classes, CIMsteel;
- Object Model Based Standards, such as UML, CORBA, and OMG; and
- Document Model Based Standards (e.g. SGML, HTML, XML, and companion specification such as RDF and DOM).

In order to understand the fundamental concepts of these approaches, a brief description of some of their main components and features would be instructive.
3.3.3 Product Model Based Standards

This section presents the various product-modelling efforts that are relevant to building construction. These include, STEP – (EXPRESS), IFC, CIMsteel, NIAM, and IDEF1x.

3.3.3.1 STEP Technology

The Standard for the Exchange of Product Model Data (STEP) is the informal name of the (ISO Standard – 10303), which is the principal working name for the standardisation of industrial automation systems, and models for product data representation and exchange (ISO, 1993a; Wix, et al. 1999). The primary aim is to provide several significant and revolutionary innovations (standards) capable of describing the product data throughout the life cycle of a product including planning, design, construction, maintaining (e.g. the complete house cycle), independent of any particular system. The basic structure of STEP attempts to reconcile two diametrically opposed objectives:

- Defined a set of data elements that are unambiguous, notably the interpretation and use of data structures in the different application domains, such as various implementation methods for handling, storing, exchanging, and archiving product information.
- Define a set of data elements that are manageable, few in number, robust (e.g. stable over time), and flexible; this is, can be used in many ways.

From an architectural perspective, STEP meets these objectives through a number of developments. The main developments (or STEP parts) can be grouped into the following categories:

*Descriptive Methods*: These form the underpinning of the STEP standard, and include several Parts (e.g.1-19) containing basic concepts and formal principles, such as the definitions that are universal to STEP and descriptions of the EXPRESS data modelling language used in STEP (ISO, 1993b).

*Integrated Information Resources*: This group contains three different types of resources: generic, application resources and interpreted constructs, and could be considered to be the basic building blocks of STEP. The integrated resources is a set of data elements (or a collection of schemas) written in EXPRESS information modelling language and the related graphical representation EXPRESS - G, and were designed to be applicable to all
applications that deal with product data. As a result, the schemas are very generic and flexible, and also, independent of any particular domain.

**Application Protocols:** The application protocols (APs) are the data exchange standards for practical use, and could be seen as the actual products of STEP. The APs usually consist of complex data models describing specific data applications (e.g. building structural frame or steel works) (ISO, 1996).

Other categories include implementation methods to store and exchange product data by means of a neutral physical file format or via a Standard Data Interface (SDAI), which specify the representation and techniques used for implementing the exchange of product data described by EXPRESS. Conformance to the standard testing methods, which provide techniques for testing of software products conformance to STEP standards, and abstract test suites that contain the testing criteria data. However, it is important to recognise that the generic design of the integrated resources and APs are directly at odds with the first objective, because an unambiguous set of data elements would require not generic entities, but very specific entities which, consequently, would result in a very large number of entities. STEP solved this problem by introducing an innovative technique called 'Interpretation'. Interpretation is a generic integrated resource construct (Part 501599), like product, which can be understood within a particular usage domain; this is, are reusable resource entities that make it easier to express identical semantics in more than one application protocol.

### 3.3.3.2 EXPRESS and EXPRESS-G

EXPRESS is an information model specification language, which was initially developed in the 1980s for providing syntax for defining classes of entities, such as resources, material, equipment, specifications, etc., in order to enable the writing of formal information models describing products that support abstraction (Schenk & Wilson, 1994). It is also, one of the technologies that have been developed (version 1.0) as a Part of the STEP standard and is, since 1991, an ISO International Standard (ISO, 1993b). However, dynamic behaviour cannot be modelled with EXPRESS. Therefore, a modification of the language was developed in 1990 as a means of graphically representing EXPRESS models. Together, EXPRESS and EXPRESS-G can capture the functional information requirements and organisation perspectives of a process. The abstraction concept of
EXPRESS are enhanced by the concept of Subtypes and Supertype, both are used in the STEP development process.

As well as the EXPRESS language and the EXPRESS-G modelling notation, STEP part 21 also provides a physical file format, which identifies how an EXPRESS file should be developed to provide a means of information sharing (Amor & Wix, 1999). This is widely used as a formal file structure even where STEP APs do not exist. For instance, the CIMsteel standards and the initial Industry Foundation Class file format use part 21 structuring. STEP also provides a means of accessing database repositories via Part 22, the Standard Data Access Interface, which provides a series of standard calls onto databases structured using one or more EXPRESS language schema and allows use of multiple databases.

3.3.3.3 NIAM

Information analysis is the activity, which leads to complete, and formal specifications of the requirements of an information system within a company. A design method (graphical schema languages) to assist in the analysis of the required information in a formal and precise way was originally developed at the Control Data Holland Company by Nijssen (1974). The main purpose of the Nijssen's Information Analysis Method (NIAM), also known as Object Role Modelling is to create, through a conceptual structure abstracted from the real-world system; that is, what the information system is supposed to do, a usable computer based information system. This is achieved is by progressing through several subsequent phases, with a step by step increase in formality, the end point of the analysis being the Information Structure Diagram (ISD). This diagram is formal enough to be directly transferable to a database structure, which can be implemented in, for instance, an Object-Oriented or Oracle database to provide support to the working of the real-world system by structuring information flows (Halpin & Nijssen, 1989).

3.3.3.4 IDEFI and IDEFIx

Originally developed under the Integrated Computer Aided Manufacturing (ICAM) programme, IDEF1 was designed as a method for both analysis and communication in the establishment of requirements, and is used to produce an information model, which captures the information that exists about objects within the scope of an organisation
(Mayor, et al. 1992). Information models represent the structure and semantics of information needed to support the functions within the modelled system or subject area. Referred to as data modelling, IDEF1x is an extended version of IDEF1 that is based on a systematic approach for identifying the things for which data objects are kept (entities) and relationships between those things (attributes) that comprise the information flows defined in the enterprise information models.

IDEFIX is a method (or graphical schema languages) used for designing relational databases with a syntax designed to support the semantic constructs necessary in developing a conceptual database schema, depicting such information as key migration (Feldmann, 1998); that is, uses a modelling language (semantics and syntax similar to IDEF1), and associated rules and techniques, for developing a logical model of data. A conceptual schema is a single integrated definition of the enterprise data that is unbiased toward any single application and independent of its access and physical storage (KBSI, 2002).

### 3.3.3.5 International Alliance for Interoperability

A more recent initiative in the development of data exchange standards is led by the International Alliance for Interoperability (IAI), which is developing the Industry Foundation Classes (IFCs). The purpose of the IAI is to define Industry Foundation Classes (IFCs), which enable the development of information exchange not only by means of file based exchange and sharing of database repositories, as in STEP, but also by promoting the development of interoperable software applications which use the newer technology of client/server interfacing (IFC, 1999). This is achieved by developing product data models for sharing information between software tools, which are utilised throughout the building industry, through a common set of intelligent building design objects that will enable the sharing of information at all stages of the construction process (IAI, 1998).

Unlike the pan-industry approach taken in STEP, the IAI's initiative concentrates on building construction. Although the two initiatives are separate, STEP has been fundamental in making available technologies, which support Computer Integrated Construction (CIC) to the IAI, which has the advantage of active industry development (Wix, 1997).
3.3.3.6 CIMsteel Integration Standards (CIS)

The Eureka project (EU131) ‘Computer Integrated Manufacture of Constructional Steelwork’ first started in 1987 with the aim to improve the efficiency and effectiveness of the European Construction Steelwork industry both through harmonization of design codes and specification, the introduction of CIM techniques (structural analysis, connection design, data exchange and project modelling) for design, analysis, detailing, fabrication, erection and management functions (Watson, 1995). Within the context of the CIMsteel project, the Computer Aided Engineering (CAE) group led the innovative application of product model concepts drawn from ISO10303 (STEP) to the problem of data exchange and information sharing within the construction industry. The initial phase of the project (definition, design and specification) involved the incremental development, implementation in software, and trial industrial deployment of a Logical Product Model (LPM). A comprehensive information model, the LPM addresses the engineering data involved in the design and analysis and the manufacture and erection of the steel frame of a building. The most recent version of the LPM has taken on board STEP methodologies and STEP data exchange technology, and covers the complete design of constructional steelwork.

To achieve its stated aim, the project has developed, through the use of these advance technologies, a set of information data exchange standards called the CIMsteel Integrated Standards (CIS). The CIS identifies Data Exchange Protocols (DEPs), based upon the LPM, that correspond to the working practices of the construction industry. They provide standards against which the vendors of engineering applications software can develop and implement translators. These translators enable the users of such software to export engineering data from one application and import it into another. Formalised as the CIMsteel Integration Standards (CIS), this key deliverable from the CIMsteel project was initially published (CIS/1) by the CAE group in 1985. The CAE group subsequently substantially enhanced the specifications, with CIS/2 being published early in 2000.

During 1998, the EDI (Electronic Data Interchange) Review Team of the American Institute of Steel Construction (AISC) evaluated data transfer standards with a view to adopting one. On December 7th 1998, their recommendation of CIS/2 was approved by the AISC Board of Directors as part of the AISC Business Plan for Standardizing the Electronic Exchange of Structural Steel Project Information. Phase I of the AISC Business Plan includes the public endorsement of CIS/2 as being the standard for the electronic
exchange of structural steel project information for the entire U.S. structural steel design and construction industry, as well as a recommendation that it serve as the basis of an international standard to be developed under ISO (Wix, et al. 1999).

3.3.4 Object Model Based Standards

According to various studies (Svensson, 1998; Wix, et al. 1999) true object models are the next generation of development. They will underlie the object-oriented software that is used by industry and it is anticipated that they will also impact on the development of data exchange and data sharing standards (DETR Report, 1999). This section reviews various object model base standards relevant to building construction. However, before these modelling efforts are discussed, it is appropriate to mention the Object-Oriented Paradigm, an alternative approach to the more traditional methods used for modelling data.

3.3.4.1 Object-Oriented Paradigm

Objects accurately model the way real world entities behave. The concepts of objects and their interactions can be closely mapped onto many real world processes. In object-oriented models, all conceptual entities and their relationships are modelled as objects; an integer or string is a much an object as a complex assembly of parts (Banerjee, et al. 1987). This provides a simple but very powerful way of modelling and reasoning about the real world. This building of models is generally accepted in all engineering disciplines, largely because model building appeals to the principles of decomposition, abstraction and hierarchy (Eastman, 1984). Models provide the opportunity to experiment with new concepts, new components, and new processes under controlled conditions and with minimal risks. As far as possible, new models are based upon old models that inspire confidence (Booch, 1994). Object-oriented modelling provides an effective mechanism for doing this. The primary reason for choosing object-oriented approaches to modelling software (apart from the problems associated with traditional methodologies) is the endeavour to improve productivity and quality in the development process (Avison & Fitzgerald, 2001). According to Yourdon (1994) the main reason why this technology increases productivity and quality is the reuse of objects and prototyping facilities. The ability to store objects and reuse them in subsequent development, enables hope of reducing in the future both the cost and time of system development.
The prefix object-oriented (OO) is used to denote the programming paradigm based on the major elements of object, class, message, abstraction, encapsulation, inheritance, and polymorphism, which provides a uniform consistent view that is relevant to all aspects of system development, called object-oriented programming (OOP). Based on this, several object-oriented programming languages such as, Smalltalk, Eiffel, C++ and Java have been developed. One part of OOP is object-oriented analysis (OOA), which is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain (Booch, 1998). Therefore, the significance of OOA is to build objects that directly represent things in the real world. The basis of the OO paradigm is the concept that systems should be built out of objects, and that objects have data and functionality, whereby attributes define the data, and methods define the functionality.

An object can be considered as a self-contained component (e.g. thing, person, event, concept, report, etc), which is able to communicate and be in association with other such entities, which is applicable to the system (Ambler, 1998). Each object has its own attribute (data) and method (behaviour) encapsulated (hidden) within the object. The term method is used to describe the code fragment to perform a particular task or the behaviour that determines what it does and how it interacts with other objects. The functionality of each object is given by its collection of operations (called its object protocol). This protocol makes an impenetrable wall of code around the internal data. This basically means that from the outside of the object, its internal attributes can only be manipulated by its operations (methods), which are invoked via messages, sent to the object. A message is a request to an object to perform some service that normally invokes an operation. For example, a task object in a project scheduling application has methods for setting and retrieving values such as start and finish times. The methods are invoked by sending messages to the object. This process can be compared to applying a function to data in conventional programming languages.

In object-oriented analysis and design, software systems are modelled as a collection of objects, where objects are treated as instances of classes within a hierarchy of classes; that is, the characteristics of abstract or real objects are modelled using objects and classes. Encapsulation is the foundation of the object-oriented approach, which entails the process of compartmentalising the elements of an abstraction that constitutes its structure and behaviour; thus, serves to separate the contractual interface of an abstraction and its
implementation (Ambler, 1998). This basically means that the variables of an object cannot be accessed directly, but through the methods of that objects. Abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer. Hierarchy denotes the ranking or ordering of abstractions, whereby inheritance is the relationships between object classes, wherein one class shares the structure or behaviour of its parent class. Polymorphism enables objects of many different classes that are related by some common super-class to be denoted by a name; thus, any object denoted by this name is able to respond to some common set of operations in different ways.

The object-oriented paradigm for information systems and its technology is multi-faceted, were each of the facets has its own benefits. Examples of these are: object-oriented data modelling (OODM), object-oriented programming language (OOPL), object-oriented database management systems (OODBMS), and object-oriented graphical user interfaces (OOGUI). It should be emphasised that the definition and interpretations of the object-oriented paradigm, are still not standardised or fully agreed upon. Therefore, the description presented here, represents this researcher's (author's) own interpretation of the concepts.

3.3.4.2 CORBA

The Common Object Request Broker Architecture (CORBA) is the Object Management Group's answer to the need for interoperability among the rapidly proliferating number of hardware and software products available today. CORBA is intended to allow applications to communicate with one another no matter where they are located or who has designed them. CORBA 1.1 was introduced in 1991 and defined the Interface Definition Language (IDL) and the Application Programming Interfaces (API) that enable client/server object interaction within a specific implementation of an Object Request Broker (ORB). CORBA 2.0, adopted in December of 1994, defines true interoperability by specifying how ORBs from different vendors can interoperate (OMG, 1995).

The ORB is responsible for managing objects, messages passing between objects, and the interface between objects and external services. An object is an identifiable, encapsulated entity that provides one or more services requested by clients, the requests are events that have parameters and a target object, and may be generated by a client object when a
service or piece of information is required. Therefore, ORBs are the middleware that establishes the client-server relationships between objects. Using an ORB, a client can transparently invoke a method on a server object, which can be on the same machine or across a network. The ORB provides interoperability between applications on different machines in heterogeneous distributed environments and seamlessly interconnects multiple object systems. Additionally, ORBs provide flexibility. They let programmers choose the appropriate operating system, execution environment and even programming language to use for each component of a system under construction. More importantly, they also allow the integration of existing components. According to Wix, et al. (1999) CORBA is a signal step on the road to object-oriented standardisation and interoperability. With CORBA, users gain access to information transparently, without them having to know what software or hardware platform it resides on or where it is located on an enterprises' network.

3.3.4.3 The Unified Modelling Method

The Unified Modelling language (UML) is a relatively new object-oriented modelling approach for specifying, visualising, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems (Rumbaugh, et al. 1999). It represents a collection of the best engineering practices that have proven successful in the modelling of complex systems. UML defines a set of nine diagram types that provide relatively orthogonal views of the single underlying model; that is, the means for system modelling through the 'static structural view' (Class, Object, Component, Development diagrams) and the 'dynamic behaviour view' (Use Case, Sequence, Collaboration, State transition and Activity diagrams) of the system under analysis. The relationships between the various models are reflections of the iterative nature of object-oriented modelling.

3.3.5 Document Model Based Standards

A document is considered generally to be a representation of information in some commonly understandable form that is presented using an acceptable standard, such as paper or electronic methods (Wix, et al.1999). This section briefly discusses standards (or languages) for electronically representing information.
3.3.5.1 Extensible Mark-Up Language (XML)

The World Wide Web Consortium (W3C) have taken an interest in the development of markup language approaches, so as to harness the expressive power of the Standard Generalised Markup Language (SGML) whilst taken away some of the unnecessary complexity document model standards. However, prior to 1997, the exchange of data and documents was limited to proprietary or loosely defined document formats. In spite of this, the advent of Hypertext Markup Language (HTML) the presentation language for displaying interactive data in a Web browser, offered the enterprise a standard format for exchange with a focus on interactive visual content (Kimmance, 2000b). The process of making SGML simpler and Internet-aware gave rise to the Extensive Markup language (XML) and companion specifications such as the Resource Description Framework (RDF) and the Document Object Model (DOM). The technology of XML is formulated upon a similar structure to that of STEP; therefore, a brief introduction to some of the fundamental characteristic of XML as a data exchange recommendation (standard) would be instructive.

The Extensible Markup Language is an abbreviated version or subset of SGML and HTML. It is extensive, as it is not a fixed single format like HTML, and is a metalanguage that enables users to design their own markup language. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML (Bray, et al. 1998). XML is, essentially, a platform-independent way to structure data objects called XML documents that are stored on computers, and describe the behaviour of programs that process these objects. Its syntax has rapidly imposed itself as a popular format for structured document information interchange on the Internet, and its specification describes a sub-language for writing Document Type Definitions (DTD's). XML documents are made up of elements, which may have a set of attributes, in the form of key-value pairs, and may contain other elements, text, or a mixture thereof. An element may refer to other specific elements via special attributes, thereby allowing arbitrary 'graphic' structures to be represented. The structure of an XML document need not follow any rules beyond those laid out in the XML specification (Bray, et al. 1998).

Walsh, (1999) suggests that to exchange documents in a meaningful way, requires their structure to be described so that the various parties involved will interpret them correctly. This can be accomplished through the use of schemas. A schema contains a set of rules
that constrains the structure and content of a document’s components, (i.e., its elements, attributes, text, etc.). A schema also describes the intended conceptual meaning of a document’s components. In other words, a specification of the syntax and semantics of a potentially infinite set of XML documents. A document is said to be valid with respect to a schema if, and only if, it satisfies the constraints described in the schema. A more detailed discussion on XML and some of the other approaches have been reported in (Wix, et al. 1999; Kimmance, 2000b).

3.3.5.2 Alternative Methodologies

Besides XML and STEP, the ISO TC 184/SC 4 is addressing interrelated work issues, to include the ISO 13584 P-LIB standards for product model data and Part - LIBraries representation (ISO, 1994), the 15531 MANDATE (MANufacturing DATa Exchange), and the message standards EDIFACT (ISO 9735), controlled by the Working Groups (WG) 2 – 8 respectively (Cutting-Decelle & Michel, 2000). Other methodologies for representing product data include:

**PSL/PSA**: META Systems commercially developed problem Statement Language and Analyser (PSL/PSA). The PSL component is a language that can be used to describe information systems in terms of objects, properties, and relationships. PSL/PSA is based on the concepts of relational database theory. Formal and graphical representations are provided and reports can be generated from the commercially available software [Kusiak, 1999].

**IEM**: Integrated Enterprise Modelling (IEM) is a public domain methodology developed by IPK Berlin (European Committee for Standardisation, 1994). Unlike the majority of the other methods, IEM is designed around the object-oriented paradigm. Objects are categorised as products, orders, and resources. A generic activity model is defined for operating on objects. The object-oriented paradigm allows for the simultaneous modelling of the functional and information perspectives through a single construct class. This methodology demonstrates the robust and generic modelling capabilities provided by the object-oriented paradigm that are considered essential in product and process modelling (Fox, et al. 1996).
**CIM-OSA:** Computer Integration Manufacturing (and) Open System Architecture (CIM-OSA) is under development by the ESPRIT Consortium AMICE (European Committee for Standardisation, 1994). This method facilitates total enterprise modelling through a model construction process that includes enterprise requirement definitions, design specifications, and an enterprise implementation description. Four enterprise perspectives are considered using this method: function, information, resources, and organisation. Within each view, generic building blocks describe the functions, information, and resources in the system. Relations between building blocks define the total enterprise (Beekman, 1989).

**GRAPES/GRADE:** The Graphical System Modelling Language (GRAPES) was designed to facilitate the development of a Graphical Reengineering Analysis Design Environment (GRADE) software tool for analysing and modelling complex domains/systems, such as business enterprises, and optimising the processes and software systems comprising them (Tenteris & Vilums, 1996). Based on graphical layout algorithms, GRADE offers a very comprehensive system-modelling environment based on a formalised Graphical Modelling Language (GRAPES). Underlying this approach is an object-oriented perception of reality and the concept of representing systems in terms of what is considered as fundamental building blocks. GRADE was designed for analysing complex domains, such as business enterprises, and optimising the processes and software systems comprising them. A more detailed discussion of GRADE has been reported in (Kimmance, 1999; Kimmance 2000b).

### 3.3.6 Process Models of the Construction Process

The literature on process modelling is also comprehensive (Chung, 1989; Sanvido, et al. 1990; Koskela, 1995; Kartam, et al. 1997; Anumba, et al. 1998) with many research projects (Svensson, 1998; Kamara, 1999; Karhu, 2001; Kimmance, et al. 2002a). A process can be defined as a structured, logically measured set of consecutive steps or activities designed to produce a specified output, such as an end product or service being delivered; whereas, the *process model* represents the all-important steps throughout a project's life cycle. Process modelling can best be described as an abstract description of an actual or proposed process that represents selected process elements; that is, a representation of a set of consecutive steps or activities (tasks and events) with an end product or service being delivered (Kartam, et al. 1997). A variety of process models have need developed in recent years. Amongst these are:
Perhaps the most influential model, based on previous research is the Integrated Building Process Model (IBPM) developed by Sanvido, et al. (1990) for supporting the provision of a facility. The aim was to support computer-integrated construction and to define the critical success factors for construction projects. The model itself was represented in IDEFØ notation, and divided into five main hierarchical activities: manage facility, plan facility, design facility, construct facility, and operate facility.

Messner (1994) focused his effort on the IBPM top levels, by developing a methodology, process model and information architecture for assisting organisation in deciding which project they should pursue. The model when decomposed yielded an activity called 'provide facility,' the top level of the IBPM, and when further decomposed included activities such as process, product (or facility), environment, commitment and organisation information.

Another model defined using the IDEFØ method, was developed by Zhong, et al. (1994). The model covers the overall building process (although, it is more aggregated then the IBPM developed by Sanvido) by dividing a building construction project into four main phases: the initial phase, the design phase, the tendering phase, and the construction phase.

Karhu & Lahdenperä (1999) have described the current Finnish design and construction practice as six different process models corresponding to building procurement, architectural design, structural design, building services design, geotechnical design and constructing. The models were also created using IDEFØ.

Karhu (1997) also specifies a process for the design of a prefabricated concrete façade from an architectural point of view. A product model of the façade was also developed, and integrated with the process activity model using checklists.

Another model, defined used the IDEFØ method, was developed by Svensson (1998). The purpose of the high-level model is to achieve a basic structure on which a particular system, together with its applications, could be built as a layered architecture, from a facility manager's point of few.

A number of other process models have been developed in the domain of construction, and claim to cover the whole construction life cycle. Amongst these include:

- STAR research program (Hannus, et al. 1997a);
- MOPO project (Karstila & Björk, 1999);
Other models mainly focus on the design stage such as the ADePT model represented by a modified version of the IDEF0 notation called IDEF0v and described by Austin, et al. (1999). The process model illustrates the detailed design stage, and represents the process in terms of activities undertaken and their information requirements (i.e. constraints). Yet others focus on specialised areas, for example:

- Fisher & Li Yin (1992) introduced the General Data-Flow Model based on data flow from a construction contractor's viewpoint, focusing on contracts and procurement methods.
- The ATLAS model (Bakkeren, 1995) which describes the architectural and structural design processes from specific actor-based points of view.

Some process models introduce concurrent engineering features, such as the model presented by Anumba & Evbuomwan (1996), and the Client Requirements Processing Model (CRPM) for concurrent life cycle design in construction developed by Kamara, (1999) as part of his recent Ph.D. thesis. The CRPM is represented by the IDEF0 notation and describes the activities needed to be carried out, as well as the tools and techniques required for effectively defining, analysing and translating explicit and implicit client requirements into solution-neutral design specifications. A detailed discussion on some of the above process models can be found in (Kimmance, 2000c; Kimmance, et al. 2000).

3.3.7 Process Modelling Techniques

A number of process modelling methodologies exist to enable a system or process to be modelled. These methods include:

- Integrated Definition Ø Methodology (IDEFO)
- Data Flow Diagrams (DFD)
- Role Activity Diagrams (RAD)
- Petri Nets
- Scheduling
- Unified Modelling Language

These methodologies mostly have their origins in software systems development. This is probably because the current interest in construction process modelling is related to the
need for improved business processes and the need to implement computer-integrated strategies, all of which have their origins from the manufacturing and software engineering industries. Thus, suggesting that they may be suitable for modelling the on-site construction process. The most suitable modelling techniques are discussed in the following sections.

3.3.7.1 IDEFØ

IDEFØ is a subset of the Structured Analysis and Design Techniques (SADT). It is a top-down functional (activity) modelling technique for which efficient software is available, and is composed of three types of components: graphic diagrams, text, and a glossary, which are all, cross-referenced to each other (IDEFØ, 1993). The graphic diagrams are the major component of an IDEFØ model, defining functions and functional relationships via box and arrow interconnections, syntax and semantics, and presented in an organised and systematic way to gain understanding, support analysis, provide logic for potential changes, specify requirements, or support systems level design and integration activities. Thus, providing the means for modelling the decisions, information functions and dynamics (operations, actions, activities, processes) needed by a system, and the functional relationships and data (objects) that support the integration of those functions. Each model is composed of a hierarchical series of diagrams that display increasing levels of detail describing functions and their interfaces within the context of a system. IDEFØ is increasingly being used in both the research community and the AEC domain, because of its flexibility and clarity for modelling activities and the information flows between them. Figure 3.2 shows the notation of the IDEFØ technique.

![Figure 3.2: IDEFØ Notation](image)
3.3.7.2 Data Flow Diagrams

Another structural analysis and design technique is the data flow diagram (DFD) approach (DeMarco, 1979), a graphical modelling technique, which functions in a similar manner to the IDEF0 method, in that it is a top-down hierarchical methodology that represents activities that receive and produce information. A number of methodologies exist (Ward & Mellor, 1996; Gane & Sarson, 1979; DeMarco, 1979), which have been used in software development in particular, but have also been used in modelling the construction processes. These include:

- A system to analyse the communication between participants, proposed by Abou-Zeid & Russell (1993);
- A Management Information Model (MIM) that captures the flow of information on building projects between the contractor and various other parties (i.e. a comparison of contractor's construction information management systems), proposed by Fisher & Li Yin (1992); and
- A system to model the building design process, proposed by Austin, et al. (1996).

However, all follow the same fundamental principles and contain the same elements. A data flow diagram can be viewed as a system's analysis tool, which is used to show how the data moves around a system and is transformed as it does so. DFDs show the source of data, their structure, how they are distributed and their final destination. The model describes a system as a network of processes (functions), data stores, and an external entity used as sources and sinks, which are interconnected by informational flows of data on various levels of detail. For example, at level 0, the DFD represents the entire system, whereas additional information can be incorporated on level 1, level 2, and so on as sub-functions of the overall system. An interpretation of the basic constructs of a DFD is depicted in Figure 3.3. A process can has one or more inputs and one or more outputs.
3.3.7.3 Role Activity Diagrams

Role activity diagrams (RAD) use a formal iterative semantics or notation that was originally developed for software process modelling, although they are widely used in other domains, such as construction, finance and retail (Ould & Roberts, 1986). The central concept of RAD is that of a role (sometimes referred to as a type), whereby the role describes a sequence of steps or activities, which can be acted out by a business or system, which interacts with people (Abeysinghe & Phalp 1997). Roles are acted out in parallel (independent) and communicated through interactions. A role has a thread of activities represented by square boxes, which are organised from top to bottom, activities being connected by state-lines (the state between them). There are two types of activities within a role, actions and interactions. An action (represented by a shaded square) is a process step, which an actor of the role carries out in isolation. An action also changes the state of the role in which it occurs. An interaction between to roles implies shared or joint behaviour, represented by joining activities (lift clear) within different roles by a horizontal line. Interaction mat change the state of any of the roles that are involved in the interaction. Iteration occurs where a state is revisited, shown by a loop back to a previous point on the role. RAD notation bears a resemblance to Finite Stare Machine (Harel, 1988) and Petri Net approaches (Proth & Xie, 1996).
A basic example is shown in Figure 3.4, illustrating the main points of the RAD notation. This example shows three roles that make up a process. Each role is involved in a number of activities and interactions with other roles. Vertical lines connecting activities and interactions depict the ‘state’ of a role; horizontal lines between roles indicate where interaction takes place.

**Figure 3.4: A basic RAD example of a construction house project**

### 3.3.7.4 Petri Nets

Petri Nets (PNs) are a graphical modelling technique devised in the early 1960s to model manufacturing processes. Recently, it has been proposed as a modelling standard (Anon, 1997), and can be used, among other methods, for modelling business and software, and for descriptions of existing business and software systems prior to re-engineering. In its basic form, a Petri Net defines transitions that are interpreted as activities, and tokens that are interpreted as resources. The idea here is that resources (tokens) are required to undertake construction activities (transitions), and then moved from one place to another.
The concept of time can also be associated with a transition, whereby places represent states of readiness, tokens depict resources such as, actors, equipment or construction information, and arcs (arrows) indicate the direction of tokens when a transition takes place; that is, the flow of research through the process. Moreover, pre-and post-conditions can be added to a Petri Net and additional attributes (colours) can be added to tokens (Viswanadham & Narahari, 1992). Some authors have described the application of Petri Nets and useful PN constructs in modelling construction processes, i.e. applicable entities for construction, may be identified as sequential execution, concurrency, conflict, merging, synchronisation, and confusion (Wakefield & Sears, 1997). In addition, Petri Nets can be used effectively for simulation of construction processes, and are also used in a modified version by Augenbroe & Amor, (1997) to describe project windows that represent parts of the total construction process. However, the decomposition or hierarchy cannot be modelled using the basic Petri nets approach. Petri nets are aimed more at simulation, and thus are beyond the scope of this thesis.

3.3.7.5 Scheduling

The term scheduling is used here to denote the familiar general project planning or networking techniques (e.g. critical path, resource levelling or smoothing, precedence method, PERT, etc.), which, along with software tools, have been under development since the 1950s (Fondahl, 1980; Neale & Neale, 1989). Traditional scheduling methods represent the construction process (operations) through the use of a plan or Schedule Gantt Chart, as illustrated in Figure 3.5 (sometimes referred to as project plan), which is broken down into operational units (typically called activities or tasks), and the resources used for performing them. Activities represent the process that must occur over a certain time interval, identified by descriptive labels (e.g., construct substructure, concrete footings, build wall, etc.), duration, and task attributes such as start date and end dates, enabling the duration of the activity to be calculated. Activities can include relationships (shown by arrows) to the resources (items necessary for the task to be performed) they consume; plans can include milestones, which are events that occur at an instant in time rather than over an interval, and which typically represent the achievement of a particular state (e.g. structure enclosed). Gantt charts are not only used to schedule activities and indicate a programme of work, they are also a useful way of showing the resources allocated to a project,
monitoring the progress made in a project and enabling the critical path through the project and the float in any task to be calculated.

Although, the representation of information in scheduling methods can be limited, as it is captured by simple textual description, they do however draw upon a broad range of project information (i.e., product, process, and resource information), and also offer the scope of providing time dependency process flows. For example, activities, sequencing, duration, cost, the assignment of resources to processes, etc., which are some of the limitations that SADT and IDEFO models possess.

![Gantt Chart Example](image)

**Figure 3.5: An example of a Gantt chart**

### 3.3.8 Project Modelling (Integrated Product and Process Models)

There are different views within the construction industry and the research community on what constitutes a project model (Gann, et al. 1996; Stumpf, et al. 1996; Fisher, et al. 1997; Anumba, et al. 2000; Kimmance, et al. 2002b). Some see it simply as an amorphous collection of all the information relating to a project, irrespective of the medium of storage (people's heads, paper drawings and specifications, document and CAD files, etc.) or the method of dissemination of the project information. Others see it in terms of a single database which holds all the information on a project and which is accessible to all members of the project team (Anumba & Amor, 1999). Yet others view a project model as an integration of product models (information relating to the building product) and process models (information regarding the construction and business processes required to translate the product information into a physical product - the constructed facility).

A **project model** provides a framework for system integration of product, process, and organisational aspects, allowing the capture and use of richer time semantics, for
construction management projects. This entails the integration of the building components with the process facilities available for producing the finished Artefact. Among the project models for the A/E/C industry include:

- The Information Reference Model of AEC (IRMA) by Luiten, et al. (1993); a generic conceptual model of construction information that focuses on identifying the relationships between information concerning real-life objects in a construction project and the real-life objects themselves. Its main aim is to clarify terminology, since similar concepts are often used in a slightly different manner. The model defines central project object types to represent key concepts, such as activities, resources, product, contract, participants, states and physical components) and their relations, in the form of EXPRESS files and EXPRESS-G diagrams. Thus, the model can be described using a modelling language such as EXPRESS, NIAM and IDEF1x. While the model was intended more as a reference model than an end product, it has served as useful vehicle for further conceptual development. Figure 3.6 shows the basic concepts that represent the context of the IRMA model; it also illustrates the relationships within a conceptual model in the form of a modified improved AEC process view model (Froese, 1994).

![Diagram of IRMA model](image)

**Figure 3.6: Basic concepts representing the IRMA model with three levels of abstraction and the improved AEC process view model (Source: Adapted from Luiten, et al. 1993)**

IRMA is a combination of four conceptual information models: the Building Project Model (BPM) by Luiten, (1994); the Unified Approach Model (UAM) by Björk, (1992); the General Construction Object Model (GenCOM) by Froese, (1992), and the Information-Integration for Construction Model (ICON) developed by Aouad, et al. (1994).
It is appropriate to mention that those models, which comprise IRMA, are recognised as conceptual models of construction information. A difference between these information models in comparison to modelling methods is that they merely define information entities in the construction process. A number of other conceptual models of construction information have been proposed (Hannus & Pietiläinen, 1995; Jägbeck 1998; Dado & Tolman, 1999).

However, the concept of a construction project model may be difficult to define precisely, and the above model definitions focus too much on the data representation aspects and thus, are neither wholly accurate nor comprehensive. Greater insight into what constitutes an integrated product and process model can be gleaned from its architecture by means of reviewing the integrated modelling environment.

3.4 SELECTION OF MODELLING TECHNIQUES

Many modelling methodologies have been examined to identify suitable techniques for representing product and process models of the on-site construction process. The modelling techniques that are most appropriate to undertake this modelling have been described in Section 3.4. Each of these techniques has advantages in modelling certain types of activity or product data. This section evaluates modelling techniques that are considered suitable for integrating product and process information.

3.4.1 Comparison of Product Modelling Techniques

The modelling methodologies (or languages) that are considered most appropriate to undertake the modelling of product data have been described in the previous section, only some of which are used in the construction industry. Amongst these are: IDEF1x, NIAM, EXPRESS, EXPRESS-G, XML, GRAPE5/GRAPES, and UML.

Some elements of a (rough) comparison between these methods are presented in Table 3.1, by means of a cross representation of their features and/or properties. A synthesis of these various approaches for modelling product data is based on a set of criteria. The criteria were intended to provide a common platform for comparison and, while every effort has been made to obtain the relevant information related to each technique, there are still gaps in the information available. It should be emphasised that the findings are largely based on
extensive literature reviews and the information provided by several researchers themselves, it was not possible to undertake a hands on evaluation of all the methodologies to establish the veracity of the claims made by the developers.

Table 3.1: Matrix of Features of Product Modelling Methods

<table>
<thead>
<tr>
<th>Criteria</th>
<th>XML</th>
<th>IDEF1x</th>
<th>NIAM</th>
<th>EXPRESS</th>
<th>GRADE/GRAPE</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modelling Approach</td>
<td>Relational (object extension)</td>
<td>Relational (object extension)</td>
<td>Object-oriented (OO)</td>
<td>Object-oriented (OO)</td>
<td>Object oriented (OO)</td>
<td></td>
</tr>
<tr>
<td>Software Availability</td>
<td>Good</td>
<td>Poor</td>
<td>No</td>
<td>Good</td>
<td>Very good</td>
<td>Very good</td>
</tr>
<tr>
<td>Standardisation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Average</td>
<td>Fair</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Understandability</td>
<td>Yes/No</td>
<td>Fair</td>
<td>Good</td>
<td>Fair</td>
<td>Good</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Dynamics Aspects</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Partly: V1: No V2: Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Constraint Expressions</td>
<td>Average</td>
<td>No</td>
<td>Yes</td>
<td>Poor</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Process Representation</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Partly: V1: No V2: Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Entity Representation</td>
<td>Fair/Rules</td>
<td>Poor</td>
<td>No</td>
<td>Rules</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Applicability</td>
<td>Program and Internet Analysis</td>
<td>Management Analysis</td>
<td>Analysis for data Management</td>
<td>Data Analysis</td>
<td>OO Analysis &amp; Reengineering</td>
<td>OO Analysis &amp; Programming</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>Yes/V. Good</td>
<td>Yes/Fair</td>
<td>Yes/Poor</td>
<td>Yes/Fair</td>
<td>Yes/V. Good</td>
<td>Yes/V. Good</td>
</tr>
<tr>
<td>Computer-Business Construction</td>
<td>Yes/Poor</td>
<td>Yes/Fair</td>
<td>No</td>
<td>Yes/Poor</td>
<td>V. Limited</td>
<td>V. Limited</td>
</tr>
<tr>
<td>Full Abstract</td>
<td>Yes: including operating constraints</td>
<td>Missing Operators</td>
<td>Missing Operators</td>
<td>Yes: including operating constraints</td>
<td>Yes: including partly operating constraints</td>
<td>Yes: including operating constraints</td>
</tr>
<tr>
<td>Data Types (needed for object semantics)</td>
<td>Partial Support</td>
<td>Missing: Not Supported Assumes total integrity</td>
<td>Not Supported Assumes total integrity</td>
<td>Partial Support Assumes total integrity</td>
<td>Partial Support</td>
<td>Full Support</td>
</tr>
<tr>
<td>Integrity Management</td>
<td>Partial Support</td>
<td>Not Supported Assumes total integrity</td>
<td>Partial Support Assumes total integrity</td>
<td>Partial Support</td>
<td>Full Support</td>
<td></td>
</tr>
<tr>
<td>Capabilities (needed for iterative design)</td>
<td>Partial Support</td>
<td>Not Supported Assumes total integrity</td>
<td>Partial Support Assumes total integrity</td>
<td>Partial Support</td>
<td>Full Support</td>
<td></td>
</tr>
</tbody>
</table>

3.4.2 Comparison of Process Modelling Techniques

The previous section has also described several tools and techniques that can be used for process modelling, among which are: IDEF0, DFD, RAD, Petri Net, Scheduling, and UML. Each method varies in complexity and functionality often-utilising very different formalisms, notations and graphical representation. The ease in which the various modelling constructs can be understood by end-users varies considerably from one approach to another and is an important factor in choosing between the existing techniques. However, the overriding factors remain the primary objective of the modelling initiative. In many cases, this sets the requirements for the usability of the resulting models. Some elements of a (rough) comparison between a selection of process modelling tools and techniques, which were considered for developing a construction process activity model is presented in Table 3.2, by means of a cross representation of their basic features. These
Chapter 3

Techniques are currently being applied mostly in the manufacturing and computer software industries. However, some are beginning to be adapted for the construction industry.

Table 3.2: Comparison of Process Modelling Methods According to Design Criteria

<table>
<thead>
<tr>
<th>CRITERIA</th>
<th>GRADE</th>
<th>UML</th>
<th>IDEFØ</th>
<th>DFD</th>
<th>RAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modelling Approach</td>
<td>Object-Oriented</td>
<td>Object-Oriented</td>
<td>Static Activities</td>
<td>Data Flow Diagrams</td>
<td>Sequence of Activity Diagrams</td>
</tr>
<tr>
<td>Software Availability</td>
<td>Yes: Little</td>
<td>Yes (Good)</td>
<td>Yes (Good)</td>
<td>Yes (Good)</td>
<td>Yes (Fair)</td>
</tr>
<tr>
<td>Standardisation</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Yes</td>
<td>Yes/No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Understandability</td>
<td>Yes/No</td>
<td>Yes/No</td>
<td>Yes</td>
<td>Yes</td>
<td>Fair</td>
</tr>
<tr>
<td>Dynamics Aspects</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Fair</td>
<td>Yes</td>
<td>Fair</td>
<td>Fair</td>
<td>Fair</td>
</tr>
<tr>
<td>Link to Data Model</td>
<td>Yes: Fair</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Layering</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Actor Prerogatives</td>
<td>Yes: Limited</td>
<td>Yes</td>
<td>Limited (mechanisms)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Used in Construction</td>
<td>No (manufacturing)</td>
<td>Very Little</td>
<td>Moderate</td>
<td>Little</td>
<td>Very Little</td>
</tr>
</tbody>
</table>

The review of modelling techniques has determined that IDEFØ and UML diagrams are the most suitable systems to integrate product and process information, with regard to the on-site construction process. The main reasons for this is that the IDEFØ technique is relatively easy to use and understand, and is extensively used for integrating many change philosophies (e.g. CE, BPR, QFD, etc), which is very important if the model is to be changed at the start of a building project and maintained throughout it. IDEFØ is also is extensively used for functional/activity modelling for which efficient software is available, and is appropriate since one of the requirements was to facilitate the required dynamic process activities for describing the on-site construction information of a facility. The main reasons for using UML is that it is a fairly new object-oriented modelling approach for integrating product and process models, for which efficient software is also available, and is rapidly being established as the de facto standard for object-oriented modelling because it is relatively comprehensive. UML notation also offers a consistent language to support the on-site construction process represented by the IDEFØ diagrams, and appears to offer the most appropriate notation for use in achieving the modelling objectives. In addition, IDEFØ and UML share similarities with each other, in the sense that they both aim at providing a basis for the understanding, definition and capture of the requirements needed to develop the integrated product and process model.
3.4.3 A Detailed Description of IDEFØ Modelling

The potential power of the Structured Analysis and Design Technique (SADT) as a communication and analysis-modelling tool was recognised by the United States Air Force and selected as the language to support the ICAM program. This resulted in the development of the ICAM DEFinition Methodology (IDEF), later to be renamed Integrated DEFinition Ø Methodology (IDEFO) (U.S. Air Force, 1981). SADT models evolve through a process of ‘top-down’ structured decomposition, whereby, a single box diagram may be broken down to form another diagram containing three to six boxes, each box can in turn be further broken down to contain three to six boxes, and so on. IDEFO does not however, share the same features of the other IDEF methodologies, such as IDEF1x or IDEF3, but its fields of application are in functional modelling.

The basis of an IDEFO model is that an input is consumed and transformed by the activity (function box) into an output. An activity is triggered when the state of the input and the control allow it, while the mechanism identifies the primary means by which the activity is executed. Therefore, a control governs when an activity occurs (and is not necessary consumed), while a mechanism is something that is used (but not consumed) during an activity. Thus, in an IDEFO diagram the boxes represent activities and the connecting arrows represent things (e.g. raw materials, products, documents, data, etc.) it is the location of an arrow relative to the box it connects which determines its role (input, control, mechanism or output), as illustrated in Figure 3.7. It is also of importance to realise that the connecting arrows can split or merge, and that a control may also act as an input (i.e. be consumed). Furthermore, as IDEFO is recognized as a ‘top-down’ modelling approach (Marca & McGowan, 1988), it is important to recognise that the IDEFO diagrams should be read top-down from the first ‘A-O’ context node diagram'. The reference node always contains a single box with external arrows defining the high-level linkage of the total model to the outsize world. This ‘scope definition’ activity is decomposed at the second level on the A0 node, which will show a number of interconnected sub-activities. These sub-activities are in turn decomposed at the third level on the A1, A2, A3, A8...nodes. Consequently, the total construction process model will contain a hierarchy of nodes within which function boxes representing ‘activities’ are progressively decomposed and coherently linked by arrows (or relationships) representing the flow of ‘things’. Figure 3.8 illustrates an example of a top-level A0 construction.
process activity diagram. Each activity or process can be partitioned to show finer detail on another diagram, ensuring a single diagram does not become too cumbersome.

**Control Arrow**
Express the conditions required to produce output, for example laws, standards and guidelines, product characteristics to produce output (Trigger to the activity)

**Environment Constraints**
Used to provide information at a specific level of decomposition that is not required for further understanding at some other level

**Input Arrow**
Represents material data that is transformed by the function activity

**Representation of a box is a:**
Function (Activity)
(A function is identified by a verb or verb phrase that describes what must be accomplished)
An activity or block has a number:

**Output Arrow**
Represents product or object data into which inputs are transformed

**Mechanism Arrow**
The class of arrows that express the means used to perform a function for example, machinery, tools or equipment

**Tunnelling Arrow**
Used to provide information at a specific level of decomposition that is not required for further understanding at some other level

**Building Team & Equipment**

**Boundary Arrow**
An arrow with one end not connected to any box on a diagram (see glossary)

**Internal Arrow**
An input, control or output arrow connected at both ends to a box on a diagram

**Other Types of Arrows**

**Figure 3.7: Basic concept of the IDEF0 method illustrating the top-level process model**

**Figure 3.8: An example of the top-level [A0] construction activity diagram**
3.4.4 The Unified Modelling Language

The unified modelling language (UML) has played a major role in this thesis, since it has been selected to develop a generic building product model (described in Chapter 4) of the construction resources and product elements (identified and represented in the IDEF0 diagrams), and the relationships between them. UML is a relatively new object-oriented modelling approach for integrating product and process models. For this reason, it is important to provide a basic understanding to some of its features and to clarify a suitable definition of UML. The following are some definitions or interpretations of UML:

- A definition by Anumba, et al. (1999) states that UML is not a modelling method in itself, rather a modelling notation, or more, a standard graphical modelling language for modelling multiple perspectives of information systems.
- Rumbaugh, et al. (1999) define UML as a visual modelling language (based on an integrated approach of three object-oriented methods) used for specifying, constructing, and documenting the artefacts of a object-oriented software-intensive system, and is not intended to be a visual programming language, in the sense of having all the necessary visual and semantic support to replace programming languages. Although, it does draw the line as it moves closer toward programming code.
- A definition by Kobryn (1999) states that UML is a graphical notation to modelling static and dynamic characteristics in an object-oriented environment.
- Object Management Group (OMG, 2000) defines UML as an Industry-Standard language consisting of graphical symbols used to model software development processes. It is an amalgam of three popular modelling languages: Booch, Rumbaugh’s OMT, and Jacobson’s OOSE models developed by a large number of software companies headed by Rational Software. UML combines some of the best features of the object-oriented paradigm, database theory, and systems design into a programming language-independent modelling system that can been used to represent large-scale systems.

The above definition logically leads to the definition of the term 'integrated project modelling', which can be understood as a systematic means of modelling the products (e.g. resources, classes, objects, components, etc.) and processes (e.g. activities, functions, tasks, etc) and all the relevant relationships between them. It is of importance to recognise some
fundamental features in order to establish the underlying reality of the unified modelling language. These are as follows:

- **Visualising** through textual and graphical symbols using a well-defined semantic programming language;
- **Specifying** a precise, unambiguous, and complete model. UML addresses the specifications of all the important analysis, design, and implementation decisions, which need to be met when developing a software system;
- **Constructing** may take place by direct connection to a variety of programming languages. UML can map a model then export it to a program language such as JAVA, C++, or Visual Basic; and
- **Documenting** artefacts such as, architecture, requirements, design constraints, specifications and source codes, project plans, tests, prototypes, releases, etc.

As the creators or authors (Booch, et al. 1999) began the unification of these methods, they established four common goals to focus their efforts. These include:

- to model systems, not just software using object-oriented concepts;
- to establish an explicit coupling to conceptual as well as executable artefacts;
- to address the issues of scale inherent in complex, mission-critical systems; and
- to create a modelling language usable by both people and computers (machines).

The principal structure of UML is built on three fundamental elements:

- the basic visual building blocks;
- the rules that dictate how the building blocks are integrated together; and
- the mechanisms that glue the UML diagrams together.

According to Booch, et al. (1999), the individual basic elements of UML encompasses three kinds of building blocks. These building blocks provide a blueprint for developers so they know what they need to construct, and enable project managers to precisely estimate the cost of a given project. They also provide the bridge between technical developers and non-technical users; thus, allowing the developers to obtain a precise understanding of the user requirements for the system to be built. These include the *things* or abstractions that
are first-class citizens in a model, relationships that tie things together, and diagrams that group collections of things.

It is also important to realise that a diagram is not a model, but only a partial graphical notation of a set of elements within a single model. A diagram is a projection onto the model, as a kind of perspective on the model, and several diagrams are necessary to illustrate the entire model (Anumba, et al. 1999). Since notations are used to model business domains and applications with a great deal of inherent complexity, no single view or diagram type is adequate to capture all-important aspects of the resulting model. Therefore, UML defines a set of nine diagram types that provide relatively orthogonal views of the single underlying model; that is, the means for system modelling through the 'static structural view' (Class, Object, Component, Development diagrams) and the 'dynamic behaviour view' (Use Case, Sequence, Collaboration, State transition and Activity diagrams) of the system under analysis.

Figure 3.9 illustrates a design data model of the 'input into' relationship between diagrams in UML. The boxes within the package are intended to represent the key UML diagrams, from an iterative point of View, whereas the arrows symbolise the relationships between them. The arrowheads indicate the input into relationship (e.g. a use case diagram is input for a class diagram).

![Figure 3.9: Package diagram showing input into relationships of the UML diagrams](image_url)
Figure 3.9 also provides insight into one of the fundamentals of object-oriented modelling. The relationships between the various models are reflections of the iterative nature of object-oriented modelling. Processes to support the use of UML have been addressed in Rational Objectory Process (Quatrany, 1998), which has more recently been referred to as the Unified Software Development Process (Jacobson, et al. (1999), and Use Cases combined with Booch methodology, Rumbaugh’s OMT and UML (Texel & Williams, 1997). The diagrams, which comprise UML, will not be discussed in any detail in this section. However, Chapter 4 describes the UML diagrams used to develop a building product model. A full description of the UML diagrams has been reported in (Kimmance, 2000b).

3.5 COMPUTER INTEGRATED MODELLING ENVIRONMENT

3.5.1 Need for Integrated Project Models

There is growing realisation that while product and process models serve a useful purpose, there is much more benefit in integrating them. In this context, ‘integration’ means merging both product and process representations into a common framework, allowing the capture and use of the same time semantics and properties coming from both models, without loss of information. Within a concurrent engineering environment, there is need for all members of the project team to have a clear understanding of the processes (and sub-processes) and intermediate products involved in the production of an artefact. This enables team members to better appreciate their roles and relationships with other team members in producing the desired artefact. Thus, the integration of product and process models is seen as a keystone of concurrent engineering (Anumba, et al. 1999). This is even more so in construction, given the fragmentation that exists between the participants in a construction project, as well as between the stages in the construction process. One or several team(s) involving architects, structural and building service engineers, project/construction managers and construction specialists (professional trades) usually carry out building design and construction. Each of these participants is responsible for certain aspects of the whole construction project. One of the main issues in multidisciplinary collaboration of this nature is the exchange of information between project team members in order to achieve a coherent project. To date, this often means face-to-face meetings, and exchange of drawings and specification documents, amongst other things. This is a very time-consuming process and requires extensive coordination efforts (Sun &
Lockley, 1995). The lack of co-ordination between teams often leads to on-site disputes or arbitration by the client or, worse, by a leader who goes beyond his current role.

The majority of communications between participants involved in a construction project is with drawings and specifications, but with recent IT developments more communications are using electronic means (e.g. computer files on disks, electronic information exchange, E-commerce, electronic data interchange (EDI), e-mail, etc.). This increase in electronic communications is causing even more problems such as incompatibility and lack of communication standards (e.g., DXF for CAD drawings). It also increases the risk of altering information without prior notification producing data inconsistency and it is a hindrance in the transfer of knowledge between team members; that is, you can change a detail on a drawing without explaining why to the other parties involved (Kagioglou, et al. 1998). The use of an integrated environment to combine the information used by all parties would help rid the industry of some of the problems discussed above, and also bring other related benefits such as reducing construction lead times, reducing cost, claims and disputes, and improving quality; resulting, in satisfied clients (Kimmance, et al. 2002b).

The main purpose of integrated product and process modelling is to automate certain tasks during the overall construction process, and to provide support to designers and contractors in an integrated approach. For this purpose, a system for integrating construction information must be able to:

- support multiple tasks by including multiple tools in the system;
- provide effective support throughout the design and construction process;
- automate data exchange between the different tasks of the system;
- facilitate effective communication between the heterogeneous software tools in order to support integration through the whole building life-cycle; and,
- provide a certain degree of 'intelligent' support in a given context.

This integrated view of product and process models relies on an explicitly formalised knowledge of the data handled during the whole construction process. According to Anumba, et al. (1999), several forms of modelling are necessary to provide this information: static modelling, functional modelling, dynamic modelling and process modelling. Static modelling is the process of examining the static structure of a system and by identifying the structure and relationships of system objects and unifying their
description (representing) through the use of a common standard/language. Functional modelling is system-specific analysis, describing functions undertaken by the system, whereas dynamic modelling represents the sequence of actions in which the building system evolves as the result of upstream decisions. Finally, process modelling describes the design and construction process, as well as the pre-conditions for the decisions and their consequences.

3.5.2 Alternative Architectures for an Integrated Project Model

Although there is widespread agreement that an integrated product and process (project) model is highly desirable for computer-integrated construction, there is far less agreement on what form it should take. This was alluded to in the discussion of the term 'project model'. It is also reflected in the architecture that have been proposed or adapted so far in the development of the model. Anumba, et al. (2000) reviews some of these architectures with references, where appropriate to research prototypes. These include: Project Model as Reference Model, Centralised Project Database, Distributed Project Database, Neutral Format Project Database, and Proprietary Approaches. The project participants and the management structure adopted for the construction project can determine the application of these different architectures.

3.5.3 Related Integrated Projects

Background research into computer integrated construction (CIC) for A/E/C projects shows that integration in construction has been addressed in various ways. For example, the communication between applications, achieved through specific software integrating a unique and invariant set of chosen applications; integration through geometry (often the case in commercial CAD packages and limited to geometrical information); knowledge-based interfaces linking multiple applications and multiple databases (Howard, 1991), and integration through shared or central project databases (sometimes referred to as integrated product and process models) holding all the information relating to a project according to a common infrastructure model (Anumba & Amor, 1999). However, many previous studies have concentrated either on product modelling or on process modelling, with inadequate consideration of the other. Both types of models are considered complementary and cannot be readily delineated. Cutting-Decelle, et al. (1997) are of the view that the intersection
between product and process models has to be highly interactive, particularly within a concurrent engineering context, and should therefore be modelled in a dynamic way.

Some researchers are beginning to explore aspects of the integration of product and process models. For example, Aalami & Fischer (1998) present an approach based on how construction method knowledge captured in predefined, computer-interpretable construction method model templates supports the joint elaboration of product and process models. The approach is intended to allow construction planners to generate design-build scenarios rapidly and to visualise them readily as 4D models. Others researchers that have addressed the development of integrated project models for construction include:

- The OSCON project at Salford University (Aouad, 1997) was funded by the Department of Environment and addresses product and process modelling to some extent by integrating an object-oriented building model with, among other things, a scheduling application for architectural design, costing and construction planning.
- The COMBINE (Computer Models for the Building Industry in Europe) project was a major European Union (EU) research project led by Delft University (Dubois et al., 1995). Its main objective was the development of intelligent integrated building design systems utilising an integrated data model conforming to ISO STEP in order to provide an interface for multiple design performance tools covering building services (e.g. energy, heating & ventilation, space planning, etc.).
- The ICON (Information Integration for Construction) was based on a research project at Salford University, which involved both academic and industry practitioners and funded by SERC (Aouad, et al. 1994). The project focused on assessing the feasibility of establishing integrated databases and to produce information models for the construction industry. The scope was limited to the integration of design, procurement and construction activities.
- The COMMIT (Construction Modelling and Methodologies for Intelligent Information Integration) project at Salford University also builds on the ICON project (Brown, et al. 1996). However, its focus is on achieving integration through information management. A COMMIT Information Management Model (CIMM) has been developed to support collaborative working with facilities for versioning, notification, object rights and ownership. The model also facilitates the recording of the intent behind construction project decisions, thereby providing a complete project history.
Chapter 3

- The SPACE (Simultaneous Prototyping for an Integrated Construction Environment) project at Salford University (Alshawi, 1996) led to the development of the prototype Integrated Construction Environment (ICE). The ICE model facilitates the automatic generation of virtual reality (VR) models, specifications, construction plans, cost estimates, and site layout plans directly from CAD drawings. It transfers project information dynamically and at run time to and from individual construction application packages. Its application focuses on the detailed design and construction stages of the construction process.

- The WISPER project also at Salford University (Faraj, et al. 2000) proposes a three-tier client-server approach for demonstrating the integration between detailed design, cost estimating, scheduling, and a VRML interface for graphic querying of a database.

- The ATLAS project (Tolman, et al. 1994) aimed at the development, demonstration, evaluation and dissemination of architects, methodologies and tools for computer integrated large-scale engineering by utilising information flow and information exchange through data integration.

In addition to the research projects described above, several other research projects that had a significant influence on this research include the following:

- The MOPO (Models of Construction Process) project is a multi-national research and development project led jointly, among others, by the Royal Institute of Technology (KTH) in Sweden and the Technical Research Centre (VTT) in Finland (Karstila & Björk, 1999). The aims are to provide new modelling methods, conceptual (process & product) models, IT-based tools and computer technologies to support construction process improvement through systematic process design and analysis using a variety of construction modelling approaches. A number of useful efforts have been developed within the project, such as a formalised process model (ProFacil) focusing on the links and activities between facility management (Björk, et al. (1999), the GEPM (Generic Construction Process Modelling Method) based on a flexible conceptual model approach, which users object-oriented concepts for improving process management, and for enhancing communications in a construction process (Karhu, 2001).

- The STAR research program has been undertaken by the Technical Research Centre (VTT) in Finland, its aim is to develop systematic methods and IT tools to support construction process re-engineering (Hannus, et al. 1997a). The STAR program
consists of various projects aimed at developing techniques for improving and re-engineering the design process and integrated information management by reducing the overall construction process time. Amongst these are:

- A Construction Process Models (STARGEN), which defines the basic concepts of project, contract, and project objects and their relations to the process (liability, flow, value, resources etc), thus forming the formal language for reference models and the basis for software tools (Hannus & Pietiläinen, 1995).

- A Production Process Model developed within the COSMOS project illustrates systematic and detailed description of the implementation part of the building processes. The model consists of the tender stage, production planning, production management, physical work and acceptance, and is represented using the IDEF0 modelling technique (Karhu, et al. 1997).

- A Design for Construction Framework developed within the CRUX project, aimed at providing a generic conceptual model that integrates production management (site processes). The model consists of a core model and four related sub-models.

- The PROMO model is a prototype software tool for construction process modelling, analysis and re-engineering (Hannus, et al. 1997b).

- The Generic Design and Construction Process Protocol (GDCPP) was developed as a result of a research project undertaken at the Salford University (Kagioglou, et al. 1998). The goal was to provide a conceptual process model, which implements, integrates and re-engineers the different functions within the construction process. The Process Protocol is a common set of definitions, documentation, procedures and principles that provide the basics to allow a wide range of organisations involved in a construction project to work together seamlessly. GDCPP is intended to provide a common framework (generic model) for meeting the 'client’s' business needs and expectations, and to improve the process for managing and controlling the whole life cycle of a construction project. An extension to the original project carried out at Loughborough University provided further support to developing the initial Activity Zones of the GDCPP in more detail by defining and modelling the high level processes into sub-processes at a lower level (i.e. level II). The aim of Process Protocol (PP) II is to provide a visual representation using a 'stage-gate approach' (Cooper, 1990) by illustrating what the sub-processes are, and how they interact.
Chapter 3

• A Generic Process Model and Building Product Model (KBS) were proposed by Svensson (1998), as part of his Ph.D. thesis on 'Integrating Facilities Management Information. The purpose of the models is to achieve a basic information structure to support the main processes of Facilities Management (FM). Activity models and object-oriented conceptual schemas were also defined for the life cycle of ordinary buildings.

The above examples demonstrate the growing interest in integrated product and process models, and addressing the related information management issues. In order to provide an understanding of these integrated product and process approaches, some elements for a comparison are provided, by means of a cross representation of their main features and properties.

3.5.4 Evaluation of Existing Model Approaches

The survey and analysis covers the main approaches to the development of product, process and integrated project models for construction. The projects reviewed include: ATLAS, ADePT, MOPO, GDCPP, IRMA, COMMIT, COMBINE, MIM, STAR, CRPM, ICOM, SPACE, RATAS, OSCON, Class Works Model and IBPM.

3.5.4.1 Criteria for Analysis

The synthesis of the various model approaches is based on a set of criteria. The criteria were intended to provide a common platform for comparison and, while every effort has been made to obtain the relevant information related to each method, there are still gaps in the information available. The criteria used in the synthesis include model aim, application stage, IT development, modelling integration approach, availability, standards used, and current and future status. Other factors relating to the criteria stipulate whether the models were layered, integrated project databases, conceptual models, made up of multiple models/projects, combination of approaches, and/or had product and process relations. A comparison based on a set of criteria of the various models addressing construction is summarised in Table 3.3.

It should be emphasised that the findings are largely based on extensive literature reviews and the information provided by several researchers themselves, it was not possible to
undertake a *hands on* evaluation of all the models to establish the veracity of the claims made by the developers. Furthermore, where models claim to address several stages of the 'whole life cycle' of the construction process, it should be noted that in some cases, this is simply limited to the provision of information for that particular stage or actor and does not necessarily imply full support.

### 3.6. SOFTWARE TOOL SELECTION

The use of information models to manage development projects is analytically needed to communicate information through ever increasing project complexity. Software tools can simplify system analysis, track requirement specifications, support popular modelling techniques and generate code from design. Each product like a construction project has a unique blend of functional characteristics, such as system architecture, design methods, programming languages, building techniques, etc. However, selecting software is generally not an easy process. There are a wide variety of tools available, and some are more appropriate for specific applications than others. The tool selection can have a large impact on the type of conceptual information models, the interface design, the degree of effort and time needed to construct the prototype and the fidelity type (high or low) of the prototype system, the number of design iterations within the schedule, whether or not code from the prototype is used in the application, cost and ease in which the software tools can be applied.

According to Rudd & Isensee (1996), a basic approach that is often used in software (research) projects, and to a certain extent also used in this thesis, for selecting software application development tools, include the following:

- to identify the available tools and evaluation their capabilities;
- to establish project/system specific criteria;
- to prioritise the criteria, and if possible, map these criteria to the tools' capabilities.
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Project</th>
<th>ATLAS</th>
<th>ADePT</th>
<th>OSCON</th>
<th>IRMA</th>
<th>COMMIT</th>
<th>STAR</th>
<th>CRPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim of Project Model</td>
<td>To demonstrate the applications and benefits of advanced IT systems in large scale engineering projects</td>
<td>To develop a techniques that improves the interdisciplinary nature of construction processes</td>
<td>To demonstrate value of integrating project information in a central project databases</td>
<td>To identify central objects/concepts and their relationships in AEC projects</td>
<td>To achieve integration through information management (IM)</td>
<td>To develop systematic methodologies and information technologies for rapid construction</td>
<td>To provide a framework for processing client requirements to support CE implementation</td>
<td></td>
</tr>
<tr>
<td>Application Domain and Coverage</td>
<td>Specialised Areas: Construction, building and process plants</td>
<td>General Areas: Detailed building and design process, planning, facilities</td>
<td>General: Feasibility, design, procurement, construction</td>
<td>General Areas: Information flow throughout project life cycle</td>
<td>General Areas: Information flow throughout the whole project life cycle</td>
<td>General Areas: Detailed design, procurement, clients, construction, HVAC</td>
<td>Specialised Areas: Client requirement throughout whole project life cycle</td>
<td></td>
</tr>
<tr>
<td>Development Environment</td>
<td>Hardware - PC Software - API tools, DOCSET tools, CAD, XP and CP tools</td>
<td>Hardware - PC Software - DSM, MS Project and AMMP tools</td>
<td>Hardware - PC Software - MS - Visual C++, Super Project, OODBMS</td>
<td>Hardware - N/S Software - EDI/PDI, Scheduling database tools and MS Project</td>
<td>Hardware - PC Software - Under development</td>
<td>Hardware - PC Software - Design - IDEF, MS Project, PROMO, STARLET</td>
<td>Hardware - PC Software - Excel MS Access, ClEntPro</td>
<td></td>
</tr>
<tr>
<td>Integration and Modelling Approach</td>
<td>Centralised project database and object-oriented modelling techniques</td>
<td>Own techniques developed IDEF0v Centralised Information flow modelling technique</td>
<td>Centralised project database object-oriented modelling technique</td>
<td>Centralised information modelling and object-oriented modelling</td>
<td>Distributed information and object-oriented modelling</td>
<td>Centralised project database and object-oriented modelling and IDEFØ techniques</td>
<td>Centralised project database and information flow and IDEFØ Modelling</td>
<td></td>
</tr>
<tr>
<td>Use of Standards</td>
<td>SDAI (STEP part 22) and EDIFAC</td>
<td>None specifies However, STEP and EXPRESS are being investigated</td>
<td>IFC Version 1.5</td>
<td>Reference model framework for data standards, such as STEP</td>
<td>CORBA standards for distributed objects</td>
<td>ISO/STEP EXPRESS Part 11 HTML/ WWW standardisation</td>
<td>ISO/STEP Part 11 EXPRESS</td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td>Available from Newcastle University</td>
<td>Available from Loughborough University</td>
<td>Available from Salford University</td>
<td>Available from the VTT centre, Finland</td>
<td>Available from Salford University</td>
<td>Available from the VTT centre, Finland</td>
<td>Available from Teesside University</td>
<td></td>
</tr>
<tr>
<td>Current and Future Status/Usage</td>
<td>Used for Teaching Research Prototype Used in industry by the original partners and developers</td>
<td>Used for Teaching Research Prototype Trying to commercialise</td>
<td>Research Prototype Extensions being made, but not commercially</td>
<td>Used for Teaching Research Prototype Not developed commercially</td>
<td>Used for Teaching Research Prototype Under development for commercial distribution</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 3

#### Continued....

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Project</th>
<th>COMBINE</th>
<th>GDCPP (II)</th>
<th>RATAS</th>
<th>SPACE</th>
<th>MOPO</th>
<th>MIM</th>
<th>ICON</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aim of Project Model</strong></td>
<td></td>
<td>To develop an integrated building design system (IBDS)</td>
<td>To develop a generic Process Protocol tool that can be adapted and applied to construction projects</td>
<td>To define a conceptual model for structuring all data related to a specific construction building</td>
<td>To develop IT based modelling tools for improving construction process analysis, planning/management</td>
<td>To analyse and improve the flow of management information to construction systems</td>
<td>To develop a method for establishing an integrated database for construction project</td>
<td></td>
</tr>
<tr>
<td><strong>Development Environment</strong></td>
<td></td>
<td>Hardware - Unix Software - AutoCAD - RSC Micro station/services Design software</td>
<td>Hardware - PC Software - AutoCAD - RSC Visio professional</td>
<td>Hardware - PC Software - AutoCAD - RSC MS Project, Excel</td>
<td>Hardware - PC Software - AutoCAD - RSC World Toolkit, RSC KAPPA - PC</td>
<td>Hardware - PC Software - Lotus Notes, MS Project, BPwin, Visio 4.0, ProFacil</td>
<td>Hardware - N/S Software - MS Visio, RSC CAD, EDI CASE tools</td>
<td>Hardware - PC Software - AutoCAD - RSC CA - Super project RSC C++ (CASE) tool</td>
</tr>
<tr>
<td><strong>Integration and Modelling Approach</strong></td>
<td></td>
<td>Centralised project database object-oriented modelling</td>
<td>New product development Stage-Gate approach centralised project process modelling</td>
<td>Centralised project database object-oriented modelling</td>
<td>Centralised project database Object-orientend modelling</td>
<td>Centralised project database and generic process modelling (GEPM)</td>
<td>Centralised information flow database and DFD modelling technique</td>
<td>Centralised project database object-oriented modelling</td>
</tr>
<tr>
<td><strong>Use of Standards</strong></td>
<td></td>
<td>Made to order product model (Bespoke), builds on STEP</td>
<td>None Specified Although, STEP and IFC are under investigation</td>
<td>Makes use of most Standard: IFC/IAI STEP, EXPRESS, Finish Standards</td>
<td>None specified (N/S) Work in progress on STEP/IFC</td>
<td>ISO/STEP EXPRESS Part</td>
<td>None specified (N/S)</td>
<td>None specified, although work is in progress on STEP/IFC</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td></td>
<td>Part 1 used for research/teaching RSC Delft University Part 2 not available</td>
<td>Available from Salford and Loughborough University</td>
<td>Available from VTT and KTH Tec. Centre</td>
<td>Available from Salford University</td>
<td>Only some parts available from Institute of Technology (KTH) Sweden</td>
<td>Available from Reading University</td>
<td>Parts still under development Available from Salford University</td>
</tr>
<tr>
<td><strong>Current Status Future Usage</strong></td>
<td></td>
<td>Abandoned However, extensions being developed by various partners</td>
<td>Level 1 is being used in industry, level 2 and 3 is under development</td>
<td>Research Prototype CD-ROMs and developed commercially</td>
<td>Used for Teaching: Research Prototype CD-ROM demo available</td>
<td>Research Prototype The MOPO Project is still on-going until 2002</td>
<td>Used for Teaching: Research Prototype Published Books</td>
<td>Research Prototype Not developed commercially Limited use</td>
</tr>
<tr>
<td><strong>Type of Approaches</strong></td>
<td></td>
<td>NONE DEFINED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Element Key</strong></td>
<td></td>
<td>IPDB</td>
<td>Product/Process Relations</td>
<td>Combines Different Models</td>
<td>Multiple Models</td>
<td>Conceptual/Core Model</td>
<td>Layering</td>
<td></td>
</tr>
</tbody>
</table>
3.6.1 Survey of Software Tools

The survey and analysis covers an extensive range of software development tools, which have been specifically developed to support various types of modelling notations, such as product and process, static and dynamic, class and object, and many other model types.

The findings of the survey and analysis of the various software application development tools that have addressed the development of information systems is summarised in Table 3.4. It should be emphasised that the findings are largely based on the author's own understanding and familiarity with the software, literature and information provided by the commercial sales managers, as it was impractical to undertake a extensive 'hands on' evaluation of all the software facilities to establish the veracity of the claims made by the software vendors. Furthermore, the use of third-generation languages was also not pursued because relatively more efficient approaches have been reported and used, such as fourth-generation languages (e.g. Visual Basic, C++, Java, etc.) and general-purpose integrated packages (e.g. Microsoft Office), which were considered suitable for assisting in the development of the prototype information system, described in Chapter 5. However, it is important to mention that Microsoft Office was not included in the evaluation, because it has become the dominant computer Windows environment integrated software package for PC users and is bundled with virtually every PC. In addition, a vast amount of literature has been published on it (i.e. everyone knows about it).

3.6.2 Evaluation Criteria

It was considered useful to formulate criteria against which the software tools could be evaluated. The criteria were also intended to provide a common platform for comparison and, while every effort has been made to obtain the relevant information for each tool. There are however gaps in the information available. Once the criteria for the software tools had been determined, it was necessary to prioritise these factors. This part of the process was particularly important because the finding from the software evaluation indicated that it would be impossible to find a software tool, which satisfies all the system requirements. Furthermore, it was becoming clear that more than one tool would be required. The results (in order) of the software evaluation are presented in Figure 3.10, followed by a brief description of the selected software (Visio) used to develop all the information models described in this thesis.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>YES Needs to be ordered</td>
<td>YES Needs to be ordered</td>
<td>YES Special conditions</td>
<td>YES Available from the Internet</td>
<td>YES Download via web</td>
<td>YES Need to pay</td>
<td>YES Order online</td>
<td>YES Only Evaluation</td>
<td>YES Order from the Web</td>
<td>N/A</td>
</tr>
<tr>
<td>Demo Size</td>
<td>100 MB (approx) 32MB RAM</td>
<td>100 MB (approx) 32MB RAM</td>
<td>110 MB (approx) 32MB RAM</td>
<td>90 MB (approx) 32MB RAM</td>
<td>60 MB approx 128MB RAM</td>
<td>50 MB approx 128MB RAM</td>
<td>70 MB approx 64MB RAM</td>
<td>100MB approx 64MB RAM</td>
<td>100MB approx 64MB RAM</td>
<td>90 MB approx 64/128 MB RAM</td>
</tr>
<tr>
<td>Modelling Technique</td>
<td>Object-Oriented Bespoke method UML version 1.3</td>
<td>Object-Oriented Bespoke method UML version 1.3</td>
<td>Bespoke Object-Oriented method UML version 1.2</td>
<td>Bespoke UML version 1.2</td>
<td>Bespoke UML version 1.2</td>
<td>Bespoke UML version 1.2</td>
<td>Bespoke UML version 1.2</td>
<td>Bespoke UML version 1.2</td>
<td>Bespoke UML version 1.2</td>
<td>N/A</td>
</tr>
<tr>
<td>Application Coverage</td>
<td>Part of a suite of products. Intended for software design using the object-oriented approach, and structured around current software technology.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>N/A</td>
</tr>
<tr>
<td>Structural Methodology</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>Supports the unified modelling language, with graphic draw modeler, rather than business or IDEF process methods.</td>
<td>N/A</td>
</tr>
<tr>
<td>Ease of Use/ Simple</td>
<td>NO: Sophisticated piece of software which needs a comprehensible understanding of both Rational Rose and UML standards. Good functionality and product utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>NO: High-sophisticated piece of software that needs a good understanding of both UML/SfP and Sparx. Good functionality and system utilities</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Chapter 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Continued......</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Rational Rose 2000</strong></th>
<th><strong>GDPRO</strong></th>
<th><strong>Software Through Pictures (STP)</strong></th>
<th><strong>Objecteering (UML)</strong></th>
<th><strong>Magicdraw&lt;sup&gt;TM&lt;/sup&gt;</strong></th>
<th><strong>WinA &amp; D Developer</strong></th>
<th><strong>MEGA Suite</strong></th>
<th><strong>BPWin&lt;sup&gt;®&lt;/sup&gt; and Paradigm Plus (PP)</strong></th>
<th><strong>System Architect 2001</strong></th>
<th><strong>Microsoft VISIO 2000</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flexibility</strong></td>
<td></td>
<td>Good integrated scope and flexibility with other software tools</td>
<td>Customization and extensibility through open and published API</td>
<td>Very flexible, can conform to different requirements, good integration scope</td>
<td>Limited: stand alone product object-oriented focused</td>
<td>Limited: Poor flexibility; Allocated to different needs &amp; requirements</td>
<td>Limited: Poor flexibility; Allocated to different needs &amp; requirements</td>
<td>Good flexibility can conform to different needs &amp; requirements</td>
<td>Good flexibility can conform to different needs &amp; requirements</td>
</tr>
<tr>
<td><strong>Adaptability/Integration</strong></td>
<td></td>
<td>Average: Good integration with other tools and upgrading available</td>
<td>Limited: Although integration available with other tools</td>
<td>Good Adaptability: Can be tailored to suit user needs</td>
<td>Limited: Basically used for designing UML modelling systems</td>
<td>Limited: Some integration is available via other software</td>
<td>Limited: Some integration is available via other software</td>
<td>Very adaptable: Can be tailored to suit user needs and scope</td>
<td>Very Adaptable: New simulation features and code generation scope</td>
</tr>
<tr>
<td><strong>Integrity/Checking</strong></td>
<td></td>
<td>None Apparent: Although there is a built-in error message box</td>
<td>None Apparent: Although there is a built-in error message box</td>
<td>None Apparent: Identifies errors in the integrity of tasks and their associated dependencies</td>
<td>None Apparent: How errors are passed through the system and their associated model verification</td>
<td>None Apparent: Known to have error problems in traceability areas</td>
<td>None Apparent: Error messages and model verification</td>
<td>Good: Diagram referential integrity and other features</td>
<td>None Apparent: However there is model traceability and verification</td>
</tr>
<tr>
<td><strong>Drawing Capability</strong></td>
<td></td>
<td>Excellent: Very good graphics, UML: World leader</td>
<td>Average: Sophisticated layout of models</td>
<td>Average: Good Graphics, but tends to be slow and complicated</td>
<td>Average: Slow response and poor graphics</td>
<td>Average: Slow but provides good graphics</td>
<td>Excellent: Top of the class</td>
<td>Good Graphics: Good all round functionality</td>
<td>Excellent: Top of the class</td>
</tr>
<tr>
<td><strong>Software Stability</strong></td>
<td></td>
<td>Very stable: Not likely to experience many sudden changes or configuration problems</td>
<td>Fairly Stable: However, can experience problems when configuring or generating code and reports</td>
<td>Very Stable: Not likely to experience many sudden changes or configuration problems</td>
<td>Good Stability: Limited on features and functionalities. Basic drawing report package</td>
<td>Poor Stability: Problems with configuration in various areas. Many compliances with the software</td>
<td>Poor Stability: Problems with configuration in various areas. Many compliances with the software</td>
<td>Good Stability: Well designed, sophisticated tool, no changes/configuration problems</td>
<td>Good Stability: Unlike to experience any sudden changes or configuration problems</td>
</tr>
<tr>
<td><strong>Reverse Engineering Capability</strong></td>
<td></td>
<td>YES: User Controlled</td>
<td>YES: Supports multiple reverse engineering standards</td>
<td>YES: Supports C++ and Java developer</td>
<td>YES: Supports C++, Java, CORBA, DL multiple maintenance engineering</td>
<td>None Apparent: None Apparent: Although, other MEGA products have data tools</td>
<td>YES: Supports multiple languages, Languages and applications</td>
<td>YES: Support for C++, CORBA, JAVA only</td>
<td>YES: New added features, however, early indication indicate Java, C++ limitations</td>
</tr>
<tr>
<td><strong>Code Generation</strong></td>
<td></td>
<td>YES: Algorithmic code generation C++, VB, JAVA, etc</td>
<td>YES: Generates complete files in SQL, XML, C++, Java, VB, etc</td>
<td>YES: Generates full facilities for a large variety of languages</td>
<td>YES: Support for multiple code languages</td>
<td>YES: Support for multiple code languages</td>
<td>YES: Support for multiple code and DRMS</td>
<td>YES: Support for multiple languages, Languages and applications</td>
<td>YES: Support for multiple languages, Languages and applications</td>
</tr>
<tr>
<td><strong>Report Generation</strong></td>
<td></td>
<td>YES: Basic report generation for text HTML/RTF/TEXT</td>
<td>YES: Data includes text and embedded graphics HTML, RTF</td>
<td>YES: Multiple support for HTML, WORD and RTF</td>
<td>YES: Limited to HTML reports only</td>
<td>YES: Limited to HTML reports only</td>
<td>YES: Limited to HTML reports only</td>
<td>YES: Limited to HTML reports only</td>
<td>YES: Limited to HTML reports only</td>
</tr>
<tr>
<td><strong>Database/Data Dictionary</strong></td>
<td></td>
<td>Multi-repository for storing application models concurrently</td>
<td>Repository based architecture object-oriented Multi-repository</td>
<td>Multi-repository for storing OO modelling data</td>
<td>Multi-repository for storing OO modelling data</td>
<td>Multi-repository for storing OO modelling data</td>
<td>Multi-repository for storing OO modelling data</td>
<td>Multi-repository for storing OO modelling data</td>
<td>Multi-repository for storing OO modelling data</td>
</tr>
<tr>
<td><strong>Applications Integration</strong></td>
<td></td>
<td>GOOD: Integration with a diverse set of tools for integrated product and process model development</td>
<td>GOOD: Provides seamless integration across the software development lifecycle, API/VBA/COM availability</td>
<td>EXCELLENT: Third party integration with various tools, XML/MAP/CAP, Java, Assembly</td>
<td>EXCELLENT: Third party integration with various tools, XML/MAP/CAP, Java, Assembly</td>
<td>LIMITED: XML embedding capabilities and imports to Rational Rose software</td>
<td>LIMITED: XML embedding capabilities and imports to Rational Rose software</td>
<td>GOOD: Good software integration links with the relevant software tools and techniques (ERW tools)</td>
<td>GOOD: Good software integration links with the relevant software tools and techniques (ERW tools)</td>
</tr>
<tr>
<td><strong>Cost One Seat</strong></td>
<td></td>
<td>Depends on customer product &amp; system requirements, no academic discounts</td>
<td>Academic discounts of 40% available on the product Full: £5000 + maintenance</td>
<td>Academic discounts of 30% Full: £1500 + maintenance</td>
<td>No discounts for academic £400 + maintenance</td>
<td>Depends on product edition £1500 all in, no academic discount</td>
<td>Depends on product edition £500 all in, no academic discount</td>
<td>Depends on edition £500 all in, Used by Loughborough University</td>
<td>Good academic discounts £500 + maintenance, Used by Loughborough University</td>
</tr>
</tbody>
</table>
The evaluation indicated that BPwin (CoolGen) and StP recorded the highest marks. However, both products were part of a software suite, which needed to be used together in order to provide the facilities that were necessary to develop the information models. In addition, StP provided inappropriate modelling methods for developing process models of the construction activities. It is also appropriate to mention that many other software tools were evaluated, apart from the application development tools illustrated in Table 3.7. However, many of them had little bearing in relation to the research domain.

3.6.3 Microsoft Visio 2000

Microsoft Visio is a stand-alone modelling tool that enables users to visualise and communicate ideas, generate information, and develop information modelling systems (Visio, 2001). Visio provided the facility to create custom shapes and stencils to support the modelling standards (IDEFO and UML) used in the development of the information models. In addition to supplying a variety of specialised diagram types, and automated diagramming facilities; it also provided import/export tools to various Internet/Web
applications. One of the main characteristics of Visio is its ability to integrate with the Windows environment and other commercial packages such as CAD, Web and XML applications. The main roles Visio played in the research work were:

- **Complements Microsoft Office**: Visio provided the means to create information-rich diagrams/models, which could be exported, edited and extended in Office applications.
- **Facilitates technical design, deployment, and maintenance**: Visio provided the facilities to diagram and document ideas; generate information, and systems to facilitate IT deployments; extend the use of developer tools, and engineering plans.
- **Enables development of custom visual solutions**: Visio provides the ability to create custom shapes and stencils to support modelling standards and could be used to build wide-scale custom visual solutions.

Within each of these roles, Visio provides three key benefits. These benefits include the ability to:

1) **Easily make an impact** by quickly creating and sharing professional-looking diagrams;
2) **Understand ideas, information, and systems** through a broad range of task-specific solutions that integrate with other Microsoft products; and,
3) **Benefit from the diagramming standard** using a single, customisable diagramming program that can be deployed across numerous applications or organizations.

A graphical representation of the modelling capabilities and facilities is presented in Figure 3.11, and is used here to illustrate the software environment in which HIPPY is implemented.

### 3.7. DISCUSSION

Current research on information modelling provides a good foundation for integrated design and construction process information for construction projects. However, many modelling efforts are not developed to the level of detail considered useful for developing practical applications specifically for the domain of on-site construction.
Figure 3.11: Microsoft Windows Platform, illustrating the drawing facilities of Visio, and Generic Method and Architecture of HIPPY
From the discussion of tools and techniques for product and process modelling, it is evident that these approaches are generally different and apart from the relatively new UML, there is a shortage of modelling methods, which are equally applicable to product and process modelling. This lack of effective methods for integrated product and process modelling is compounded by the fact that different aspects of construction need to be modelled with different techniques (Hannus, 1992). It has also resulted in researchers adopting; to varying degrees of success, a range of pragmatic approaches to forge a link between product models developed using one technique and process models developed using another technique. Nevertheless, there are many different types of information models and the intended role of any specific model is not always clear. The majority of information models rely on a general structure of objects, relationships and attributes; all specify relations through the definition of constraints. These constraints in systems allow specifications of database structure, as needed for relational-database integrity. At this level, many information models referred to in this Chapter are similar. However, at a more detailed level, distinctions are apparent in relation to those models that have been developed for general use (i.e. intended to be high-level models that provide a unifying reference for more detailed models that will be constructed on top of them), and models developed for representing specific concepts required to fully represent the actual domain information (i.e. specific response to the needs). The general models lack many of the concepts needed to support actual implementation.

Also, many of the reviewed integrated modelling approaches focus on the integration of construction applications through the use of monolithic product models (sometimes referred to as conceptual or type models), in that they define the data representation; that is, provide the basic entities and relationships, and possibly the attributes used to characterise the objects needed for capturing and representing information about some domain. While the use of these models have been criticised for their complexity (Eastman, 1992), the use of partial models capable of exchanging information is not seen as an ideal solution, given that the models may become dated and too narrow, and imply data exchange (Jägbeck, 1996). Nevertheless, most of these model types strive for better relationships between product and process. On the other hand, another approach is to draw upon the basic perspective that processes have inputs and outputs, as embodied in the IDEFØ models (Chung, 1989; Sanvido. 1990), where these inputs and outputs are various product entities, which may be broken down in to sub-entities (Froese, 1996).
Alternatively, several researchers have focused their efforts on the dynamic approach, and developed modifications, variants and enhancements to established methods and tools, so as to accommodate their desired modelling perspective. For example, Abeysinghe & Phalp (1997), combined RAD with CSP (Communicating Sequential Processes), a formal process modelling paradigm based on concurrency and communications), while Kartam, et al. (1997), proposed a 'work-mapping' model that has its roots in the conventional system conversion model but incorporates some features of SADT. It is also recognised that these variants and hybrid models extend the capabilities of existing methods. However, process models still represent only a partial representation of the development of an artefact, as they usually hold no information on the end products of sub-processes or the final end product of the overall process.

Based on the concept of an integrated product and process (project) model, it was also acknowledged from the discussion on information modelling and communication technologies necessary to implement an effective communications infrastructure for construction team members, are in existence. These technologies are able to support a range of approaches and architectures but there is need to some level of rationalisation and standardisation to avoid duplication of efforts and ensure true and long-term interoperability. According to Anumba, et al. (2000), several issues that need to be addressed include:

- how best to utilise the technologies to implement a model that provides the core functionalities required, particularly, the flow of information between project team members and across stages in the project life cycle;
- how to overcome the socio-technical barriers of using an integrated product and process model thereby engendering a culture of collaboration and information sharing.

It is important to acknowledge that a basic and flexible approach, which allows for the integration of product and process modelling methods, in addition to the many legacy IT systems within the constriction industry, will be favoured for addressing the many problems prevalent in the construction industry. Such an approach will need to be based on both established and emerging standards such as STEP, XML, CORBA, UML etc., and also embody developments such as object-oriented methodologies and virtual reality and/or Internet technologies. These emerging standards are likely to have the most impact, in that they will facilitate 'seamless integration, between the heterogeneous software tools
and IT systems of construction team members. Furthermore, many of the modelling tools are increasingly being based on the object-oriented modelling paradigm, which has beneficial implementations for integrated model development and re-use. The advent of independent standards such as UML and enhancements to established application development software tools have an important role to play in the development of an effective configuration management information system for the integrated product and process models, and in enhancing the flexibility of interactions between individual disciplines and the integrated conceptual model.

Therefore, the architecture of the integrated product and process model (described in Chapters 4-5) is of the utmost importance in ensuring that it serves its intended purpose and delivers the potential benefits outlined earlier and further discussed in Chapter 7. Two issues can be considered vital in the representation and communication of integrated information, as proposed in the integrated product and process model: the degree of sharing required and the kind of information to be shared. The early resolution of these two aspects of information sharing; in addition to appropriate modelling standards, is vital in transforming the construction industry, which has a traditionally conservative and confrontational culture, into a collective environment where construction project information is readily shared via an integrated product and process model. The cultural issues in the use of an integrated product and process model need to be addressed simultaneously with development of an appropriate architecture. In this regard, there is growing need for better education and training of construction personnel, the documentation and dissemination of case studies that demonstrate the business benefits of collaborative working, and reducing the cost and other barriers that militate against the adoption of new technologies.

It is one of the goals of this thesis to propose an architecture for a generic integrated product and process information modelling system to effectively support collaboration and data exchange (communications) between project team members during all stages in the life cycle of on-site construction (i.e. from construction to operation of small to medium size building projects).
Chapter 3

3.8 SUMMARY

This Chapter has described the research domain from an information handling perspective, by reviewing the state-of-the-art in integrated developments, information modelling methodologies, including product, process and integrated project models, established and emerging standards such as STEP, CORBA, UML, XML, IFCs, and object-oriented approaches for addressing integration in construction. The Chapter also reviewed several related research projects within the domain of this thesis. The techniques most appropriate to model on-site construction have also been described in this Chapter. IDEFØ and UML have been identified as an appropriate modelling methodology to satisfy the requirements, and have been described in detail.

Many of the existing models fall into a number of categories: those that are represented by product interoperability standards or languages such as EXPRESS, NAIM, etc., and those that are represented by process modelling methodologies such as IDEFØ, RAD, DFD, etc., and those that combine the functions of both to create an integrated project model. Although these project models have, in many cases, successfully demonstrated the sharing of construction product and process information, there is significant benefits in adopting a unified approach to integrating them based on product and process models of the entire construction project.

In summary, several of the modelling approaches are either not applicable to modern construction works or too specific to an organisation’s needs. Many of them are also rather theoretical and too generic in nature, and have not yet reached a stage where software companies can implement them. Thus, there is a growing need to produce less generic and more unambiguous proposals, possibly for a specific domain, such as construction management, and to test them in real-life construction situations. Modelling the construction process has a tendency to focus on a particular viewpoint. A more flexible information system can be developed by combining various activity and object-oriented modelling techniques that integrate multiple applications and viewpoints. The product and process information models described in the next Chapter aims to resolve some of the issues discussed here and to provide the basic support for developing an integrated product and process information modelling system, described in Chapter 5.
CHAPTER 4

INTEGRATED CONCEPTUAL MODEL

4.1 INTRODUCTION

This chapter provides a description of a generic construction process model and a generic building product model, in terms of activity functions and system requirements to enable an integrated conceptual model of the on-site construction process to be developed. It shows the importance of integrating information; in particular, the problem of capturing ill-defined, informal real world, and user requirements, and transforming these into clear, formal and consistent specifications. The features and characteristics of the product and process models are also discussed along with the requirements to develop an integrated conceptual model for on-site construction, with house building as a test bed. The trialling of the generic product and process models through their representation of modelling techniques and construction processes are also discussed.

The chapter starts by presenting a conceptual description of the building delivery process, using the principle of 'general to specific'; that is, describes the information requirements for the proposed integrated information modelling system (HIPPY), the methods used to develop the conceptual model, and their components; classes, objects, entities, attributes; and, resources, operations, functions, tasks, events, etc. The chapter then presents an conceptual information framework and model structure, which integrates the features and characteristics of both generic product and process models. It discussed the approaches used to develop a set of generic construction process models and building product models, and the benefits of their use are described. In addition, a description of the formal process model of the building process using IDEFØ notation and the building product model using UML notation, both in terms of scope and structure is presented. Finally, a integrated conceptual model of the generic product and process models is described; together with details of how the integration of IDEFØ and Scheduling is used in concurrence with UML object class, Use Cases and Sequence diagrams is achieved. The chapter, taken as a whole, is a domain description and a basis for the further research work that is described in the following chapters.
4.2 CONCEPTUAL DESCRIPTION OF THE BUILDING DELIVERY PROCESS

4.2.1 Background

Improving the building delivery process (sometimes referred to as the project delivery process) has been the major aim of many researchers and practitioners within the construction industry (CIC, 1996; Dado & Tolman, 1998; DETR, 1998; Kagioglou, et al. 1998; Baldwin, et al, 1999). There is currently much discussion (Latham, 1994; CIB, 1997; Accelerating Change, 2002) on how the construction sector can improve its building processes so as to become more efficient and competitive in the delivery of facilities, which satisfy, the industry’s clients (Anumba & Amor, 1999). An extensive literature review has been conducted as part this thesis to highlight relevant advancements in the building delivery process. The results indicated that many recent studies (some of which have been government sponsored) have exposed the weaknesses of the construction industry and set performance targets that the industry must achieve. It also revealed that leading construction companies have specifically addressed the subject of radical changes to current construction practices with the objective of reducing construction time by up to 40%, through the use of change strategies, such as CE, and the integration of product and process models, and IT (Latham, 1994; Construct IT, 1996; CIB, 1997; DETR, 1998; Egan, 2002). The industry is responding to these challenges and several initiatives (described in the preceding chapters) have been started to facilitate and demonstrate efforts geared towards a rethink of the building delivery process (Anumba & Amor, 1999).

This section presents a multidimensional entity model, as an information framework for on site construction, in order to illustrate and facilitate the implementation of product and process information, which is applicable to the management of information during the on site construction stage, with house building as the focus point.

4.2.2 Definition and Terminology

It is generally agreed that if several building experts are asked to define exactly what a construction process is, they would probably convey twenty different answers. Thus, it is logical to say that the building delivery process (BDP) can be viewed from different perspectives and can include many different construction elements. Therefore, before starting a discussion, there must be a clear mutual understanding of what is meant by the ‘building delivery process’; here, the on site construction processes for building a facility.
Chapter 4

In the building industry a construction process is often defined as a sequence of activities performed using different resources (both material and human) on construction entities or objects. Whereas, the building delivery process can be viewed as a combination of conversion activities, where inputs are transformed into outputs (value-adding activities) and flow activities (the non value-adding activities such as moving, waiting, inspecting, cleaning, etc.). In this context, a process is often associated with terms such as activity, operation, task, event, cost, duration, etc. Therefore, clear definitions of these terms need to be established.

This research, adopted the preposition that any methodology for the consideration of operations and processes of construction must consider the hierarchical nature of construction. According to Halpin & Riggs (1992), six levels of hierarchy are definable in construction management. These include:

1. **Task Level:** The task level is concerned with the identification and work assignment of elemental segment of work (i.e. work task) to resources (agents, equipment, plant, etc.), which can be defined further as:
   - A work task is the basic description unit in construction practice, and is the fundamental building blocks of processes and operations.
   - A work assignment is the collection of work tasks specifically assigned to resources for performance. These usually involve sequences of work tasks appropriate to a specific trade or resource skill level and, therefore may define a construction process.

2. **Process Level:** A construction process is a unique collection of work tasks and assignments related to each other through a well-defined structure and sequence, which represents an identifiable segment of a construction operation.

3. **Operation Level:** The operation level is concerned with the technology and details of how construction work is performed. The construction operation results in the placement of a definable part of a product (i.e. closely related to the means of achieving an end product) and is a synthesis of construction processes.

4. **Activity Level:** A construction activity is a time and resource-consuming element of a building project normally defined for the purpose of time and cost control by a project planner or scheduler, estimator or cost engineer, and is directly related to a definable part of a product. An activity is the aggregation of operations or processes for the
purpose of project scheduling and costing, and therefore only has implicit information relating to how construction is performed.

5. **Project level:** The project level is dominated by terms relating to the project breakdown for the purpose of time and cost control (e.g. the project activity and the project cost account). In addition, the concept of resources is defined and related to the activity as either an added descriptive attribute of the activity or for resource scheduling purposes.

6. **Organisation Level:** This is concerned with the legal and business structure of a company, the various functional areas of management, and the interaction between departments and personnel performing these management functions.

A breakdown of the typical items in the process at each level of hierarchy is presented in Table 4.1. At the organisation and project levels, activities within the schedule related to the construction works of the sub/superstructures of a detached house unit. At the work task level, location, unloading and loading, selecting and laying, and other resource related activity is required. The distinction between an activity and an operation is firmly tied to the duration of the function and whether it is primarily concerned with a physical segment unique to the time cost control of the projects (activity) or to the technological process or method required to achieve a specific end product (operation); in this case, a house facility. The operation is therefore more fundamental for an understanding of on site (field) building methods.

**Table 4.1: Construction Project Example Displaying the Hierarchical Breakdown of Terms in Construction Management.**

<table>
<thead>
<tr>
<th>Hierarchical Level</th>
<th>DESCRIPTION AND BASIC FOCUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organisation</td>
<td>Company Structure, Project Focus and Site Functions: Project #1: Detached house with masonry walls and pitched roof tiles.</td>
</tr>
<tr>
<td>Project</td>
<td>Project Plans &amp; Definition: Construction of Substructure works, Superstructure, Services, and Interior Fixings and Furnishings.</td>
</tr>
<tr>
<td>Activity</td>
<td>Time and Cost Control: Concrete to foundations, £1,000, 2 days, 4 labourers &amp; concrete to ground floor slab, £1,500, 3 days, 4 labourers.</td>
</tr>
<tr>
<td>Operation</td>
<td>Construction method: External brick envelope to include the installation of 8 UPVC window frame systems, bricklaying operation using scaffolding and labour.</td>
</tr>
<tr>
<td>Process</td>
<td>Trade Actions: Brick delivery process by labourers and forklift; erect, move and adjust independent scaffold unit; lay brick process by bricklayers.</td>
</tr>
<tr>
<td>Task (work)</td>
<td>Skill and Labour Level: Locate and load out spot boards with cement mortar; load scaffold with bricks; select, cut and lay bricks.</td>
</tr>
</tbody>
</table>
4.2.3 Factors Affecting the Productivity in the Building Delivery Process (BDP)

As previously explained in Chapter 1, many countries’ construction activities in one form or another accounts for at least half their annual expenditure on capital goods (Kimmance, 1999). Therefore, an increase in productivity throughout the construction sector should not only raise the earnings and profits of those working in the sector, but also will contribute to an improvement of productivity in other sectors. If this is to be achieved, a combination of resources, including information technology (IT), land, buildings, materials, mechanical plant, tools, equipment and manpower need to be used efficiently. It is therefore the task of construction site management to combine these resources, as illustrated in Figure 4.1, to enable the delivery of a building to be executed expeditiously and economically.

Realisation of the building delivery process involves the relationships and joint-efforts from many different parties with there being an abundance of alternatives to organise the tasks and activities needed. However, experience has shown that there is always scope for improvement in building facilities. It has often been suggested that the delivery of a building is a slow meticulous process, which could also be extended to suggest that
because construction site erection times are greater, costs must also be higher; thus, the delivery of a building will be delayed (Ferry & Brandon, 1991). Figure 4.2 illustrates the difference between actual time and productive time for a standard housing project. The basic work content is the absolute minimum time in which a house or building operation can theoretically be completed. In practice this is hardly ever achieved, due to the following factors:

**Figure 4.2 Work content for a typical housing project**

**Non-Added Value Activities:** This includes repetitive work due to poor workmanship, repeated work inspections, delays in the delivery of materials, insufficient amount of materials, defects in materials, poor design and specifications, poor site management and supervision, and inefficient methods of construction. It is also appropriate to mention that due to the nature of the building delivery process, the proportion of non value-adding activities in projects have not fully be established; yet it is believed to be significant. Therefore, non value-adding activities need to be reduced and kept to a minimum, if substantial time reductions and cost savings are to be achieved, while maintaining the highest quality work standards.
Unproductive Time: This occurs when workers, equipment and machinery are standing idle. Unproductive time may be due to extraneous reasons beyond the control of site management, or as a result of the shortcomings of site management or labour. Unproductive time factors may include poor weather conditions, poor interfaces and coordination between building trades or sub-contractors. Although some of these are beyond the control of site management, anticipation through risk assessment and efficient management and work scheduling can often lessen their effects.

Poor Management: The main factors associated with poor management include the following:

- delays in obtaining necessary information such as working drawings, specifications, plans, etc.;
- delays in the delivery of materials, plant and equipment, as a result of poor planning;
- inadequate planning of work sequences, resulting in workers and machinery to stand idle;
- poor supervision;
- poor health and safety management; and,
- poor management of workforce.

Shortcomings of Workforce: The main factors associated with the shortcomings of labour include the following:

- poor attendance (e.g. taking time off without good cause, failing to start work promptly (lateness), idleness or deliberate slowdown;
- poor workmanship, resulting in rework;
- failure to observe safe working methods causing accidents; and,
- poor maintenance of building tools, equipment, plant, etc.

The above issues are by no means exhaustive and to a certain extent used here to identify some of the shortcomings associated with construction activities within the building delivery process. In general, problems are related to inadequate site control and poor management rather than the shortcomings of the workforce. To reduce both the total work content and effective time, the main task of site management is to organise and control all available resources in order to achieve maximum efficiency and productivity, that is, reduce the actual construction time in order to minimise unproductive time. The use of
integrated information systems capable of managing information essential to the building delivery process offers significant potential for enabling the construction industry to achieve its current and future challenges.

4.2.4 Information Scope for House-building

The lifecycle of a building usually evolves through design, construction, operation, maintenance and demolition, and requires the expertise of many participants using a variety of resources. The parties involved during this cycle, particularly with regards to design and site construction, would typically involve a variety of individuals such as clients, contractors, specialists, suppliers, and construction managers, all exchanging large quantities of complex information at different times and locations. The type of information and extent of detail would clearly be related to the project and the parties involved, but would typically include information on: resources, time, cost, quality, health and safety, environment and administrative, technical, financial and legal information, and the systems and procedures that they give rise to). Figure 4.3 presents a multidimensional entity model, representing the scope of information needed to construct a house-building. The model consists of four axes, namely, the product, process, actor and resource axes, and is used here to illustrate and identify the research domain.

4.2.4.1 Product Axis

This axis describes the physical system, including subsystems, and components, which constitute the house-building. A house unit consists of several major systems, such as Substructure (foundation, footing, slab), Superstructure (wall, floor, roof, windows, etc.), Services (water, gas, electric, etc.), and Interior fixture and fittings, furnishing, etc. These systems can be further decomposed in smaller subsystems, components etc., all the way down to the material elements such as bricks and blocks, concrete, reinforcement bars, and steel members, timber trusses or joists, doors and windows, etc.

The work presented in this thesis concentrates on the substructure and superstructure systems (highlighted in Figure 4.3) of a house-building. The function of these systems is to protect the end users (house owners) from fluctuating outdoor weather conditions, in addition to providing accommodation, shelter and security (i.e. place of safety from adverse or unseen conveniences).
4.2.4.2 The Process Axis

The traditional construction process is divided into many stages (RIBA, 1997). The process axis represents the various stages of construction. Each stage, represented by activities and/or functions, can be decomposed in more detail to show the operational processes involved in constructing the house-building. These stages can be grouped into three major phases: the decision phase, the design phase and the construction phase. Once a building is completed, it goes through an operational phase and possibly several renovations. The life
of the building ends when it is demolished. The work presented in this thesis focuses on
the on-site construction phase, with house-building as a test bed. One of the aims of the on-
site construction phase is to identify and analyse the relevant process information
(activities, operations, tasks, etc.) needed to construct the house-building. Table 4.2
presents a typical procedure of the building activities (rough sequence of operations)
needed to construct both the substructure and superstructure of a house unit.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Activity/Operation Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Demolition of existing structures</td>
</tr>
<tr>
<td>2.</td>
<td>Clear whole site area</td>
</tr>
<tr>
<td>3.</td>
<td>Set up building cabins and lock-ups</td>
</tr>
<tr>
<td>4.</td>
<td>Excavation of building area</td>
</tr>
<tr>
<td>5.</td>
<td>Removal of excavated material</td>
</tr>
<tr>
<td>6.</td>
<td>Levels and pegging to foundation</td>
</tr>
<tr>
<td>7.</td>
<td>Delivery of/or mixing concrete for foundation</td>
</tr>
<tr>
<td>8.</td>
<td>Pouring concrete to foundation</td>
</tr>
<tr>
<td>9.</td>
<td>Compacting and tamping concrete foundation</td>
</tr>
<tr>
<td>10.</td>
<td>Inspection of foundation (feedback)</td>
</tr>
<tr>
<td>11.</td>
<td>Load bricks and blocks for footings</td>
</tr>
<tr>
<td>12.</td>
<td>Mix mortar materials for footing</td>
</tr>
<tr>
<td>13.</td>
<td>Load mortar for footings</td>
</tr>
<tr>
<td>14.</td>
<td>Construction of footings walls to dpc level</td>
</tr>
<tr>
<td>15.</td>
<td>Drainage and pipe ground works</td>
</tr>
<tr>
<td>16.</td>
<td>Concrete backfill below damp proof course (dpc) level</td>
</tr>
<tr>
<td>17.</td>
<td>Compact soil area for concrete ground floor slab</td>
</tr>
<tr>
<td>18.</td>
<td>Clean and Inspection of all substructure works (feedback)</td>
</tr>
<tr>
<td>19.</td>
<td>Delivery of/or mixing of concrete materials for ground floor slab</td>
</tr>
<tr>
<td>20.</td>
<td>Pouring concrete to ground floor slab</td>
</tr>
<tr>
<td>21.</td>
<td>Compacting and tamping to concrete slab</td>
</tr>
<tr>
<td>22.</td>
<td>Delivery of and/or erect scaffold and ladders</td>
</tr>
<tr>
<td>23.</td>
<td>Load bricks and blocks for superstructure frame</td>
</tr>
<tr>
<td>24.</td>
<td>Mix masonry mortar for cavity wall system</td>
</tr>
<tr>
<td>25.</td>
<td>Load mortar using available equipment</td>
</tr>
<tr>
<td>26.</td>
<td>Installation of damp proof course and adhesives</td>
</tr>
<tr>
<td>27.</td>
<td>Set out house materials (frames, arches, lintels, etc.)</td>
</tr>
<tr>
<td>28.</td>
<td>Construction of masonry cavity wall system above dpc</td>
</tr>
<tr>
<td>29.</td>
<td>Roof frame erection (completion of roof, felt, battens, tiles, etc.) ceilings made</td>
</tr>
<tr>
<td>30.</td>
<td>Clean, brush, and inspect brickwork and roof superstructures (feedback)</td>
</tr>
</tbody>
</table>

4.2.4.3 Actor Axis

This axis lists the main participants involved, directly or indirectly, throughout the building
construction phase. An actor represents anyone who interfaces with a system, and is
always external to the system being modelled. An actor’s roles may be performed by one
person for small projects, or by several people on larger projects. Actors involved in the
building construction phase generally include the construction management team such as: client’s representative, design team, project/site manager, engineers (planning, structural, electrical, etc.); contractors and sub-contractors (building, plant, etc.); specialist contractors, labourers, and building controllers. A list of the main actors involved in the construction of the sub/superstructure systems can be found in Table 4.3. It is also acknowledge that in larger projects other disciplines would usually be involved at the building construction phase, such as consultants, environmental engineers, geologists, etc.

Table 4.3: Actors Involved with Constructing Sub/Superstructure Systems for a House

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site Manager:</td>
<td>The person who is in charge of running the building site and all its building activities.</td>
</tr>
<tr>
<td>Plant Contractor:</td>
<td>The Company who supplies plant items (machinery, lorries and equipment) for clearing, excavating, digging, removing and transporting building materials (topsoil, concrete, bricks, etc.) on the building site.</td>
</tr>
<tr>
<td>Site Engineer:</td>
<td>The person who performs surveying and engineering tasks such as setting out the building site, planning, supervision, inspection and materials testing.</td>
</tr>
<tr>
<td>Building Materials Supplier:</td>
<td>The Company who supplies and delivers building materials to the building site.</td>
</tr>
<tr>
<td>Forklift Operator:</td>
<td>The person who distributes building materials on the building site using a forklift truck.</td>
</tr>
<tr>
<td>Cement Mixer Operator:</td>
<td>The person who operates the cement mixer for making brick mortar on the building site.</td>
</tr>
<tr>
<td>Bricklayer Contractor:</td>
<td>The person who lays bricks and blocks (builds brickwork structures) on the building site.</td>
</tr>
<tr>
<td>Building Labourer:</td>
<td>The person who performs heavy manual labour chores and semi-skilled building tasks on the building site.</td>
</tr>
</tbody>
</table>

4.2.4.4 Resources Axis

This axis and possibly the most important, lists the resources used for a particular activity by associating the activity with the objects throughout the building construction phase. The resources axis is also closely linked to the actors involved within the construction process, given that actors are generally regarded as human resources. The rationale for separating actors from the resources axis is associated with the building product model, which will be discussed later in this Chapter.
Chapter 4

The resources have been divided into groups to represent the physical or information elements (classes) described in the building product and process models (discussed later in this Chapter). Classes of resources include financial, time, human resource, architect, engineers, etc., physical (materials, plant, equipment, tools etc.), workspace (area, volume, etc.), and information resources (standards, regulations, building documents, etc.). Some elemental resources can be further defined at lower levels such as a particular component or composite elements (e.g. concrete or bricks, doors, windows, JCB, concrete mixer, etc.). However, this will depend on the classification and type of relationship between element types. Not every process will require resources from every category. The critical attributes defined for resources might include type, availability, supply, function, quality, and cost.

4.3 CONCEPTUAL INFORMATION MODEL

The combination of products, processes, resources and actors provides a consistent representation of the building delivery process. One of the main aims of an integrated product and process model is to automate and combine certain tasks during the overall construction process, and to provide support to designers, construction managers, and contractors in an integrated fashion. The use of standardised computer data representation that integrates product and process information would greatly improve the communication of design, planning and building decisions, analysis results and comments between the various actors involved. For this purpose, a conceptual information model (modelling information framework) that integrates both product and process concepts is proposed.

4.3.1 Functional Requirements

The need for integrated information modelling systems capable of accommodating both product and process information is becoming vital for construction companies in acquiring, structuring, manipulating, and exchanging complex technical data, to be handled during the construction and operation stages of SMsBP. In this regard, the proposed information modelling framework is mainly aimed at small to medium sized enterprises (SMEs) who often have to face situations for which they do not have the necessary skills or tools for the continuous updating of the technical information needed by the projects they work on or the parties they work with. Furthermore, the conceptual information model will need to differentiate between the relational structuring of product and process models, so as to enable a selection of competitive solutions intelligent enough to meet the identified
requirements and given specifications. To facilitate this, the main characteristics of the conceptual information model are illustrated in Figure 4.4, and have been developed to meet the following requirements:

- to facilitate a comprehensive data structure for storing, managing, and retrieving information during the on-site construction stage of house-building;
- to define the categories of information required to support IT applications and the management of information systems during on-site construction phases;
- to combine text data (e.g. regulations and standards, technical drawings, specifications, environmental controls, quality, safety, etc.) and multimedia data (e.g. graphic pictures and Web images of house-building sections, plant, equipment, building elements, etc);
- to facilitate information generated from heterogeneous sources to sustain an efficient decision-making process, namely from material manufacturers, government regulations; project data (time, durations, etc.) resource textual and spatial data (size, measurements, etc) generated during the design and construction phases, in addition to sources such as human resources essential for the management of information systems;
- to facilitate the integration of IDEFO process models with UML product models through the use of hyperlinks and OLE data transferring techniques set-up within ODBC applications; and,
- to proceed from existing domain information, modelling techniques and other concepts.

A number of other requirements need to be considered with respect to developing an conceptual information model for construction management purposes, which can be used to facilitate the integration of product and process information. Some of these can be derived from general requirements for any information model or modern IT application (e.g. prototyping, graphical user interfaces). Others are requirements that apply to any database system (e.g. non-redundancy of data). A final category consists of requirements that are particular to the on-site construction stage (e.g. information needs to cover the life cycle of a building, and a large percentage of this information needs to be reusable several years later on). An information model should provide general requirements such as:

- **Flexibility:** The model can be used in different situations and meet different requirements as they occur;
- **Ease of Use:** The model should be intuitive;
- **Cost Effective:** The resources spent on developing the system model should be kept to a minimum in comparison with its benefits.
Figure 4.4: Information structure in the proposed Conceptual Information Model
4.3.2 System Architecture to Support Information Integration

The concept of a central, unified data repository and user interface implemented as an integrated data sources to support on site construction appears to be a promising strategy for enabling data exchange and sharing. Figure 4.5 presents a logical view of the conceptual framework to support the conceptual information model illustrated in Figure 4.4. The conceptual framework shows the integration and relationships between many elements through the use of CASE tools, generic product and process modelling techniques, graphical user interface, multiple hyperlinks to MS/HTML document files, and object linking and embedding to other soft applications (Word, Excel, Adobe, etc.). The data repository contains information, which was captured and represented in the product and process models described later in this Chapter. This information is structured in terms of objects (entities and classes) and represents a common data store for construction information. The combination provides a consistent representation of the construction facility (defined object elements of information) and the means needed to construct the facility. The advantages of such a concept are summarised as follows:

- It stores data in a structure, which is independent of any construction management application. Therefore different construction applications can access and share consistent data.
- It promotes data reusability by eliminating re-entry and redundancy.
- It does not constrain the data structure to a specific type of database, i.e., it can be implemented in relational database or in an object-oriented database.
- It allows for the data structure to be scaleable, in terms of adding new objects or new attributes to existing objects.

Figure 4.5 illustrates the main features and requirements (i.e. presents a graphical representation of the functional reasoning), which were evaluated, to provide a description of the integrated information model and the building processes to be devised. A description of the main characteristics of the integrated information model is as follows:

- The integrated modelling system evolved from the requirements of the applications that use it, and the analysis of the fundamental concepts involved in on-site construction.
- The system architecture (conceptual framework) consists of a two-layered approach, which separates the document data files from the product and process models.
Chapter 4

BUILDING PRODUCT MODEL

House Components

Static Structure
- UML Diagrams
- Object Oriented Modelling method
- Standard Product Object Definition
- Hierarchy Format
- Software Tools

Microsoft Visio 2000 Modelling Tool
ODBC Database Storage

CONSTRUCTION PROCESS MODEL

Construct House Components

Dynamic Structure
- Process Diagrams
- IDEF0 Notation
- Activities, Sequencing, Resources, Actions
- Standard specification
- Decomposition hierarchy
- Software Modelling Tools

User Interface
Direct Hyperlinks to:
- Product/Process Models
- MS PowerPoint Slides
- Presentations, Word Files
- HTML Document Files
- MS Access Database
- VBA Code, Quality (QS)
- OBE Transferring Data

Data Flow

Product Class Model
House Unit

Decomposes into Relationships

Integration

Hyperlink relationships to product and process information

Data Repository

House Unit

A1

A2

A3

A4

A31

A32

A33

Click to Hyperlink to Project Scheduling information

Click to view Use Cases scenarios

Click on box to show hyperlink to MS document information files

SuperStrut

SubStrut

Wall

Roof

Slab

Foundation

Is made up of UML diagrams

Component Activity Link

Is made up of IDEF0 diagrams

3D Building House
CAD Model

Web Graphics, Images & Pictures

Finished End Product
Outcome of Integrating Product and Process Information

3D Building House
Outcome of Integrating Product and Process Information

Figure 4.5: Logical view of the integrated information modelling environment.
- Given that many modelling approaches are activity based, the activities will act on the system components, or object entities that are at different levels of detail in the product model (e.g. building foundations and footings are sub-systems of substructure, which in turn is a sub-system of structure).

- Activities in the construction process model (construct house_components) need to act on the component in the building product model (house_component). These components will be classified using a standard object definition such as, UML, STEP (part 21), IAI (IFC v. 1.5), etc., and arranged into a decomposed hierarchy, which can be further decomposed into levels and sub-levels to show more detail.

- The generic construction process model (GCPM) exploits the SADT notation and is composed of activity diagrams with hyperlinks connected to scheduling information, which represent the time and duration variables of the construction processes used to create the facility described by the generic building product model (GBPM).

- The unified modelling language (UML) will be used to develop a compliant building product model of a house facility as input, consisting of object classes, represented by Use Cases and Sequence diagrams for identifying the interaction (scenarios analysis) between the object classes.

- The generic building product model consists of class definitions using UML class notation, which represent the construction concepts and information flows. These will be arranged into major categories and sub-categories (i.e. product elements, components, and sub-components consisting of associations, relationships, attributes and operations arranged in a hierarchy format). The object classes represent the actual house-building being constructed (described by the process model), and its representation is central to all reasoning about the house products.

- While the product model describes the state of the facility, the process model describes the processes used to create the facility – the house-building construction processes, which are also arranged into major categories and sub-categories consisting of relationships, associations, and hyperlinks to product and process information arranged in a hierarchy format.

- For integration purposes, the product model will need to contain information relating to the types of components in the facility, the size and location of these components (attributes), the composition of the facility into various systems and sub-systems, and supported by relationships between these components and systems.
• The activities are linked through hyperlinks and OLE data transferring methods (discussed in Chapter 5) to the product object class components in a decomposed hierarchy format, similar to the RATAS model (described in Chapter 3), so as to support the generation of hierarchical process activities. This will support the relationships between product components and process activities, allowing a common understanding between the models.

• Product and process models, diagrams, scheduling charts, graphical pictures, Microsoft and Web document files will be accessible through the use of an graphical user interface (GUI), developed in a standard MS Windows environment, generated with a standardised forth generated programming code language (Visual Basic), so as to be compatible with the software applications and modelling techniques.

• All software applications/tools will incorporate (have as a feature) an Open Database Connectivity (ODBC) as their strategic interface for accessing data in a heterogeneous environment of relational and non-relational database management system. ODBC is based on the Microsoft’s Call Level Interface specification of the SQL Access Group, and will provide an open, vendor - neutral way of accessing data stored in personnel computers, laptop computers, minicomputers and mainframe databases.

4.4 THE GENERIC CONSTRUCTION PROCESS MODEL (GCPM)

4.4.1 Development of the Process Model

The development of a generic construction process model involved an iterative process, to ensure its relevance to, and practical application in, the construction industry. This involved three distinct stages: review of modelling techniques and tools, identification of requirements (scope and purpose), and identification of on-site construction activities.

4.4.1.1 Review of Modelling Tools and Techniques

The need for a suitable tool to represent the methodology for processing both product and process information necessitated a review of modelling tools. In a sub-task (objective) of this research, the modelling capabilities of various software application development tools and modelling techniques were analysed and compared (see Chapter 3). The outcome of the evaluation identified IDEFØ as the most appropriate technique to model the on-site construction activities because:
it deals with functional/activity modelling which is appropriate since one of the requirements was to facilitate the required dynamic process activities for describing the on-site construction information of a facility;

it serves linear processes (in this case, the processing activity) very well and, in addition to functional modelling, it also provides for information and dynamic modelling, thus facilitating the development of a comprehensive process model;

it is relatively easy to use and understand, and is extensively used for integrating many change philosophies (e.g. CE, BPR, QFD, etc) for which efficient software is available;

it is most common used methodology for activity modelling in the A/E/C domain and beyond, and has been proven to be suitable for modelling the construction process (e.g. Sanvido, et al. 1990; Hannus, 1992; Karhu, et al. 1997; Kamara, 1999) since it provides for the decomposition of an activity in to levels of detail.

4.4.1.2 Identification of Requirements

Taking the functional requirements described previously in this chapter as a starting point, a generic construction process model (GCPM) of the on site construction processes was developed. The scope of the process-oriented model spans the main aspects of design, development, and implementation of a generic integrated information modelling system, developed from the viewpoint of a construction manager (or other professionals) who have oversight over the information requirements, which represents a more holistic view of the activities involved in the building delivery process. The generic construction process model incorporates a framework that:

- manages the required dynamic process activities for describing the construction of a new building facility (i.e. house dwelling), and minimises bias in decision-making;
- facilitates external links to the overall design and construction documentation of the building (e.g. resources, product data, building standards, material/plant specifications, environmental regulations, such as CDM, health and safety issues, etc.);
- facilitates external links to project planning/scheduling information (e.g. sequence of tasks, time durations, schedule charts, etc.);
- can be implemented in a computer environment to exploit the benefits of IT, as well as facilitate its integration with other stages in the construction process; and
- facilitates collaborative working through its output, and the process that is adopted to generate those outputs.
The purpose of the high-level GCPM is to achieve a basic structure on which a particular specific system, together with its applications, could be built as a layered architecture. The GCPM should facilitate the management of information essential to construction team members and the integration of the construction process, as well as reducing the non value-added activities, which influences the way in which activities are performed, and accounts for a large percentage of wasted time, and increasing cost.

### 4.4.1.3 Identification of Construction Activities

Following a detailed review of environmental constraints, detailed mapping of the identified activities within the on-site construction process were modelling and a hierarchical structure determined. The information and functional requirements of each individual activity at the lower levels of hierarchy were then identified, allowing the process model to be constructed. The processes were then mapped and modelled on to four levels (depending on activity) of details using a CASE tool called Microsoft Visio 2002.

### 4.4.2 Description of the Generic Construction Process Model

Through personal experience, extensive literature reviews and various semi-structured interviews with industry practitioners, this research has established the sub-processes and functional activities, and their hierarchy. The overall activity process functions included in the generic construction process model are shown in the function node tree diagram in Figure 4.6. At the highest level of the hierarchy is the process of 'construct house facility' (Node: A-0), which is then separated into the following five sub-process functions.

- Organise Site Resources [Node A1]
- Groundwork Excavate [Node A2]
- Construct Substructure [Node A3]
- Construct Superstructure [Node A4]
- Inspect and Approve Work [Node A5]

Due to the complexity and absolute quantity of data generated from the process activities, a full description of all the process activities is beyond the scope of this thesis. Therefore, for simplicity and readability reasons, only the diagrams of the process functions and sub-
functions used to facilitate a prototype information modelling system (described in chapter 5) are presented as an example, and also highlighted (1,2,3,4) in Figure 4.6.

Figure 4.6: Construct house facility hierarchy node tree

4.4.2.1 The IDEF0 Notation

The utilisation of a top-down modelling approach was adopted, in order to determine the building levels (i.e. the amount of detail e.g. resources, environment constraints and/or external conditions, etc.) necessary for the development of the modelling system. The approach exploited the IDEF0 modelling methodology (described in Chapter 3), which provided the means for modelling the functions (activities, processes, operations) and the functional associations (relationship flows between the inherent activities), using a series of hierarchical process diagrams containing boxes, arrows, interconnections and associated relationships. The boxes represent the functions of a subject; in the work described here, ‘construct house facility’. These functions (described using verbs or verb phrases) are
decomposed into more detailed diagrams, until the subject is described at a level necessary to support the goals of a project. The top-level diagram in the model provides the most general description of the subject and is followed by a series of child diagrams providing more detail about the subject. Arrows describe the things (i.e. data and objects) that constitute the system using nouns or noun phrases (IDEF, 1993).

4.4.2.2 The Contents of the Generic Construction Process Model

Table 4.4 presents a node index (table of contents) for the GCPM. Excluding the top-level diagram, four diagrams are specified (highlighted) in the node index: the ‘context’ (node: A0) diagram [construct building facility]; the ‘construct sub-structure’ (node: A3) diagram; the ‘construct foundation’ (node: A33) diagram; and the ‘construct superstructure’ (node: A4) diagram. A fifth diagram listed in the node index is the single box top-level context (node: A-0) diagram, which is presented in Figure 4.7.

Table 4.4: Diagram Nodes and List of Activities for the Construction Process Model

<table>
<thead>
<tr>
<th>Diagram Node Reference</th>
<th>Description of Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node: A-0</td>
<td>Construct House Facility (Top-Level)</td>
</tr>
<tr>
<td>Node: A0</td>
<td>Construct House Facility (Context Diagram)</td>
</tr>
<tr>
<td>Node: A1</td>
<td>Organise Site Resources</td>
</tr>
<tr>
<td>Node: A2</td>
<td>Groundwork Excavation</td>
</tr>
<tr>
<td>Node: A3</td>
<td>Construct Substructure</td>
</tr>
<tr>
<td>Node: A31</td>
<td>Plan or Work Schedule</td>
</tr>
<tr>
<td>Node: A32</td>
<td>Distribute the Resources</td>
</tr>
<tr>
<td>Node: A33</td>
<td>Construct Foundation</td>
</tr>
<tr>
<td>Node: A331</td>
<td>Load out Foundation</td>
</tr>
<tr>
<td>Node: A332</td>
<td>Setout and Concrete Foundation</td>
</tr>
<tr>
<td>Node: A333</td>
<td>Setout and Construct footing</td>
</tr>
<tr>
<td>Node: A334</td>
<td>Setout and Construct Concrete Slab</td>
</tr>
<tr>
<td>Node: A335</td>
<td>Clean Up and Endorse Work</td>
</tr>
<tr>
<td>Node: A34</td>
<td>Set-Up Building Service Systems</td>
</tr>
<tr>
<td>Node: A35</td>
<td>Set-Up and Erect Scaffold</td>
</tr>
<tr>
<td>Node: A4</td>
<td>Construct Superstructure</td>
</tr>
<tr>
<td>Node: A41</td>
<td>Setout and Construct Drainage</td>
</tr>
<tr>
<td>Node: A42</td>
<td>Setout and Construct Walls</td>
</tr>
<tr>
<td>Node: A43</td>
<td>Setout and Construct Roof</td>
</tr>
<tr>
<td>Node: A44</td>
<td>Setout and Construct Floors</td>
</tr>
<tr>
<td>Node: A45</td>
<td>Future and Fittings</td>
</tr>
<tr>
<td>Node: A5</td>
<td>Inspect and Approve Work</td>
</tr>
</tbody>
</table>
4.4.2.3 The Top Level Process Activity Diagram

Figure 4.7 illustrates the top-level (Node: A-0) diagram for the GCPM, and is used here to represent the on-site construction processes (building delivery process). The diagram contains a single process activity function box, ‘construct house facility’ which transforms the input, ‘resources’, into the finished product output, ‘constructed house’, using the mechanisms, ‘construction team’, ‘mechanical plant’, ‘tools’, and ‘equipment’. The controls, ‘building regulations and standards’, and ‘environmental constraints’ provide the context for the ‘construct house facility’ activity. The top-level diagram defines the overall scope and external elements and connections of the total construction process model, and is decomposed into 5 main stages (sub-process activities), illustrated in Figure 4.8 and listed in Table 4.4.

![TOP LEVEL [IDEFO] DIAGRAM](image)

**Figure 4.7: The top-level process activity diagram for the GCPM**
4.4.2.4 The Context Diagram: Construct House Facility

The second level context (node A0) diagram, illustrates the five main stages of the GCPM. It portrays a broader view of the top-level diagram where the ICOMs are described in greater detail from the construction manager’s point of view. The top-level (context diagram) of the process model for the provision of a building facility is broken down into five main processes, as illustrated in Figure 4.8. Each activity function is referenced with a unique identification number, prefixed with the letter A (for activity), which enables the location of the function within the process hierarchy (Figure 4.6) to be identified. The procedure of undertaking the processes (operation or task) is actually that of determining and establishing the activity described. These functions are described as follows:

![Figure 4.8: Node A0 illustrating the five main activity stages of the GCPM](image-url)
Organise Site Resources: This function (node: A1) is broken down into four sub-functions to enable the mobilisation of the resources (plant, equipment, services, materials, etc) needed to construct the building facility. In addition, the work area is established and organised to enable all building activities to proceed without any obstruction or hindrance.

Groundwork Excavation: This function (node: A2) is broken down into four sub-functions to organise and establish the workspace for setting out, site levels, excavation, and removal activities from site and foundation areas.

Construct Substructure: This function (node: A3) is further broken down in to five sub-function, as illustrated in Figure 4.9, and is one of the main activities where other functions exist to support this process of converting resources into completed elements of the substructure.

Construct Superstructure: This function (node: A4) is broken down in to five sub-functions, as illustrated in Figure 4.10, and is conceived to be the main activity function, whereby all other functions exist to support this process of converting resources into completed elements of the superstructure.

Inspect and Approve Work: This function (node: A5) is broken down into four sub-functions, whereby the completed work is checked to assure the dimensions and quality of the products, elements and components are satisfactory and that the contract requirements were fulfilled. Payment and documentation is finalised on completion of the project. The building is finally turned over to the client (or client representative e.g. facility manager).

4.4.2 5 Third Level Diagrams

The third level diagrams, illustrated in Figure 4.9 and Figure 4.10 are a decomposition of boxes A3 and A4 in the context (node: A0) diagram, and include the sub-functions (listed in Table 4.4) needed to perform the main stages, as illustrated in the second level context diagram (Figure 4.8). These functions (described using verbs) are decomposed into more detailed diagrams (four level), until the ‘construct house facility’ is described (i.e. provide a broader range of information, so as to identify the requirements in accordance with the design constraints.
Figure 4.9: Node: A3 illustrating a third level diagram 'Construct Substructure'

Figure 4.10: Showing Node A4 third level diagram 'Construct Superstructure'
4.4.2.6 Four Level Diagrams

The forth-level ‘construct foundation’ diagram (Figure 4.11) is a decomposition of box A33 in the ‘construct substructure’ (node: A3) diagram, and includes the sub-functions (listed in Table 4.4) needed to perform all the construct foundation activities. The ‘construct foundation’ diagram provides an even-more detailed description of information, so as to identify the specific requirements (i.e. translate resources such as building materials, components, equipment, services, etc.), in accordance with the design constraints (e.g. specifications, standards, regulations, etc.), into a finished product (the completed building facility). The arrows (ICOMs) describe the data and objects that constitute the system (using noun phases), and contain hyperlinks to graphical data and documentation, and other relevant information files. The combination of diagrams, functions, sub-functions, arrows, and hyperlinks to graphics and document files provide a consistent representation of the information necessary to support the ‘construct house facility’ (i.e. the goal of the project).

Figure 4.11: Fourth level diagram-illustrating Node: A33 ‘Construct Foundation’
4.4.2.7 Information, ICOM Attribute and Hyperlink

The activity diagrams, which make up the contents of the generic construction process model has been partitioned and structured in such a manner, so as to contain as much building information as possible. Each activity box and/or information component in the process model has attributes and hyperlinks (or link connections) attached to other construction information, such as product and process models, scheduling charts, building documentation, Web files, graphics, images and pictures. The structure of a process activity information component (PAIC), illustrated in Figure 4.12, undertakes a similar approach to the process based information architecture (PBIA) adopted by the computer information construction (CIC) research project (Sanvido, et al. 1995).

![Figure 4.12: The framework of the Process Activity Information Component](image-url)
The Figure shows the attributes of 'construction foundation' function (node: A33) as being the function box, description, inputs, outputs, controls/constraints and mechanisms (arrow links) of the process. Each process activity information component comprise of a single individual function box (similar to a IDEF0 top level diagram) showing the ICOM arrow links required to perform or transform the activity into output. The process information table (Figure 4.12) lists the many hyperlinks associated with the decomposed function 'construct foundation', which includes links to information sources such as, scheduling data, information models, Offices Docs/files, Web pictures, etc.), and specifically relates to those attributes connected to each individual function box. The information concerning the process activity input, output, controls and mechanisms is more appropriately classified and represented in other information models and/or document files. Therefore, the links are used so that this information can be accesses through the GCPM or the integrated product and process information modelling system’s prototype (HIPPY) user interface (described in Chapter 5). The GCPM corresponds to, and is consistent with the traditional mode of operations on a general building site in the United Kingdom. This is because the information sources material (CI/SfB, BSAB, RIBA, SMM7) is a generally accepted guideline for the processes of on-site construction operations (Bindslev, 1995). Thus, the GCPM can be considered as a systematic and generic process reference model, which provides a template, allowing end-users to enter their own constraints, parameters, etc.

4.4.3 Discussions and Limitations

The GCPM shows a method for generically classifying (or categorising) the process information required during the on-site construction stage of SMsBP. It is necessary to keep the term generic with the intention that projects can be referred to by using a similar numbering system. However, since it is generic; some of the shortcomings of the process model include the following:

- It is sometimes difficult to determine the cut-off points between the different processes (e.g. when does construct substructure finish and construct superstructure begin).

- The process model lacks a structure for representing the importance and complexity of performing the different processes. These attributes could be considered for further research and added to the model at a later date; however, this information tends to be quite subjective. If these attributes could be added to the process model, a formal measuring system should be developed to measure these attributes, and much of this
information might be duplicated or made redundant, because it could be facilitated in the product model, or the 'feedback system', which has not been described here because it is associated with further research discussed in chapter 7.

- The model has been specifically developed to focus on the on-site construction phase and only shows the upper levels of the process model. In order to implement the complete model, the model should be developed to a more detailed level. This could potentially make the model very large and confusing.

- In its present form (i.e., because the process model was developed using the IDEF0 methodology), the model exhibits serious difficulties in addressing some of the very common, typical business system analysis problems, such as the identification of objects (i.e., lack of a clear distinction between flow of material objects and flow of information). As a consequence, it is incapable of modelling process information flows, due to its lack of time dependency input. These limitations impose a strong restriction on the use of this modelling technique for the development of information systems. The need to integrate other methods capable of capturing the sequences of the process activities and the resources used to perform them is evident.

To overcome some of the problems identified above, scheduling the process activities detailed within the GCPM, by means of attaching hyperlinks to scheduling charts were used. This method enabled the construction processes to be decomposed into operational units (activities or tasks) and the resources used for performing them. As previously mentioned (Chapter 3), activities represent the process that occurs over a certain time interval, identified by descriptive function labels (e.g., construct substructure, construct foundation, construct walls, etc.); duration, and task attributes, such as start and end dates, allowing the 'duration' of the activity to be calculated. Each process activity information component frame (Fig. 4.12) contains links or connections to project scheduling information (Gantt charts), which sequentially contain additional links to other information sources. Figure 4.13 illustrates how the top-level 'construct house facility' (node: A0) diagram is combined (integrated) with the sequence of activities (i.e., scheduling chart).

The GCPM is one of two generic information models that provide the foundation for the integrated information architecture; that is the hierarchy structure for the integrated conceptual model. The second key information model is the generic building product model (GBPM), which is described in the following section.
Chapter 4

4.5 THE GENERIC BUILDING PRODUCT MODEL

4.5.1 Development of the Generic Building Product Model

The development of the generic building product model also involved an iterative process, to ensure its relevance to, and practical application in, the construction industry. This section presents the generic building product model (GBPM) of a house-building as part of the information architecture of the integrated conceptual model. The model is based on the unified modelling language (UML), an object-oriented modelling methodology (described in chapter 3) that enables individuals to model the conceptual entities with a powerful modelling concept: the object.

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>End</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Organise Site</td>
<td>12/11/01</td>
<td>20/11/01</td>
<td>1w 1d</td>
</tr>
<tr>
<td>2</td>
<td>Excavate Site</td>
<td>21/11/01</td>
<td>28/11/01</td>
<td>1w 1d</td>
</tr>
<tr>
<td>3</td>
<td>Construct Substructure</td>
<td>09/11/01</td>
<td>13/12/01</td>
<td>2w 1d</td>
</tr>
<tr>
<td>4</td>
<td>Construct Superstructure</td>
<td>11/12/01</td>
<td>08/02/02</td>
<td>8w 4d</td>
</tr>
<tr>
<td>5</td>
<td>Inspect and Approve</td>
<td>14/12/01</td>
<td>26/02/02</td>
<td>10w 3d</td>
</tr>
</tbody>
</table>

Figure 4.13: Capturing and integrating the information for the GCPM
The context of the generic building product model is the life cycle of on-site construction products (and to some degree processes) essential for constructing ordinary buildings. The scope of the model is frequent information about such product elements, components, etc., and buildings handled by the professional participants (e.g. construction managers, contractors, users, etc) and external systems (referred to as actors in UML), during the on-site construction stage. After identifying the context and scope (i.e. the information captured and modelled in the generic construction process model), a bottom-up modelling approach was applied utilising the features offered by UML. The approach adopted is to breakdown a house-building’s elements and its processes into objects and classes of objects with information attributes. As well as, completing the basic structure of the information architecture, with the aim of developing an integrated conceptual model to support the management of information during the on-site construction stage of SMsBP. The purpose is to demonstrate how the UML standard can be used to develop a generic building product model of the on-site construction stage, in addition to providing an effective way of representing the integrated conceptual model of the software system design (i.e. the hyper-integrated information modelling system (HIPPY) prototype).

4.5.2 Modelling the System Requirements

System modelling aims to define levels of system details in terms of a set of models. Two conditions were applied in capturing the requirements of system modelling:

- Firstly, the system must fully embrace and comply with the industry standard object models and architecture to enable interoperability with other information modelling systems and a wide variety of other applications software systems.
- Secondly, the system must employ industry best-practice modelling techniques in a proposed development process to facilitate the management of system complexity. In order to complete the information architecture (i.e. the generic building product model), the system-modelling phase employs UML since it has emerged as the notational standard for object-oriented modelling, and because it is relatively comprehensive.

UML is a standard graphical modelling language for modelling multiple perspectives of information systems (i.e. a language used to specify, visualise and document the artefacts of an object-oriented system under development). UML defines a set of basic diagrams that provide the multiple perspectives (static and dynamic) of the system under analysis or
development. Standard modelling technique may standardise and facilitate the development process through common concepts, notations and supporting tools and thus increase compatibility with other software systems. However, the utilisation and understanding of the UML notation can be very complex, and the discussion of specific processes to support UML has not been defined as an objective of this thesis. Therefore, in the work described in the system-modelling phase, where the definition of product modelling is central, the important aspects of UML are Use Case and Dynamic Modelling, and Class or Object Modelling. A more detailed representation of UML and how the models are used concurrently can be found in (Rumbaugh, et al. 1999; Kimmance, 2000b).

4.5.3 Description of the UML Diagrams

One of the problems encountered with the early representation of the generic building product model with UML was the determination of the types of UML diagrams to be developed and their sequences, since the subject to be modelled is different enough from the common usage of the language, notably the nature of the system (subject) to be described. Compared to software development, the specificity of the use, which is made of the language, lies in the way of defining the specifications and requirements of the system. The specifications of a building should be known from the beginning, since the client representative defines them in the feasibility and design stages of a project. However, to capture and identify the requirements, involved the modelling of three distinctive diagrams: Use Cases, Sequences and Class diagrams. In order to understand the underlying reality of the system, it is important to recognise some fundamental characteristic of these diagrams.

4.5.3.1 Modelling Use Case Diagrams

The Use Case approach bears some similarities with the technique used to develop the IDEF0 activity diagrams, in the sense that they both aim at providing a basis for the underlying, definition and capture of the requirements of the information modelling system to be developed (i.e. the GBPM). The construction process model treats the construction stage as a series of processes. Each process is broken down into activities and sub-activities. These activities provided a good basis for identifying Use Cases.
In the work described in this thesis, a use case is a sequence of actions that describe a way in which a real-world scenario (or actor) interacts with a system; that is, what a system may or may not be able to handle. Use Cases represent the high-level functionality of a system in development, describing 'what the system should do', as well as being a simplified, abstract-generalisation that captures the intentions of a user in a technological, operational and independent manner (Jacobson, et al. 1992). Although, individual Use Cases contain limited information, they do however provide a complete description when linked to its Use Case scenarios (e.g. a detailed description of events usually represented in a Sequence diagram), which ideally defines what the Use Case should achieve throughout its functionality. The Use Case scenarios (descriptions) are composed of the flow of events, such as details about user actions, software actions and reactions, constraints, requirements for graphical interface, relationships with other Use Cases, etc. Use Case descriptions also provide valuable information when specifying the properties (attributes and methods) of the classes needed to perform the Use Case. Three elements can be identified in a Use Case. These include:

- **Actors**: which represents the people, organisations, or external systems (stimulus) that interact with the system, and were identified from the 'resources' or 'mechanisms', indicated in the proposed construction process model.

- **Actions**: which represents a capability requested of a system, and were identified from the activity function and operations indicated in the lower-levels of the GCPM.

- **Subject**: which represents the items acted upon (response) by an action requested of the system.

When the Use Cases were identified, a UML Use Case diagram was modelled to represent the relationships between actors and their actions. A Use Case diagram uncovers the classes of an application. It takes a step back and initially identifies who the actors are in the system. Once it has identified the actors, the Use Case diagram then captures actual Use Cases, the high-level 'what's' that the future system (HIPPY) is responsible for implementing.
Figure 4.14 illustrates the top-level Use Case diagram, depicting the main Use Cases and actors connected with the on-site construct stage. The diagram also providing links to detailed descriptions of the elements involved with each Use Case. The description attempts to clarify the interactions between the actors and the system through describing what, and how, the actor is doing by using the system, instead of what the system is doing. In contrast to IDEF0 representations, the sequence of activities and data are not the focus of attention, but rather the emphasis is on the main functions that the system must perform.
4.5.3 Dynamic Modelling through Sequence Diagrams

Once the actors have been identified, the activities of the actors involved with the project were defined through dynamic modelling. Dynamic modelling addresses the dynamic behaviour of objects via the different events and associated state changes that can happen to an object during different time periods. Sequence diagrams provide a dynamic high-level representation of the sequencing of the activities, through the description of a 'working scenario' of the actors involved with the system. They capture and represent interactions required between objects, through their methods, emphasising the time ordering of messages. These diagrams mainly represent the behavioural aspects of objects, showing what methods (functions) are required to satisfy a specific Use Case. A scenario is a sequence of particular events that occur during the execution of a system. Scenario analysis is performed by sequentially arranging the events shown in the Use Case descriptions, resulting in sequence diagrams of objects interactions arranged sequentially. Each Use Case (or Sub-Use Case) has at least one associated Sequence diagram and a scenario description, as illustrated in Figure 4.15 and Table 4.5.

![Figure 4.15: Sequence diagram for the 'Construct Foundation' Use Case](image)
These depicts how the Use Cases would be performed by the objects (or classes of objects), what messages are passed between them (at a later stage, these messages will be used to determine which objects call which methods of which of the other objects, and in what order). This also includes a detailed description of how the users external to the system take part in the process.

**Table 4.5: A Description of the ‘Concrete Foundation’ Use Case**

<table>
<thead>
<tr>
<th>Name:</th>
<th>Concrete Foundation Use Case No 12:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions:</td>
<td>The Labourer carries out Concrete works and drivers of the concrete lorries pour the ready mixed concrete to foundations</td>
</tr>
<tr>
<td>Objective in Context:</td>
<td>For ready mixed concrete to be poured in the building foundations, vibrated and levelled by the Building Labourer, and inspected by Site Manager/Engineer</td>
</tr>
<tr>
<td>Primary Actors:</td>
<td>Skilled Labourer and Building Material Supplier</td>
</tr>
<tr>
<td>Secondary Actors:</td>
<td>Site Manager/Engineer</td>
</tr>
<tr>
<td>Use Case begins:</td>
<td>When concrete is poured into the building foundations</td>
</tr>
<tr>
<td>This Use Case does:</td>
<td>The drivers of the concrete lorries pour the ready mixed concrete into the building foundation and the Building Labourer vibrates and tamps concrete</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message flow between Actors and Use Case:</th>
<th>Actors Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Site Manager receives ready mixed concrete from Building Material Supplier</td>
<td>Logs amounts and times</td>
<td></td>
</tr>
<tr>
<td>The concrete is poured into the building foundation</td>
<td>Records operation details</td>
<td></td>
</tr>
<tr>
<td>The Building Labourer vibrates and tamps concrete in trenches</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The building foundation is inspected by the correct authority</td>
<td>The system records the decisions made</td>
<td></td>
</tr>
</tbody>
</table>

| This Use Case Ends: | When the building foundation has been cast in concrete, levelled, inspected and passed |

It is important to recognise that each Sequence diagram can be relatively complex in nature, and can also run to great lengths. Therefore, it is appropriate to mentioning that over 18 Sequence diagrams have been produced within the first iteration (Figure 4.14). However, to aid readability and comprehension of the approach used to develop the product building information model, Figure 4.15 depicts how the example Use Case ‘construct foundation’ (illustrated in Figure 4.14), is related in order to represent the
dynamic aspects of the information system. Also, to facilitate the way in which information (the flow of events) is recorded and described, for example the message sent from the 'builder' (actor) to the 'foundation' (Use Case) stating 'concrete to foundation' (Fig. 4.15.). Table 4.5 presents a description of the 'concrete foundation' Use Case, which is one of the activities identified in the 'construct foundation' (node: A33) diagram of the construction process model. The description includes the purpose of the Use Case, actors involved, and the detailed flow of events that follow the initiation of a Use Case, which can then be modelled or represented through the Use Case and/or Sequence scenarios. This is to give the reader an appreciation of how Use Cases, scenario descriptions and Sequence diagrams fit into the overall methodology employed within this thesis.

In the initial version of the Sequence diagram, illustrated in Figure 4.15, the original messages sent between objects and actors were written in plain English. However, this version of the diagram shows these same messages, but in a syntax more suited to a technical implementation. Each of these messages are fully documented and stored within Visio 2002, the CASE tool used to produce all the diagrams presented in this thesis.

4.5.3.3 Class Modelling

Class modelling aims to identify the internal structure of objects; this is, the classes of objects involved with the information structure of the system, each of which can be described by its name, attributes and methods, and its relationships with other objects. It produces class diagrams that define the static, structural, and data aspects of the information system architecture in terms of classes and relationships that correspond to the management of information during the on-site construction stage of SMsBP. This process facilitates the understanding of real world entities and thus provides a practical basis for system implementation (described in chapter 5).

A class diagram consists of classes, links and associations. A class is a set of objects that share common attributes, operations, semantic structure and behaviour. An object is a concept, abstract, or thing in a certain application domain (i.e. an instance of a certain class). Meanwhile, a link is a physical or conceptual connection between instances of classes (association). Finally, an association describes a group of links that share a common structure and semantics. Two of the most commonly used associations are generalisation and aggregation.
Aggregation is the 'part_of' relationship in which lower-levels classes are associated as a higher-level aggregate class. Meanwhile, generalisation is the 'is_a' relationship in which lower-level subclasses are a specialisation of their superclass, as shown in Figure 4.16.

**Figure 4.16: Basic structure of a Class diagram**

Objects of the same class may be characterised by the same set of alternative views, attributes, and operations, although their attribute values may vary with each instance. Therefore, by grouping objects that have the same set of object elements into classes of objects, can improve the abstraction and reusability of an object/class model. For example, in the build house application (system) the site manager will at some time initiate the start of an activity or operation (i.e. give a instruction to perform a building task), the site management instructions are instances of a site management instruction class, which could have the following elements:

- **Attribute:** Name_of or Instruction_Name, Type of Instruction, Quantity, Time or Date, etc.;
- **Operation:** Get_Name, Change or Modify Instruction_Name, Create Instruction_File, Delete_File, etc.

Each instruction has its own name, type, time or date of instruction. Each instruction will also be involved in other cases, which are instances of case classes. The operation of these other classes could possibly change the attribute values and create and delete an instruction. This kind of approach (class diagrams) supports the definition of the data modelling of the integrated modelling system HIPPY. A more details view on the UML notation can be found in product and process modelling approaches (Kimmance, 2000b), which was one of the reports generated from the research work described in this thesis.
In the A/E/C domains, EXPRESS has been recognised as the standard language for supporting such representations. However, with the growing use of OO techniques, and more recently UML and XML, conversions between these representations have become common (Ghodous and Vandorpe, 1998; Burkett, 2001). Also, considering that the final system will be developed using a hybrid approach integrating both IDEF0 and OO methods, the use of an OO standard become significantly advantageous for developing the integrated conceptual model. Therefore, UML class diagrams can represent the product, and in some circumstances the process information levels of HIPPY.

4.5.4 Description of the Generic Building Product Model

The basic idea behind this approach was to breakdown a house-building and its processes into objects with information attributes and operations. After having identified the context and scope (the basic functionality) of the product model the next stage was to identify the construction entities (classes of objects), with the aim of developing the generic building product model. In order to store these attributes and operations in a proper location and avoid redundancy, a hierarchical structure was used for convenience to organise the information. Classes were stored according to their primary functions, although information regarding other functions may also be stored in the attributes. The purpose of this product class breakdown hierarchy structure is to provide an organisation and categorisation of information describing product elements and components in terms of the attribute categories, operations and functions.

4.5.4.1 Information Structure of the Generic Building Product Model

Figure 4.17 divides a house-building (facility) into several levels, and is used here to illustrate how the classes of objects are decomposed into product element sub-levels. The Figure shows the functional class hierarchy of a house-building (i.e. breakdown of product classes) into its product parts and sub-parts; showing the top-level site facility, top-level product type, sub-product level, building element types, sub-element level, building component type, and sub-component levels. Classes at different levels have various degrees of detail (specification levels) in this hierarchy of information. Entities within each level are broken down into unique, detailed categories of information at the next lower level. For example, the top-levels focus on a more generic representation level, which
corresponds to the products and building elements of the sub-products (e.g. structure, foundation, wall, roof, etc.). The lower levels focus on a specific identifier level, whereby the positions \( (x, y, z) \), lengths and sizes of the building elements and components are identified and detailed. However, entities at the component and sub-component level may share information categories at the next lower levels, because of the sheer quantity of information; thus, needing to be duplicated.

*Functional Hierarchy*

**Figure 4.17: Breakdown of the building product data into classes and sub-classes**
4.5.4.2 Basic Entities of the Generic Building Product Model

Once the classes were established, the next step was to define the basic 'core' structure of the generic building product model. The requirements that were identified in section 4.3 and the information models (Use Cases, Sequence, etc.) are included in the information structure. The basic entities (most significant construction class entities) in the definition of a GBPM for on-site construction and their relationships are described using the UML Class notation (Figure 4.18). The main entities of the core part of the generic building product model have been described in Section 4.2.4. These include: resources, processes, products and actors, and are highlighted within the dotted line.

![Diagram of the core of the generic building product model using UML class notation](image)

**Figure 4.18: The ‘core’ of the generic building product model using UML class notation**
Other Class entities of the GBPIM include:

- **Method Approach:** Represents the way in which the decisions are made on the use and organisation of the resources and processes (i.e. describes how an activity is performed), and are controlled by the specifications.

- **Surrounding:** This includes the parts connected to the building but not belonging to it (e.g. municipal networks or roads, etc.).

- **Building:** The main body of the building includes the structure that is decomposed into other sub-systems such as: substructure (foundation) and superstructure (walls, roof, etc) and can be grouped under the space class. All systems must belong to one building.

- **Site Area:** This describes the site areas for storage, parking, building area, etc., which contains the building, and is associated with organise site resources, mobilise site, etc.

- **Functions:** These include the spaces, which are used by the actors to carry out an activity such as plaster wall, construct floor, etc.

Figure 4.18 illustrates a UML Class diagram representing a simplified version of the core generic building product model used in the integrated information system's (HIPPY) structure. The diagram represents object classes (entities), which contain hyperlinks to other diagrams and document files connected by different types of relations: normal unidirectional relations (with cardinalities, single roles and open arrows), bi-directional relations (with cardinalities and roles), specialisation (with closed arrows) and aggregation (with diamond). Each Class contains attribute properties (above) and method operations (below). The attributes and operations can be sorted into different classes depending on their information content. The operations can relate to the actions or commands, which are performed in the user interface (e.g. Find Building Component, Print Report, Add, Delete, GoTo Process Information, etc.). It must be pointed out that for readability reasons only some (main core classes are modelled very schematically) examples are given, so as to keep the description simple.

The core of the GBPM (Fig. 4.18) proposes that a facility is a building or consists of a set of buildings, a set of processes, and a set of method approaches, which are controlled by the processes, and is represented by the product model. To construct the building, requires the use of temporary services, which include supply installation products such as gas, electric, water, etc., which also has a structure divided into sub-sections. These sub-sections have building spaces (voids) enclosed by walls (façade) and roofs, and internally
divided by walls and floors (space separators). Each building house functions (the trade (actor) work functions), which need a certain volume and floor area that requires construction aids (materials, plant, equipment, tools, etc.) for providing the specified working levels, light, power, etc., together with the window openings and doors, and building components needed to perform the construction activities (processes), which are controlled by specifications, regulations and standards (see Figure 4.19). Thus, the product model basically describes the facility (house-building) through the use of building Products (materials, elements, components, etc), which are used by the Actors to perform the Processes (activities, functions, etc.) that require Resources (construction aids: plant, equipment, etc.), which are controlled by approaches and specifications.

Figure 4.17 and 4.18 can also be conceived as an organisation of system functions from a system perspective. Therefore, each level (functional class block) in the hierarchy is potentially a ‘user class’ that defines the user interface for performing its corresponding function. The functional class hierarchy also reveals the associations between levels of user classes. By referring to the proposed system information architecture and configuration, the user classes identified from both the functional hierarchy and the generic building product model can be refined to fit the system organisation better. The generic building product model is not very detailed and excludes many object classes that are for example, included in the IAI-IFC model. The reason for this is that IAI-IFC mainly focuses on design and this model is about construction. Also, for clarity reasons, many attributes and operations have been hidden or deleted. However, a more detailed representation of the product classes (or classes of objects) associated with the generic building product model is illustrated in Figure 4.19. The information has been collected from a combination of IDEFO models, Use Cases and Sequence diagrams, Scheduling Charts and Microsoft Office document files capturing the information of the classes, and linked to each individual class via hyperlinks, as illustrated in Figure 4.18 and 4.19.
Figure 4.19: Elaborated description of product classes associated with the GBPM
4.6 THE INTEGRATED CONCEPTUAL MODEL

Conceptual modelling is a means of conceptualising some well-defined part or entities of the real world. A conceptual model should show the structure of information in these "mini-worlds". Conceptual models provide formal definitions of the basic entities and relationships required to fully represent information about the domain in question. The term "conceptual schema" is often used for such a model. The conceptual model may be presented as a single diagram covering the area of interest, or as a diagram for each subject area and a single, high-level overview diagram. This will depend on the extent of the area being modelled. A repository containing detailed definitions of the diagram elements will normally support the diagrams. There are two approaches to developing conceptual model diagrams: top-down and bottom-up approaches. In the work described in this section, both the top-down and bottom-up approaches were used in parallel to develop the 'integrated conceptual model' of on-site construction, with house-building as a test bed.

Following a detailed evaluation of the system requirements and information captured during the analysis stage, the conceptual product and process information models were developed utilising a CASE tools (Visio 2002) that supported all the modelling notations needed to develop the information models, and provided the graphical environment for the research work described in this thesis. Detailed mapping and modelling of product and process information on to several levels were then carried out in order to develop an integrated product and process conceptual model of the real-world problem domain, as illustrated in Figure 4.20. The integrated conceptual model exhibits a hierarchy format of the construction process information (e.g. activities, operations, materials, equipment, standards, etc.), described in the GCPM hierarchy (Fig. 4.6), and the building product data (e.g. classes, elements, components, resources, etc.), described in the GBPM hierarchy (Fig.4.17). The structure of both modelling methods has been developed, so as to illustrate the decomposed relationship levels, and to identify the 'specific' type of product and process information being integrated during the on-site construction stage of SMsBP.

The integrated conceptual model represents the building objects, functions, and relationships handled throughout construction on-site activities and is used here to demonstrate how the information is decomposed and grouped to form the classes and subclasses, which is then broken-down, organised and placed/stored into a object-based relational database (i.e. the components of a relational database containing the fields and objects of the product and process models (integrated conceptual model) defined by the
prototype system's scope). Therefore, the integrated conceptual model represents the information models; that is, the product and process information, which have been decomposed in a hierarchical manner, so as to facilitate the breakdown of the modelling data, and amalgamated with the relational database (Access 2000), in order to develop a prototype system for the implementation of the integrated conceptual model.

4.6.1 Discussion of the Modelling Methods

Although the modelling approaches used to develop the information models performed slightly differently, they do however bear some similarities with each other, given that they both aim at providing a foundation for the understanding definition, and capture of the requirements needed to model the identified problem domains (on-site construction). The method adopted, followed a combination of top-down and bottom-up modelling approaches. While UML defines a notation that is designed to represent the physical world entities (although no process is currently defined), IDEFØ attempts to represent the physical world within its notation.

The top-down hierarchy approach was performed starting with the analysis of the functional requirements for developing process activity models (e.g. operations, tasks, resources, environmental conditions, etc.), in addition to the information and material flows between the inherent activities, and the definition of the basic functions that the information models have to support. This was achieved by modelling the activities with IDEFØ and the operations (process tasks) through the use of Scheduling Gantt charts, which were broken-down into operational units. These methods proved to be very effective in modelling the information at the process-required levels (building delivery process stages), and the combined use overcame their individual limitations. In contract, UML provides a bottom-up approach for analysing various ways of using a system by specifying the proposed Use Cases, and detailing them through a sequence description of the interactions (messages) between the artefacts. It provided the necessary details to the general structure, detailing Classes, defining their attributes and operations, and building the relationships with other Classes.
Figure 4.20: Integrated conceptual model for on-site construction
However, neither IDEFØ nor UML offers the scope for providing time dependency process flows offered by Scheduling methods. As a consequence, the shortcomings of UML are indicators that reflect the strengths of IDEFØ and Scheduling methods. Both IDEFØ and UML methods are complementary to each other and are used in concurrence, as illustrated in Figure 4.21. The Figure shows how IDEFØ is integrated (fit together) with UML Class, Use Cases and Sequence diagrams. The IDEFØ activity models and Scheduling charts present the scope; context, needs and construction environment in which the proposed integrated modelling system is used. Sequence diagrams provide a powerful representation of the sequencing of the different activities, through the description of working scenario (Use Cases) of the actors involved, thus enabling a detection of possible strategic crossings that could be improved using concurrent engineering features.

Even though the UML methodology can be used as a standalone approach for software design; it has however, been found that the UML approach needs extra support in the requirements capture stage, mainly for the definition of the structure of the information models. Although there is no direct interaction between the information models in IDEFØ and UML, the scheduling and IDEFØ modelled information can be used as a guide in the definition of the attributes and behaviour of the classes.

4.7 SUMMARY

In this chapter, a compilation of generic product and process information models (conceptual models), which focuses on the information needs with respect to on-site construction, with house-building as a test bed, has been presented. It has provided a conceptual description of the building delivery process (i.e. described the information requirements: purpose, viewpoint, scope, etc.), and the product and process information modelling systems functions and requirements to enable an integrated conceptual product and process model that describe on-site construction information to be developed. The chapter has discussed the importance of integrating construction information, and presented a method for modelling ill defined, informal real world and end user requirements, and transforming these into clear, formal and consistent representations, through the use of information models. The features and characteristics of the generic information models have also been described along with the requirements for creating the integrated conceptual model specific to small to medium sized building projects.
Chapter 4

Figure 4.21: Integration of IDEF0, scheduling and UML modelling approaches

NOTE
All relationships and boxes contain hyperlinks to other sources of information
Furthermore, a detailed architectural structure and logical view of the integrated information-modelling environment for describing on-site construction information has been presented. The structure of the integrated conceptual model reflects the framework of the product and process hierarchy breakdown structures, along with additional necessary information from a construction managers (site agent) point of view.

The approach used for developing the information structures followed a combination of top-down and bottom-up modelling approaches. These approaches exploited both the IDEFØ methodology and the unified object-oriented modelling language. The shortcomings of IDEFØ are indicators that reflect the strengths of UML. While IDEFØ models are inclined to have some restrictive quantities, such as the type of arrows permitted and time dependencies, UML models (although no process is currently defined) tend to have fewer limitations. Furthermore, the hierarchy that is essential in IDEFØ models is not required in UML diagrams. Therefore, the combined use of these methods is a better way of communicating with collaborating companies, construction managers, and professionals within client organisations. In addition, a CASE tool (Visio) has been utilised as a development tool, which facilitates additional information to be incorporated into the information models.

Finally, the integrated conceptual model illustrating how the integration of product and process information is presented; in addition to, how the relationships and integration of IDEFØ and scheduling charts is used in concurrence with UML Classes, Use Cases, and Sequence diagrams is achieved.

The next chapter presents the means of integrating product and process information by discussing the development of a prototype system (HIPPY) for the implementation of the conceptual information models described in this chapter. The chapter provides evidence of the effectiveness of HIPPY for managing the processing of product and process information essential to the construction manager during the on-site construction stage of a small-to-medium-sized building project (using house-building as an example), in order to establish automated and controlled information flow.
CHAPTER 5
IMPLEMENTATION OF THE PROTOTYPE SYSTEM

5.1 INTRODUCTION

This chapter addresses the means of communicating information by discussing the implementation of a prototype system (HIPPY) for integrating and managing the processing of both product and process information during the on-site construction stage of a construction project, using a house as an example. It gives an overall description of the scope and objectives of HIPPY, which were in the context of house building, and influenced by resource constraints and the principles of software prototyping. A brief description of the potential practical application of the prototype system is also presented.

5.2 SCOPE, OBJECTIVE AND CONTEXT OF PROTOTYPE SYSTEM

The scope of the prototype system is defined as the area (or range) of on-site construction activities and the operations being supported, and is generally defined by the focus of this research work (i.e. on-site construction). The objective describes what the prototyping is specifically aimed at achieving, through the individual features of the prototype system. The context includes a description of other aspects of the prototype, such as who uses it, where can it be applied, what resources are used (e.g. available software platforms), and duration of work.

5.2.1 Scope of HIPPY

The scope of the prototype system (HIPPY) was limited to the on-site construction stage of a small to medium size building project. More specifically, it manages the integration of product and process information essential to the construction manager on a house project, in order to facilitate an efficient information flow. The prototype system focused on integrating various information sources, such as the information models, described in Chapter 4, graphical/non-graphical documentation, and a variety of other software development applications (e.g. CASE modelling tools and database applications).
5.2.2 Objectives of HIPPY

HIPPY is a prototype system for the proposed integrated conceptual model, which incorporates the product and process models, document files, and various other information sources (described in Chapter 4). Information technology is intended to demonstrate the implementation of the information models in a computer-based environment, and to show how these facilitate the effective integration of 'product data' (based on building elements, components, etc) and 'process information' (based on the necessary on-site construction activities needed to construct the building elements). The development of HIPPY reflects the growing trend for integrated construction information modelling tools and techniques to be computer-based. The objectives of the prototype system include the following:

- to develop an object-based development information system capable of utilising entities from both product and process models, as well as fulfilling a number of requirements (described in Section 4.3) emanating from observed shortcomings in existing information systems for construction purposes (see Section 1.3);
- to demonstrate the basic concepts of the integrated product and process models-based information systems (documentation, generation, etc);
- to demonstrate the integration and management of product and process information (through modelling, documentation, measurements and calculations of building elements, traceability of information); and
- to explore the potential for future integration of information management with other computer-based design and construction tasks.

In order to achieve the outlined objectives of HIPPY, the prototype was intended to accommodate system attributes which:

- facilitate an object-based platform capable of managing, storing, viewing, editing, retrieving, updating, integrating and processing construction project information;
- facilitate the reuse of stored information at a later stage in the processing activity;
- facilitate calculations of building products, such as building component and building element measurements for downstream construction operations;
- provide an integrated Windows environment that supports multiple tasks by including multiple software applications;
facilitate time and duration information of construction activities;
be practical and easy to use, with reliable and consistent outcomes;
facilitate the generation and printing of reports on various aspects of the information models and document files;
provide a certain degree of intelligent support (adequate instructions on how the system operates) in a given context; and
allow for future extension of the system, and integration with other computer-based downstream (e.g. facility management) construction activities.

The list of system requirements for HIPPY is by no means exhaustive, although, it does represents the basic capabilities and system attributes for demonstrating the computer-implementation of the integrated product and process model. However, in order to appreciate the extent in which these were applied, Chapter 6 demonstrates the use of HIPPY to on-site construction (with house building as a test bed), and provides an evaluation of the system requirements.

5.2.3 Context of HIPPY

The initial objective of improving communications between construction site managers, clients and contractors, in addition to the other participants involved with the building delivery process provided the rationale to the research study. The need for integrated information systems capable of managing both product and process information is becoming 'vital' for construction companies in acquiring, structuring, manipulating, and exchanging complex technical data, which they have to handle during the on-site construction stage of a construction project. The prototype system is therefore mainly aimed at the small to medium sized enterprises (SMEs) who often have to face situations for which they do not have the necessary skills or tools for continuously updating the technical information needed by the projects they work on, as discussed in Section 4.3. In this regard, the system uses a-three-tier client-server architecture to demonstrate the integration between products (building resources, elements, components, measurements, quantities, etc.) and processes (operations, activities, functions, tasks, scheduling, durations, etc.).

The following sections describe the main characteristics and stages as they apply to the development of the prototype system (HIPPY).
5.3 PROTOTYPE SOFTWARE DEVELOPMENT

This section briefly discusses prototyping and the approaches, and strategies used to develop prototype systems.

5.3.1 Approaches to Prototyping

Many information systems have been developed for construction; all exploiting a variety of methodologies (e.g. structured development life-cycle approach, object-oriented approach, etc) to assist in the processes involved (Avison & Fitzgerald, 2001; Lee, 1993; Isensee & Rudd, 1996). In practice, each system incorporates particular CASE tools and techniques (e.g. system analysis, prototyping, etc) for information collection, documentation and diagrammatic representation, in addition to automated software packages to assist in these processes involved. However, there are many different types of prototyping, all utilising variant methods, tools, or concepts. For this reason, they are most suitable in different types of system development and/or environments. For instance, physical prototyping is used to build a physical mock-up for the developing system. The mock-up can provide the end-users with a very close impression of what the real system will be like. Digital prototyping, or so-called virtual prototyping, is the method that uses the computer to build a working model of the system. This is now becoming the preferred method for the development of prototype software because it can iteratively improve the design and reduce cost in developing a system.

Rapid prototyping, one of the most popular methods for prototyping, is an approach which utilises less formal development methods and standards in order to provide for a fast development cycle that quickly produces an initial version of the full product (Bennatam, 1995). This is also the reason why it earned its reputation of 'quick and dirty' from its opponents. The utilisation of the aforementioned prototyping methods is often overlapped. Therefore, no single prototyping method is used in a given system. To ensure that the desired objectives and system requirements of the prototype system were satisfied, the analysis and development of HIPPY was carried out utilising elements from two different approaches: rapid throwaway prototyping and evolutionary prototyping. A brief description of the adopted methods follows.
Chapter 5

Rapid ‘disposable or throwaway’ prototyping (sometimes referred to as revolutionary prototyping) is generally used to analyse the uncertainties of a system (Crinnion, 1991, Bennatam, 1995). In this approach, working models, typically in the form of quick and dirty prototypes of the various elements of the system are built (usually at a very early stage in its development), verified with customers (e.g. regulator, clients, etc), and thrown away until a satisfactory prototype is reached; at which time, full-scale development begins (Isensee & Rudd, 1996). In contrast, the strategy for evolutionary prototyping (sometimes referred to rapid application development) is to build a life cycle model in which a prototype system is developed in increments, so that it can readily be modified in response to the end-user and customer feedback. Evolutionary prototyping usually takes place after a more careful investigation, whereby a prototype is built based on known system requirements and understanding, the system model is then refined and evolved, resulting in a high quality prototype (Crinnion, 1991; Davis, 1995). Evolutionary prototypes form the backbone of the new system being developed, and are therefore not thrown away.

The software prototype also makes use of the object-oriented paradigm such as encapsulation (as described in Section 3.4), the central concept of object-oriented programming, which enables data and code to reside together, making up a definable and reusable object. The object-oriented software development life cycle shares several characteristics with the rapid prototyping approach, and is both iterative and evolutionary (Lee, 1993). As the prototype is intended to demonstrate the proposed methodology for managing and processing construction project information, in addition to utilising software tools, which enable rapid construction and iterative evolution of the system, it can be considered to be a rapid prototype. On the other hand, the information models (described in Chapter 4) represent significant systems analysis, which is a typical characteristic of evolutionary prototyping. The intention for HIPPY to evolve into a well-developed system for integrating product and process information, also reflect the characteristics of evolutionary prototyping.

5.3.2 Strategies for Prototype Software Development

As previously mentioned, prototyping is an iterative technique, which consists of developing an experimental system to demonstrate to an end-user. It is not based on any specific strategic approach or methodology, nor does it follow the traditional stages of software development, which involves systems analysis, design and programming, etc.,
According to Isensee & Rudd, (1996), choosing from the variety of strategic approaches for developing prototypes is often based on a number of major factors, such as:

- speed or actual time in which the 'disposable' prototypes can be constructed;
- the actual cost of constructing the prototypes;
- the computer skill level (i.e. the degree of experience and programming skill required);
- the degree in which the software prototype must faithfully represent the appearance (look) and interaction of the final product; and
- the incorporation of the appropriate features to enable the attainment of the objectives for developing the specific prototype.

The fundamental goal of prototyping in this research work is to construct a small-scale prototype system that can then be used by on-site construction managers to gather more detailed information, or examine the capabilities of the developing system. However, the implementation environment for software and general development is usually based on a number of options that according to Britton & Doake, 1996, include:

- programming in a procedural, third-generation language (3GL) (e.g. FORTRAN) where the programmer has to describe in detail how every task is to be carried out;
- programming in a problem-oriented fourth-generation language (4GL) (e.g. C, C++, Visual Basic) where the programmer merely has to define what must be done;
- using a general-purpose integrated package which incorporates facilities such as word processing, spreadsheets, database and report generators; and
- the use and customisation of specific application development packages.

5.4 IMPLEMENTATION ENVIRONMENT

This section briefly discusses the implementation environment with respect to the main factors, research constraints, and rational behind the development of HIPPY. It also describes the implementation and components of the prototype system, and selected software applications for the implementation environment of HIPPY.
5.4.1 Development of the Prototype System HIPPY

The development of HIPPY was influenced by a number of factors: Firstly, the general principles of prototype software development (formerly discussed); secondly, the results of the software evaluation for the application development packages (described in Section 3.6); and thirdly, the resource constraints of the research project (e.g. time, cost, etc.). Through an analysis of these factors provided the necessary input, with regard to selecting appropriate software for the implementation environment of HIPPY. As a result, it was determined that the use of third-generation languages would not be pursued, since relatively more efficient approaches can be used, such as fourth-generation languages and general-purpose integrated software packages. In addition, the results from the software evaluation revealed that many of the application development tools, although appropriate (for what needed to be developed), had limited facilities for transferring data and generating multiple custom shapes and stencils to support the various modelling standards needed to develop the information models (described in Chapter 4). Consequently, their use would have been incompatible with the integrated information-modelling platform, and also involved greater costs (e.g. acquisition of software, permission to access source code, etc.) and time to understand the packages and their capabilities, which are two fundamental resource constraints of this work (i.e. economy and ease of use).

Therefore, in the development of the HIPPY software, the following decisions were reached:

- Microsoft (MS) Windows would offer the best environment for the implementation of the integrated product and process information models.
- A general-purpose standalone CASE-modelling package (Microsoft Visio 2000 for Windows) would offer suitable facilities for performing an analysis of output needs (e.g. exploiting the various modelling standards used to construct the information models).
- A general-purpose database package (Microsoft Access 2000 for Windows) as an event-driven application would offer suitable facilities for designing multiple applications, and will be used to implement the information models developed within MS Visio 2000.
• A general-purpose integrated suite of (Microsoft Office 2000 for Windows) would offer excellent support for providing links to product and process document files, and for interchanging data types with other MS software packages.

• A high-level Fourth-generation language of (Microsoft Visual Basic for Applications VBA) will be used to produce the prototype code. VBA is common to almost all MS applications.

The incentive (or rational) for using the Microsoft Window environment for the implementation was due to the need to quickly develop the software program at minimal expense, utilising readily available application development packages, all capable of running in a Windows environment, which was a major factor in the software selection process, and complies with the resource constraints of the research. Also, MS Window packages can seamlessly interchange data between other MS Office objects and components, in addition to offering and sharing very similar features, such as MS Object Linking and Embedded (OLE) and Open Database Connectivity (ODBC) drivers, which can be used to access the database of several development tools and vendors, in addition to enabling reverse engineering of non-rational data sources.

However, it is important to mention that the Suite of Microsoft Office was not included in the evaluation, because it has become the dominant computer Windows environment integrated software package for PC users and is bundled with virtually every PC. In addition, a vast amount of literature has been published on it (i.e. everyone knows about it). The choice of these MS packages (Access 2000, Office, Visio) for the development of the HIPPY software was also based on the following rational:

• Access 2000 utilises an object-oriented database management system, composed entirely of objects (tables, queries, forms pages, micros, modules, etc), which have certain attributes called properties that determine their structure, appearance, and behaviour. This was ideal for the system analysis approaches adopted in this work.

• MS Access is also part of the Office Suite of programmes, and is compatible with other Window-based packages, such as Word, Excel, PowerPoint, and Visio 2000.

• MS development packages also utilise the full graphical capability of Windows components, and provides visual access to data, in addition to simple, direct ways of working, viewing, editing and transferring information. This was ideal for importing the information models and document files developed within MS Visio and MS Office.
- MS Office packages are readily available, easy to use, and cost effective, which fulfilled some of the resource constraints of this research. Furthermore, relational database systems play a vital role in the management and storage of construction information.

- In-built design tools in MS Access and MS Visio assist in the development of forms and the generation of results by exporting data to office applications, such as Excel for generating diagrams, charts, etc., and importing these back for presentation.

- All MS software tools utilise VBA, although not a true object-oriented programming language in the strictest sense, Visual Basic incorporate an object-oriented philosophy, based on an event-driven language, whereby an object can react to different events such as a mouse click. Furthermore, Access also facilitates through the use of macros and wizards, the automation of many tasks without the need for much VBA code.

- As a relational database, Access contains querying and connective capabilities that facilitate data navigation functions, and the minimisation of redundant data storage. This satisfies the reuse of stored information objective, identified in the system attributes.

- MS Access has also been used in similar object-oriented research projects dealing with the capture and management of construction information (Hussein, et al. 2000), client requirement processing and readiness assessment tools using CE concepts (Kamara, 1999; Khalfan, 2001), and for developing a computer-aided interactive performance tool for construction (Ahmad, et al. 2001).

- The capacity of Access is sufficient to facilitate future extension and integration of the software prototype with other computer-based construction activity or systems, which is an importance feature for future research.

It is also acknowledged that a relational database is not a true object-oriented database system, and may have limitations, when compared to true object-oriented database management systems (OODBMSs), in the handling of complex data types, and in their overall management (Eaglestone & Ridley, 1998). It does however make extensive use of some important object-oriented capabilities such as ‘encapsulation’. In Access, an excellent example of this is the addition of a command button to a form, allowing the data and VBA code to be encapsulated within the object (i.e. residing together), making up a definable, and reusable object. As a result, the extensive use of objects with all the intelligence encapsulated within the object lets users realise the same overall goal as with a true object-oriented development environment. That is, fast development of event-driven
graphical user interface (GUI) applications with limited need for writing extensive amounts of program code. Therefore, the availability of Access in a format that facilities the development of relational database applications, as opposed to the relatively new OODBMS technology, makes it more appropriate within the context of this research.

5.4.2 Implementation of the Prototype System (HIPPY)

A prototype software system (HIPPY) for the implementation of the integrated conceptual model has been developed to map the proposed concepts and emulate the on-site construction process. The system development reflects the need to realise the benefits of concurrent engineering through the use of information technology, in addition to the growing trend for the integrated construction information modelling tools and techniques to be computer-based. The prototype not only demonstrated how the research relates to construction site practice, but also ensured that the product and process models performed in accordance with the identified characteristics. It also facilitated an object-based platform capable of managing, integrating, storing, viewing, editing, retrieving, updating, report generating, and integrates product and process construction project information.

The software selected had to be object-based as it could then be used as a development tool for client/server applications. For this situation, the system prototype was implemented utilising the programming environment of Microsoft Access 2000 and the integration of Visual Basic for Applications (VBA) as its high level programming language. The rational behind the choice, is that Access is a powerful database package and a development tool under MS Windows, it therefore ensures wide application as most construction sites in the UK only use standard desktop computers with the basic Microsoft Windows applications. Using the capability of MS Access to attach to files stored on a SQL server, MS Access can be used as a development platform to provide applications for enterprise-wide database tasks.

5.4.3 Prototype System Components

The system components, illustrated in Figure 5.1 consist of three main ‘platforms’, each integrating various objects and components. Each component was specifically structured to provide the end-users with maximum ‘output’ with respect to integrating product and process models and information. Brief descriptions of these platforms follow.
5.4.3.1 Infrastructure Platform

The first platform represents the technical infrastructure such as the hardware and software that supports the ‘input’ of data from the users (construction managers) with regard to the product and process information required during the on-site construction stage of a house building. The hardware support includes user input devices (e.g. mouse, keyboard, scanner, etc.) and the graphical-display ‘output’ devices (Pentium processor and a high-resolution PC monitor). The software provisions include Microsoft (MS) Office, MS Access 2000 and MS Visio 2000 applications, all utilising Visual Basic for Application (VBA) Code and implemented as a distributed set of Access Objects for the Windows NT and Windows 95/98/2000 platforms, in order to take full advantage of the functionalities offered by the

Figure 5.1: System components for the HIPPY prototype
event-driven database-programming paradigm (e.g. Activity-X Data Objects (ADO), Data Access Objects (DAO) and Object Linking and Embedding (OLE) concepts).

5.4.3.2 Database Storage Platform

The second platform represents the object-based development storage facilities, and consists of the components of a relational database containing the fields and objects of the product and process models defined by the prototype's scope. The information models, described in Charter 4, are decomposed in a hierarchical manner, so as to facilitate the breakdown of the modelling data, and integrated into the database tables, queries, forms, reports, and macros used by the prototype application. The data includes building components and elements, etc., and the activities and tasks needed to build the components.

5.4.3.3 User Interface Platform

The third platform represents a general-purpose graphical user interface (GUI), the prototype development platform for managing and processing product and process information. The GUI is composed of several object modules, which facilitate the storage and retrieval of information from the database storage facility (platform 2). The modules contain the event procedures, such as a set of commands, in the form of macros and VBA code, which represent custom objects (forms and reports, and their properties) that respond to buttons being clicked, and provide the means for analysing structured data, then converting the data into useful information output (e.g. printing product and process reports). The GUI also contains numerous browser forms and queries for retrieving product and process information with respect to the on-site construction process (i.e. building a house). The user interface is the backbone to the prototype system, and interacts with all the modules by means of command buttons, which are interconnected. The arrows linking the modules indicate that entry, storage, viewing and editing of responses, can be achieved at any instance in the processing stage. It has been implemented as a set of Access objects, information models, MS Office document files, Graphic Web pages, and Pictures/Images to support the different elements and components. The output from the GUI includes numerous screen layouts with multiple menu choice, each providing various macros, command buttons and text controls for accessing a variety of information, which the
construction manager can enter, view, display and print reports, and manipulate construction project information during the on-site construction stage of SMsBP.

5.4.4 The Microsoft Windows Environment

The components of HIPPY have been distributed between various Microsoft Windows applications, so as to exploit the many potential benefits offered when working within an integrated Windows environment (Morris, 1999). Figure 5.2 illustrates the MS application development tools, which have been used in the development of the HIPPY software. The integrated MS platform not only represents the software applications for developing the information models during the system analysis stage, it also represents the applications for managing the interoperability (data transfer) between the prototype software components.
It is appropriate to mention that a vast amount of literature has been published on MS Visio and the MS Office suite of programmes; for this reason, only a brief description of how MS applications were applied in the development of HIPPY before implementation follows.

5.4.4.1 Microsoft Office 2000

Microsoft Office is composed of a suite of applications that complement each other to accomplish tasks in a similar way. They provide facilities for creating documents to meet various requirements, handle complex financial analysis using spreadsheets, and produce professional presentations; as well as facilitating easy access to data shared between the individual applications. In the work presented here, MS Office provided the means to categorise, document and store product and process information, with the aim of attaching links and exporting the files to the appropriate applications.

5.4.4.2 Microsoft Visio 2000

Microsoft Visio is a stand-alone modelling tool that provided the facilities for creating custom shapes and stencils to support the modelling standards used in the development of the information models, described in Chapter 4. As part of the MS suite of programmes, Visio has the ability to integrate with the Windows environment, and other commercial software packages such as CAD and XML applications, as well as sharing the same facilities with other MS packages for importing and exporting files/tools to various Web applications or locations (Visio, 2001). Visio contributed to the research work in the following ways:

- Provided the means to create information-rich models/diagrams from the information documented and stored within the Office repository, which could then be also exported, edited, opened, stored, or extended in Access and Office applications;
- Provided the means to document ideas, generate information, and systems to facilitate IT deployment; extend the use of developer tools, and engineering plans; and,
- Provided the facilities for categorising, storing and transferring data via OLE techniques to other MS application sources.
Within each of these roles, Visio also provides key benefits. These benefits include the ability to:

1) Easily make an impact by quickly creating and sharing professional-looking diagrams;
2) Understand ideas, information, and systems through a broad range of task-specific solutions that integrate with other Microsoft products; and,
3) Benefit from the diagram standard using a single, customisable diagramming program that can be deployed across numerous applications or organizations.

Once the information models had been developed, categorised, and stored within Visio's repository, data transfer techniques (e.g. OLE and hyperlinks) were used to connect various product and process information to diagrams, objects, relationship arrows, associations, components, files, applications, etc. When these links to the various applications and/or objects were initiated, users are able to edit or view the attached data/linked information, or navigate to any file stored within the applications of the MS Windows environment. The document files, information models, images, pictures, and Web files, which are all categorised and stored within the MS application repositories, are then implemented through a series of Access objects. A graphical representation of the facilities, which MS Visio offered is presented in Figure 5.3, and is used here only to graphically illustrate the potential capability, objects, components, functions, and software environment in which HIPPY is implemented.

The following sections briefly describe how HIPPY was designed using Microsoft Access as the enabling database technology. A description of the potential practical application of the system from a users (construction managers) point of view is also presented, using the foundation and wall details of a house building as an example. A more detailed description on how the database objects and components (prototype system) were developed within MS Access is presented in Appendix 1. However, before describing the development process for HIPPY, it is important to determine the use of the prototype software environment, so as to establish the main functional requirements, such as systems abilities, user facilities, and environment characteristics. For example, who is going to use the system (the end-users), why they are going to use it (e.g. problems with current methods), where the system is to be used (e.g. on-site planning and construction stages), how the system is used (manually or electronically), and the need for the system (e.g. to improve site efficiency).
Figure 5.3: Microsoft Windows Platform, illustrating the drawing facilities of Visio, and generic method and architecture of HIPPY
Chapter 5

5.5 POTENTIAL PRACTICAL APPLICATION OF HIPPY

The functional requirements of HIPPY are divided into the following four groups:

1. **System Abilities**: describing what capabilities the system should have, such as:
   - Ability to provide, maintain and display a meaningful model or representation (picture/image) of the data, which is consistent with the logic of the modelling world (as described in Chapter 4).
   - Ability to exchange information, based on common technology and terminology.

2. **User Facilities**: representing important facilities the user requires, such as:
   - The user is in control of the system and not the system in control of the user, implying that the system does not limit the sequence of design actions.
   - The user can see the knowledge, enabling the environment to be transparent.
   - User can modify knowledge, enabling the information to be edited and augmented by the user at any time (e.g., data transfer using modern methods, such as OLE).

3. **Environment Characteristics**: presenting not only important characteristics that an integrated environment should have (see Section 4.3.1), but also providing guidelines for evaluation of the system. Characteristics include:
   - The system should be flexible in nature, so as to support different applications and meet different requirements as they occur.
   - Ease of use (discussed later).

4. **Practical Application Dimensions**: representing the fundamental potential application and use of the system’s environment, such as:
   - Who are the end users of the system?
   - Where the system is to be used (construction area).
   - Why the system is important.
   - How the system is used.

Many of the system requirements have been discussed in the preceding chapter. Therefore, in the following sections, the use of the prototype system with respect to the development of HIPPY will be discussed to establish an understanding of the functional issues.
5.5.1 Use of the Prototype System

The design and development of the system is aimed at construction managers (project managers, site managers, etc) who have some experience of using computer applications (e.g. MS Windows, MS Office, MS project) and prior knowledge of on-site construction, although they may not be familiar with this particular integrated product and process-modelling (IPPM) environment. The identified problems with respect to fragmentation, information technology, site practices and planning of work generally (discussed in Chapter 1), would indicate a need for the construction managers to have knowledge of the practical issues and site requirements involved with on-site construction. The lack of designer or practical experience of end-users may also make the process of appreciating technical concepts difficult. Quite often it is difficult to relate the theoretical concepts to how work is actually carried out on construction sites. Moreover, the construction site is full of technical terms and jargon that can be a barrier to knowledge and understanding.

It is the construction manager who is responsible for ensuring the smooth progress of work across the on-site planning and construction stages. Also, project information is carefully managed, monitored and controlled by construction managers or construction organisations, where they apply their own experience and rules to run the project effectively. The specific experience or characteristics for a construction manager are defined by the tasks that are performed, and should include:

- considerable knowledge of the on-site construction process (e.g. construction planning and operations), as described in Chapter 4 (Section 4.2), and the building products, elements, components, etc.;
- application domain-skills (e.g. understanding how to perform tasks with IT computer applications using software tools such as MS project, CAD drawing tools, etc); as described in Chapter 3 (Section 3.5 & 3.6); and
- experience (limited) in modelling on-site construction processes (e.g. scheduling and/or sequencing of tasks, workflow diagrams, modelling plans and specifications, etc), as described in (Sections 3.1 - 3.4).

Construction/project management is also a key skill, which underpins the construction industry. Recent reports from Egan (2002) and Latham (1994) have brought new change philosophies and ideas into construction (e.g. concurrent engineering, partnering, design
management integration, knowledge management, lean production, etc), and are not just fancy terms used by consultants, they are key concepts and principles for construction managers. An information system capable of integrating the various heterogeneous information sources (e.g. drawings, plans, orders, quantities, specifications, etc) during the on-site construction planning and operations stages, would therefore play an important role in promoting awareness, providing support to better management and planning control, contribute towards new forms of project team working (such as concurrent engineering, partnering, knowledge management, etc), in addition to a learning and training environment to enhance understanding of the issues involved, and to enable the shared design to be implemented successfully.

However, many of the existing computer-aided (CAD) drafting tools, modelling tools, and domain-specific application systems are automating specific industry processes, and generate data structures in a proprietary format, which reinforces the fragmented nature of the construction industry. Furthermore, there is no established "systematic approach" for data collection throughout the various on-site construction planning and operations stages of a project. Therefore, despite the fact that much of the product and process information generated during the design and production stages of a construction project is useful for construction planning activities, no systems are available that can consistently represent both product and process information concurrently, or to 'visualise' and 'link' the graphical and non-graphical data needed in the planning stage of the on-site construction process. More specifically, construction managers typically receive 'masses' of information that is usually limited to CAD drawings, and document format (e.g. planning specifications, schedules, resources, activities, building standards, quantities, etc) from the designer or design team. This is usually in 'paper' format, although use of electronic methods (such as e-mail, computer CD-ROMs, and the Internet) is growing. As a result, the problem encountered is three-fold:

- Masses of paper documents clogging up the site office.
- Lack of data acquisition with respect to the information describing the construction operations (process elements), and information describing the physical resources needed to construct a facility (product elements) during the planning stages of the on-site construction process.
- Most of the data, such as building documentation, is in non-electronic media, which requires re-entering, and is individually entered.
5.5.2 On-site Construction Scenario

One of the biggest problems associated with many construction projects is 'dealing' with the masses of paperwork. For example, storing and filing information, documenting, changing and updating information, and finding the right document, in particular, drawings, plans and specifications, building measurements and quantities, ordering of materials, etc. Probably the first arguments construction managers have on-site are: “you issued the drawings late”, “I don’t have a full set of drawings”, “how long will the new drawings take and do they show all the alterations”, and “the material quantities are wrong”.

Therefore, what is needed is an information system capable of providing facilities for integrating the masses of product and process information, visualisation of graphical modelling data, links to other heterogeneous information sources, and enabling technology to allow construction managers to manually update or electronically enter, view and edit this information during the construction planning stage of a house-building. The prototype system (HIPPY) resolves the masses of paper problems, and provides many of the facilities described in the previous sections above.

The following sections briefly describe the development process and system architecture for HIPPY. It also provides a description of how the system can be used, illustrated through a practical application, using a simple foundation and wall design as an example. Chapter 6 presents a more detailed description of the potential practical operation of HIPPY.

5.6 DESIGNING HIPPY IN MICROSOFT ACCESS

The prototype system allows the construction manager not only to input product and process information on their own, but also to view, edit, integrate, and print the results in the form of text and graphical diagram reports, similar to the information models (described in Chapter 4). The development of the software followed the general procedure for developing MS Access database applications. According to Smith and Sussman (1999), one way of defining an Access application is to describe it as:

'A collection of database objects and VBA code that co-ordinate together to perform a common ask or set of related tasks to achieve a specific objective'.
Therefore, the development of HIPPY involved a co-ordinated set of database objects, in the shape of tables, forms, queries, macros, reports, modules and VBA code, which enable construction managers to maintain the data contained within the database. This was carried out utilising the facilities provided within MS Access, and linked with other MS applications for reviewing the integrated results of the product and process information.

The following are the steps taken to develop HIPPY using MS Access:

- designing tables;
- designing forms;
- designing queries to support application;
- implementing macro and VBA code;
- adding macro and VBA code to forms;
- putting the application together;
- providing navigation throughout the application; and
- sharing data with OLE and Hypertext Links to heterogeneous information sources.

The first task was to create the database file (.mdb file), since this is the container that will eventually hold a wide variety of different Access objects. Many tables were designed to store data (i.e. breakdown of the information models, document files, hyperlinks, etc); queries were designed to retrieve the data in meaningful ways; forms and reports were designed to display the results of the queries and the information stored in the tables, in such a way that can be understood by the end users (construction managers); and macros and VBA code (modules) provided the logic which 'glues' the whole application together.

The data output from the prototype system, as illustrated in Figure 5.1, can be viewed in a number of diverse formats. For example, Visio modelling templates, CAD drawings, Schedule planning charts, resource graphical pictures and images of equipment, hyperlinks attached to MS Office and HTML files, in addition to viewing and modifying the underlying product and process models/schemas using data transfer methods such as object, linking and embedding (OLE). The information captured, not only consists of links to the usual concepts found in existing integrated project models (e.g. buildings, wall components, activities, etc.) but also hyperlinks to construction project information including; components, measurements, activity durations, times, actors, roles, resources, equipment, plant, building standards and specifications, and environmental details in text.
format. For example, the tables store the product and process data, queries allow end-users to get at the data in a rational way, and reports and forms containing controls, such as text boxes and command buttons allow the data to be displayed in a user-friendly way.

5.6.1 Development Process of the Prototype System (HIPPY)

Irrespective of the number of people involved, or the development methodology employed, the development lifecycle for an Access application will typically involve the following steps (Smith & Sussman, 1999):

Analysis $\Rightarrow$ Design $\Rightarrow$ Coding $\Rightarrow$ Testing $\Rightarrow$ Documentation $\Rightarrow$ Acceptance $\Rightarrow$ Review

In practice, however, these steps do not rigidly follow one after another. It is beyond the scope of this report to enter in a detailed discussion of different project life cycle plans. However, it is acknowledge that the speed with which Access forms and reports were created makes MS Access an excellent tool for using in a more iterative life cycle approach, such as the procedure used to develop HIPPY. The development lifecycle adopted in this research, and shown in Figure 5.4, generally involved the following steps:

1) **System and Requirement Analysis**: this involves the evaluation of technical information and user's (e.g. construction manager) needs (i.e. system requirements from a user's point of view), in order to establish what they require the prototype system to do, with particular reference to identifying problems, flow of data, volume of data, technical operating system data, and establishing the entities, attributes, relationships.

2) **System Design and Coding**: the design of tables, forms/subforms and screens, and the sub-systems (e.g. queries, menus, macros, modules, and the VBA code and commands needed to perform the tasks) that will solve the information problems.

3) **Testing**: involves testing the prototype software, by means of a series of system checks and trial runs, leading to a final demonstration and evaluation of the finished system (described in Chapter 6).
4) **Documentation, Acceptance and Review:** involves documenting the software procedures and results from the evaluation of the prototype system; deciding what worked and what needed to be improved, analysing the final scores and remarks, and drawing conclusions from the results of the evaluation. It also includes reviewing the comments (if any) for improving the system.

---

**Figure 5.4: The development lifecycle process**

**5.6.2 Operational System Architecture of HIPPY**

Figure 5.5 illustrates the operational system architecture of HIPPY. The main user-interface provides access to all the object screens, which are interconnected. The arrows linking the object boxes indicate that entry, storage, viewing and editing of information, can be achieved at any stage of the process. The user-interface also displays a list of options containing the most frequently undertaken activities with one or more choices that
enable new users to open the various other pages. These contain buttons and menus for accessing the relevant forms, and for displaying and printing reports (output), which the construction manager will need to view or edit as appropriate.

Figure 5.5: Operational system architecture of HIPPY

Navigation control and human computer interaction (HCI) issues are particularly important and will, to a large extent, govern the success of the systems in terms of its use and communication of knowledge and information. Navigation control has been introduced to allow users with different levels of knowledge and computer literacy to utilise the systems. Also, self-assessment of knowledge acquisition by the designer has been made available (to new users) to check on theory, practical issues, legislations, and administration.
The design of the system is structured to include a flexible environment, not only regarding the various applications, information models, processes, products, and construction resources used in the produced solutions (output), but also regarding evolution (modification) of the information itself. It was also conceived that a simple computationally efficient system and model scheme would be easier to use and understand than complex and intelligent ones, so as to allow all levels of construction managers (e.g. training, assistant, and experienced) to use the system. It would be useless to design a complex and powerful industrial application if it is too sophisticated for anyone to understand. For this reason, the design of HIPPY was kept simple to be easily understood by any reasonably computer-literate construction manager, and to enable them to enter, view, and edit data. Ease of use features considered useful in the design of HIPPY include:

- the user-interface should be clear and easily understood;
- the environment should be based on accepted and known terminology;
- transparency should apply, providing construction managers with means for understanding what the system and the tools do; and
- the visual representations or information models must be clear and easily understood.

After the main user interface screen was developed, the next page (welcome screen) provides a short introduction to how to get started and what HIPPY can do, followed by various screens and pages explaining briefly the aims and objectives of HIPPY, administrative issues, and potential integrated application sources. When the construction manager has viewed the relevant information with respect to operating the system, the next stage is to enter, view, and edit product and process information. The next section provides a practical application of the development of the database objects.

5.6.3 The Development of Database Object for HIPPY

An application example is presented here to demonstrate the potential practical use of the system, from a construction manager's perspective. For simplicity, a simple drawing of a foundation wall design and associated information details (e.g. products, process activities, actors, resources, equipment, documentation, etc.) is used, as an example to demonstrate how the information is incorporated into the system, using the database objects of MS Access.
The foundation wall design, as illustrated in Figure 5.6, is part of the substructure for a house building. The construction manager will generally receive this information with respect to the house building from the design team, in the form of building drawings, bills of quantities, plans and specifications, sequence of activities, equipment, construction team, etc., and usually in paper format.

**Figure 5.6: Foundation wall design**

Table 5.1 also presents a simple list of the information details, which a construction manager would typically need, in order to construct the foundation wall.
Chapter 5

Table 5.1: Information Details for Foundation Wall Design

<table>
<thead>
<tr>
<th>Product Elements</th>
<th>Environmental Constraints</th>
<th>Process Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Measurements</strong></td>
<td><strong>Drawings</strong></td>
<td><strong>Organise Site Area</strong></td>
</tr>
<tr>
<td>Foundations:</td>
<td><strong>Plans &amp; Specifications</strong></td>
<td><strong>Excavate Site Area</strong></td>
</tr>
<tr>
<td>Length = 8.50 m</td>
<td><strong>Safety Regulations (CDM)</strong></td>
<td><strong>Construct Foundation</strong></td>
</tr>
<tr>
<td>Width = 0.540 m</td>
<td><strong>Building Standards</strong></td>
<td><strong>Construct Wall</strong></td>
</tr>
<tr>
<td>Height = 0.225 m</td>
<td><strong>Planning legislation</strong></td>
<td><strong>Inspect And Improve</strong></td>
</tr>
<tr>
<td>Brickwall:</td>
<td><strong>Information Models/Schema</strong></td>
<td></td>
</tr>
<tr>
<td>Length = 8.0 m</td>
<td><strong>Plant &amp; Equipment</strong></td>
<td><strong>Durations</strong></td>
</tr>
<tr>
<td>Width = 0.225 m</td>
<td><strong>Hydraulic Digger (JCB)</strong></td>
<td><strong>Start Times</strong></td>
</tr>
<tr>
<td>Height = 3.0 m</td>
<td><strong>Surveying Tools</strong></td>
<td><strong>Finish Times</strong></td>
</tr>
<tr>
<td><strong>Product Materials</strong></td>
<td><strong>Mortar Mixer</strong></td>
<td></td>
</tr>
<tr>
<td>Concrete</td>
<td><strong>Concreting Tools</strong></td>
<td><strong>Building Team</strong></td>
</tr>
<tr>
<td>Bricks</td>
<td><strong>Bricklaying Tools</strong></td>
<td><strong>Site Agent</strong></td>
</tr>
<tr>
<td>Brick Mortar</td>
<td><strong>Dumper</strong></td>
<td><strong>Digger driver</strong></td>
</tr>
<tr>
<td>Wall Ties</td>
<td></td>
<td><strong>Bricklayer</strong></td>
</tr>
<tr>
<td>Dpc</td>
<td></td>
<td><strong>Concretor</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Labourer</strong></td>
</tr>
</tbody>
</table>

5.6.3.1 Designing Tables and Forms

The foundation wall detailed information, presented in Table 5.1 is represented by the information models and process activity information component (described in Chapter 4), and are decomposed in a hierarchical manner (so as to facilitate the breakdown of the modelling data) and incorporated in the database storage platform using tables and forms. A database table is the basis of the MS Access applications. Multiple tables are designed for the various headings or entities (e.g. activities, materials, equipment, actors, products, etc), and linked by creating relationships between them, which means that all the product and process tables are related or joined together, depending on the type of relationship. 

Figure 5.7 shows a typical table illustrating how data is grouped and organized.

<table>
<thead>
<tr>
<th>Activity Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunctionID</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Figure 5.7: Datasheet view of an activity table
However, the information for specific data, such as building standards, MS Word document files, HTML Web pages, graphical modelling data, pictures, and images were obtained via external sources, through the use of hyperlinks and OLE method fields. A more detailed description on how the tables and relationships were designed can be found in Appendix 1.

The tables provided the foundation for creating the forms, which are used for displaying, entering, and editing information into the tables. The information is entered in the form, whereby the application stores the data in the corresponding (underlying) table within the database platform. The product and process browser modules (database storage platform), illustrated in Figure 5.1, correspond to the information captured in the information models; document files, Web pictures, etc., basically involved the documentation of various types of information. The information is decomposed in a hierarchical manner, to facilitate the data breakdown and stored in tables, which can be facilitated further by simply designing multiple forms based on the tables storing the data. Since there is more than one form for each table, they are linked by using macros and VBA code. This allows the user to go through all the form and enter information continuously without having to leave the system and go back in again.

The form designs are made up of many elements called controls, which represent the data displayed or the action performed by the user. They also run the macros and VBA code attached to OLE objects, calculated value expressions, hyperlinks to graphical information models, pictures and Web images. Three types of forms have been designed to accommodate the breakdown of modelling data and information. These include:

1. Single forms that only allow the end user to enter, view, and edit product or process information in text format.
2. Hierarchical forms that consist of subforms (or multiple subforms) allowing the end user to enter, view, and edit product and process information in text format, as illustrated in Figure 5.8.
3. Multiple hierarchical forms consisting of a subform (or multiple subforms), allowing the end user to enter, view, and edit various product and process information, from a variety of heterogeneous information sources (e.g. information models, hyperlinks to Web pictures/images, links to MS Office documents files, database applications, calculation functions, etc), as illustrated in Figure 5.9.
Figure 5.8: Hierarchy form showing product element fields, controls, and property sheet

Figure 5.9: Multiple form design showing process information and model fields
The two-hierarchy forms have been developed to provide the user with more freedom in entering, viewing, and editing information. Properties are used to establish the characteristics of forms and report design elements, whereby every objects in a form or report design has properties attached. These properties set the structure, appearance, and behaviour of the controls, and can also determine the characteristics of the text and data contained in a control. Control objects who's properties are based on calculation fields (Figure 5.8), acquire their data from values in tables and are 'expressions' containing functions and operations, in addition to fields, that produce a result for calculating material quantities. For example, the expression is used to perform the calculation generally is based on one or more fields of the underlying table or query, and enables the site manager to order the correct amount of materials without waste. The value shown in the calculated control field, changes as the values in the underlying fields change. However, calculated controls cannot directly be edited.

In developing the calculated fields, simple queries were used and attached to the calculation control fields to give the total amount for that particular element. First, MS Access presents the data as a set of records in what is called a dynaset. A dynaset strongly resembles a table; it is a dynamic set of records derived from a table on which the query is based. If changes are made to the data in the answer provided by the query, the data in the original table will not change. Queries can be created manually or using the Query Wizard. The objective of creating a query is to provide the users with calculated quantities for facilitating benefits, such as the precise ordering of material quantities to reduce the cost of waste. To perform the calculation in the query, open the query in 'Design View' and choose 'View' – 'Total'. Inside the 'Total' row in the query grid appears the designation 'Group By' on every field. To choose the options, click the down arrow to open the list box of possible calculation types and choose the desired type (see appendix 1).

The model frame (Figure 5.9) has been designed to allow construction managers to alternate between drawing applications, through the use of data transfer techniques. When selected, activates the OLE automation option between the different applications of the prototype system, which enables the user to edit, change, update, delete or develop new diagrams within the software modelling tools original source file. The multiple hierarchy forms also provide a variety of other options for accessing, viewing or editing information from various heterogeneous data sources, on account of the multiple hyperlinks connected to the control field, or attached to the diagram components (described in Section 4.4.2.7).
Chapter 5

Navigation and functionality for all the form designs include hyperlinks to various data sources, records or applications, and selection of appropriate subject areas to display the relevant information or form screen. For example, selection of 'Brick Wall Diagram' in the subform at the bottom of Figure 5.8, will display the diagram, as illustrated in Figure 5.6, and display information in more detail. A more detailed discussion of the different form designs, and the options available is described in Chapter 6 and Appendix 1.

5.6.3.2 Designing Reports and Macros

Forms were developed for inputting data and viewing it on the computer screen. However, to enable the construction manager to view hard copies of this data, reports were developed for converting data to written information (output). A report is a printed document that displays information (e.g. the finished product, such as the foundation wall design) from the data tables stored in the database application. The reports were developed in much the same way as the forms, and composed of similar design elements with event procedures, macros, VBA code, or expressions attached to them. In order to provide navigation control options, a command button was created on the forms and linked to the event procedures, and when activated, the users can either preview or print the selected information in a more professional manner. A special option was also created for selecting integrated reports, enabling users to preview and print integrated product and process information reports. Figures 5.10 - 5.12 illustrate various reports, which offer the construction manager addition feature for viewing information such as, process activities and durations, product elements, material sizes and quantities, graphical modelling representations and summaries of various heterogeneous information sources.

However, at this stage of the development process, reports were used to demonstrate how the information could be implemented, and only a few selected forms and data fields have been developed (using macros and VBA code) to support the integration of product and process modelling information report building. A full report generation module can be introduced at a later stage of development, which would not only integrate text but also diagrammatic representation of the selected data.

Macros are a fundamental part of creating an application in MS Access. They can provide complete menu-driven systems that will help users to use a system more easily.
Chapter 5

Figure 5.10: Report ‘Design View’ for product and process information

Figure 5.11: Report ‘Print Review’ of integrated product and process information
Figure 5.12: Report structure illustrating process activity modelling information

A macro is essentially a list of actions that is specified in advance and, when it is run, MS Access carries out that list of actions. Specifically choosing the actions to be performed in a macro window can do this. The macro window is divided into three areas - an upper portion that contains the macro name, the actions and the comments. The macro name column specifies the actual name given to a macro. Within the action column, the actions are selected from a list in a pull down menu and the comment column describes what the macro does. The comments are optional and provide a reference to help remember how the macro operates.

Although, macros have their uses for simple tasks, they do however; possess limitations when developing buttons that are intelligent enough to allow users to click them only if they represented a valid choice (i.e. disable or enable macros according to where they were in the records behind the form). While this may not be a problem for some users, one of the
requirements of HIPPY is to provide a certain degree of intelligence, facilitating a user-friendly interface that will appeal to the end-user (i.e. encourage people to use the system). To achieve this, the Command Button Wizard was used to generate VBA code. This enabled a more varied selection of actions to be executed than were available through macros. The use of VBA code also executes faster than the equivalent macro action, and since ‘speed’ is generally a critical factor in impressing the end-user, it was therefore practical (based on reason) to convert macros (except the macros used for display) into VBA code.

Once the macros or VDA code have been defined, the next step is to attach them to forms that have been developed. In HIPPY, most forms have a dialog-box ‘Next’ or ‘Previous’, providing the option of either opening the next form for the former or returning to previous forms for the latter. This allows flexibility for the user to either continue with entering information on the following page or return to view the previous page. Most of the macros and code developed in HIPPY are for navigational purposes, such as open and close-form actions. In addition, all forms are equipped with self-explanatory labels and object control buttons (e.g. when a user moves the mouse pointer over a control in the form, a message appears specifying a control tip, which has been attached to that particular label or control). Also, users are provided with further instructions or explanations on the status bar at the bottom of the form, whenever a text box or control button receives the focus.

It is important to mention that once the information has been categorized and placed in tables by the design team or system designer, most if not all, can be reused at a later date because many projects, although unique in nature, follow or make use of similar processes and products. When different activities or products are used, the construction manager entered them in the appropriate form, and the database application stores or saves the new data in the underlying table. In addition, the models provide a set of templates, allowing construction managers to enter their own constraints, parameters, etc., in the forms provided. This information is also saved and updated in the model’s original source files and the corresponding table.

To use the system, the users (construction managers) are first introduced to the main interface page where they can choose from the menu the options that best describe what action is to be carried out. They then go through all the form designs; entering, viewing and editing the appropriate information with respect to the data received from the design
team, such as drawing, specifications, standards, measurements, etc., until all the information has been entered into the forms, or HIPPY takes them back to the main page.

The data output from the prototype system can be viewed in a number of diverse formats. For example, Visio modelling templates, CAD drawings, Schedule planning charts, resource graphical pictures and images of equipment, hyperlinks attached to MS Office and HTML files, in addition to viewing and modifying the underlying product and process models/schemas using data transfer methods such as object, linking and embedding (OLE). The information captured, not only consists of links to the usual concepts found in existing integrated project models (e.g. buildings, wall components, activities, etc.) but also hyperlinks to construction project information including; components, measurements, activity durations, times, actors, roles, resources, equipment, plant, building standards and specifications, and environmental details in text format. For example, the tables store the product and process data, queries allow end-users to get at the data in a rational way, and reports and forms containing controls, such as text boxes and command buttons allow the data to be displayed in a user-friendly way.

5.6.4 Transferring Data and Objects between Forms

The Windows environment provides a high degree of compatibility between applications and, as a result, the HIPPY platform provides several ways of transferring data between applications. Hyperlinking is one method by which different links can be created between applications and their objects. The Windows clipboard is another method by which different types of data: text (either standard text or DDE link data) and graphics (bitmaps, metafiles, etc) can be moved from one application to another. It is also possible to communicate between Windows applications and ensure that product data is updated automatically. Two methods are used: dynamic data exchange (DDE) and object linking and embedding (OLE).

The DDE method allows links to be created between applications with DDE capabilities. Most, if not all, of the applications used to create HIPPY have DDE/OLE built in. The OLE method is an alternative form of data transfer that allows applications or components to be activated from within another program allowing several types of data to be processed. The HIPPY prototype supports two primary aspects of OLE: the compound document architecture and the OLE automation facility [Kimmanence, et. al, 2003].
Chapter 5

Many, if not all, of the forms and reports created in HIPPY were developed using objects that can give the prototype system remote control over other MS applications (e.g. Excel, Word, PowerPoint, Visio, etc.), allowing these applications to use OLE automation to access the HIPPY database via MS Access. The OLE objects remain stored in separate files and a link is established between OLE objects in MS Access and the original file. However, any changes made to these files, will automatically change (or update) the connected data source fields (files) within the HIPPY application. For data security reasons, only the construction manager will be dealing with this section.

This type of control could extend over just about any other type of application, and is another major discussion point for further research work into OLE-BD and ActivityX data objects (ADO) for transferring data via the Internet.

5.6.5 Hippy and the Implemented Features for the Prototype System

The development of HIPPY was designed to satisfy the required scope, objectives, context, and features for the prototype described in Section 5.2. Table 5.2 presents a summary of how these features were implemented in HIPPY.

Table 5.2: Summary of how the Required Features for HIPPY were Implemented

<table>
<thead>
<tr>
<th>Required Feature</th>
<th>How it was satisfied in HIPPY</th>
</tr>
</thead>
</table>
| Provide an object-based platform capable of managing, storing, viewing, editing, retrieving, updating and processing construction project information. | • Designed forms allowed the display and editing of stored data.  
• The customised menu included on every form facilitated the display/editing of information at any stage of the process.  
• The development of more than one form for a given table also, facilitated the display and editing of information at appropriate stages in the processing activity. |
| Automate data exchange between different tasks and applications of the system. | • Use of data transferring methods such as hyperlinking, object linking and embedding.  
• Creating unbound controls and linking them to forms and reports via macros and VBA code. |
| Facilitate the reuse of stored information at a later stage in the processing activity. | • Use of Access database for product and process information.  
• The insertion of sub-forms also facilitated the reuse of stored information at a later stage in the processing activity. |
| Facilitate material resource calculations and measurements for downstream operations. | • This was achieved through adding calculated fields, event procedures using the expression builder, macros and code attached to functions.  
• The entry and storage of values was also achieved in HIPPY via an automated process for determining values using filtering and queries. |
Chapter 5

Be simple but not simplistic. Easy to use with adequate instructions on how to carry out the processing activity.

<table>
<thead>
<tr>
<th>Allow more than one user to use the system, both simultaneously and at different times.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow for future extension, and integration with other computer-based construction activities.</td>
</tr>
</tbody>
</table>

- Although all 'Help' menus were not implemented in HIPPY, every effort was made to ensure that forms were clear and unambiguous.
- Guidance was also provided on every control, by written instructions in the 'status bar text' property of those controls.
- This ensures that when a button or text box receives the focus, a text appears on the status bar of the form to explain what action is required.
- PowerPoint presentations were attached to record fields.

- At the current stage of development, HIPPY is designed as a single user application. However, further research into Access pages and data transfer methods on the Internet is on going. Therefore, it should be possible to transfer HIPPY on to a network that facilitates multiple use.

- Several reports, which summarise and/or combine different kinds of information, were designed.
- Appropriate macros/code and buttons were also designed to open these reports.

- This was achieved through enforcing referential integrity using the cascade features and relationship functions.
- Also editing Via OLE allowed automatic updating and deleting of data.

- The ease with which it is possible to add other tables to HIPPY should facilitate its extension.
- As an application in the MS Office group, Access can be successfully linked with other packages such as Word, other database systems, and AutoCAD.

It can be seen from Table 5.2 that most of the features necessary to satisfy the objectives for developing the prototype were implemented in HIPPY.

5.6.6 Benefits of HIPPY

The benefits of such a system capable of providing these integrated facilities lie in added data consistency, improved lifecycle product and process information support for the construction manager during the on-site planning and construction operation stages, better management and planning control, provides an insight to activity time/duration factors, in order to provide a clear picture of what needs to be done, such as process activities, product quantities and ordering of materials, reduction in errors and time, improved communication and collaboration between construction team members, and improved management of the overall construction industry. A more detailed description of the direct benefits that HIPPY offers to construction managers is presented in Chapter 7.
5.7 SUMMARY

This chapter described an integrated information platform that integrates product and process modelling information to effectively support collaboration and project team communications during the design and construction stages of construction projects, using house-building as a test bed. The chapter also detailed an integrated information prototype system (HIPPY), developed to manage, store, retrieve and update product and process information essential to construction managers, and identified the information requirements associated with its implementation. The development of the prototype software for the implementation of the information models described in Chapter 4, which was influenced by the principles of software prototyping and resource constraints, was designed within Microsoft Access. The design of the various elements, objects, functional requirements, features, and ways in which HIPPY can facilitate the integration and management of product and process information, have also been described. A brief description of the system architecture and potential practical application was also presented.

The integration of product and process information is achieved through the combination of data transferring methods and the Microsoft software applications. Although, as is characteristic with prototypes, certain features were not fully implemented at this stage of its development, for example the provision of limited online help and a full report-generating module, it is clear that HIPPY adequately demonstrates the implementation of the information models (integrated conceptual model) in a computer environment.

The next chapter discusses the potential practical operation and evaluation of the prototype system and demonstrates the effectiveness of HIPPY in managing and integrating product and process information.
CHAPTER 6

OPERATION AND EVALUATION

6.1 INTRODUCTION

This chapter demonstrates the use and structure of the integrated information modelling system prototype (HIPPY) and the procedure that governs the operation of the system is explained. The actual building products (elements, components, etc) and processes (activities, functions, operations, etc) for a house-building (represented by the product and process models described in Chapter 4) are used, as an example to explain the system’s operation and functionality. The chapter also describes how practitioners in the industry evaluated the prototype system. The suggestions, recommendations and results of the evaluation are also reported in this chapter.

6.2 OPERATIONAL OBJECTIVES

The prototype software is a computer implementation of the product and process information models, structured to provide templates, allowing end-users (construction managers) the interactive input of on-site construction information. The objectives for its operation therefore derive from the requirements to support the integration of product and process information during on-site construction, with house-building as a test bed. The operational objectives of the HIPPY implementation were to:

1. Provide a flexible, easy to use object-based information modelling system capable of managing, storing, viewing, editing, printing, updating, and integrating product and process information. The prototype system should demonstrate:

   - the integration with other product and process information and application sources;
   - the generation of graphical product and process diagrams; and
   - the process for generating reports on various aspects of the product and process models, document files, and integrating them.

2. Automate data exchange between different information tasks and applications of the system, in order to demonstrate the use of:
innovative data transfer approaches, such as hyper-linking, object-linking and embedding (OLE);

- the measurement and calculation of building elements or components; and

- the production of project sequence information for the duration of construction activities, and associated information.

The actual product and process information for the substructure of a 'house-building' are used as a test bed to demonstrate the prototype system's use with respect to achieving the operational objectives. The system has been designed as a demonstrator, which will show support for the effective integration of product and process information, structured to provide a template, allowing the end-user to enter this product and process information, or their own constraints, parameters, etc., in an interactive way. In this respect, the prototype system is aimed at construction managers (project managers, site managers, etc) who have some experience of using computer applications (e.g. MS Windows, MS Office), as discussed in Chapter 5.

6.3 SYSTEM OPERATION

This section demonstrates the use of HIPPY for integrating product and process information. It has been designed to operate on a PC running Windows 95 or better and requires Access 97 (or above) to be installed. It needs about 2Mb of RAM disk space to store the application and at least 5Mb of RAM to store data.

6.3.1 Starting the Prototype System (HIPPY)

The HIPPY application is stored as a Microsoft Access file called 'hippy.mdb' and is held in a directory named 'IntegratedProject'. To start the application from the Access menu, the user selects 'File: Open'. When the file is first opened, the welcome screen for HIPPY is automatically displayed, as illustrated in Figure 6.1. The screen contains a short statement about what the application is, and what it can do. The two command buttons on the form 'continue' and 'exit' allow the user to either proceed to the next stage, or to quit the application. Clicking on the 'continue' button opens the main graphical user interface control form, which contains links to all other forms in the application.
Welcome to the Hyper-Integrated Product and Process Information System (HIPPY), a prototype system you can use to edit and review construction product and process modelling information, and also create new records. Users can evaluate and experiment with the existing data stored within the HIPPY database, and exploit the forms, reports and other database objects as models for their own data system.

All construction objects in HIPPY are available from the main HIPPY Graphical User Interface (GUI), an integrated Database window browser, which will be displayed when you click continue. In the integrated product and process database browser, you can display descriptions of the objects by clicking Open Database Window then Details on the View Menu.

Figure 6.1: Welcome Screen

Clicking on the ‘exit’ button enables the user to quit the application. To proceed with the integrating (or processing) of product and process information, the user will have to open the main user interface. This gives the user a choice of activities to embark on. The main user interface displays a list of options containing the most frequently undertaken activities with one or more choices that enable a non-technical person to open the various other user interfaces (within the database) that list the relevant product and process forms and reports that the user will need to view or edit as appropriate. This type of interface is called a ‘switchboard’, illustrated in Figure 6.2, and described in Chapter 5. Menu items also provide access to forms and reports. This allows for continuous operation without the need to close a form before moving on to the next activity. However, to minimise computer memory use, it is advisable that the user closes a form before going on to the next one.
6.3.2 Defining Information Requirements

In order to make the integrated information modelling system efficient, it is necessary to provide users with a consistent user-interface through which they can access product and process information. This involves the input of information, which is represented by the integrated product and process model (described in Chapter 4) and stored within the database tables (see Appendix 1). The proposed information includes:

- process information to support the construction activities such as, resources, equipment, specifications, plant, and activity durations (start and finish times);
- product information needed to construct the house facility, such as building elements, components, and the measurements/calculations of these elements, components, etc;
- integrated information generated from heterogeneous sources (e.g. hyperlinks, OLE technology) to sustain the automated data exchange between different information tasks and applications.
Various forms are used in a hierarchical (top-down) manner to input, view, edit, and integrate this information with respect to the house-building substructure example. These forms can be accessed by clicking on the appropriate buttons on the main user interface switchboard form, by selecting one of the options on the Windows menu, or by clicking on the ‘open database window’ button. A list of the various forms or object commands linked to the main user interface is presented in Table 6.1.

Table 6.1: List of Forms and Commands Accessible from the Main Switchboard

<table>
<thead>
<tr>
<th>Main Form Reference</th>
<th>Connected Form/Command Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to Get Started:</td>
<td>Review HIPPY LOGO</td>
</tr>
<tr>
<td></td>
<td>Review HIPPY Presentation</td>
</tr>
<tr>
<td></td>
<td>Return to Main HIPPY Switchboard Interface</td>
</tr>
<tr>
<td>View process Modelling Information:</td>
<td>View/Edit Top Level Process Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Top level Sub-process Activity Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Sub-activity Function Box Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Function Box ICOM Arrow Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Activity ICOM Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Process Activity ICOM Type Form</td>
</tr>
<tr>
<td></td>
<td>Return to Main HIPPY Switchboard Interface</td>
</tr>
<tr>
<td>View Product Modelling Data:</td>
<td>View/Edit Top Level Product Data Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Top Level Sub-Product Data Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Types of Building Elements Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Product Building Elements Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Types of Building Components Form</td>
</tr>
<tr>
<td></td>
<td>View/Edit Building Components Data Entry Form</td>
</tr>
<tr>
<td></td>
<td>Return to Main HIPPY Switchboard Interface</td>
</tr>
<tr>
<td>View Integrated Sources:</td>
<td>Merge with WORD</td>
</tr>
<tr>
<td></td>
<td>Merge with EXCEL</td>
</tr>
<tr>
<td></td>
<td>Merge with VISIO</td>
</tr>
<tr>
<td></td>
<td>Merge with CAD</td>
</tr>
<tr>
<td></td>
<td>Import Output data From OUTLOOK</td>
</tr>
<tr>
<td></td>
<td>Preview Output Data as HTML</td>
</tr>
<tr>
<td></td>
<td>Return to Main HIPPY Switchboard Interface</td>
</tr>
<tr>
<td>Preview Reports:</td>
<td>Preview Process Function Boxes</td>
</tr>
<tr>
<td></td>
<td>Preview top-level Product Data</td>
</tr>
<tr>
<td></td>
<td>Preview Table Relationships</td>
</tr>
<tr>
<td></td>
<td>Preview Integrated PP Information</td>
</tr>
<tr>
<td></td>
<td>Return to Main HIPPY Switchboard Interface</td>
</tr>
<tr>
<td>Administration:</td>
<td>Change Switchboard Items</td>
</tr>
<tr>
<td></td>
<td>Remove Sample Data</td>
</tr>
<tr>
<td></td>
<td>Return to Main HIPPY Switchboard Interface</td>
</tr>
<tr>
<td>Close HIPPY Application:</td>
<td>Returns to Microsoft Access Start Up Window</td>
</tr>
<tr>
<td>Blank Entry Forms:</td>
<td>Lists the Data Entry/Input Forms</td>
</tr>
</tbody>
</table>
6.3.3 Entering and Editing Information

When the construction manager receives the product information from the design team, generally in paper format, the 'starting point' is to classify the information with regard to the data entry levels. For this reason, the structuring of the forms is designed in a similar hierarchical format to that of the information models, so as to enable data entry in accordance with the information levels represented by the integrated product and process model (described in Chapter 4). The following sections demonstrate through the use of an example ('Construct House Substructure') how data between product and process information is integrated using HIPPY. It also describes how information is entered and edited in the product and process forms.

6.3.4 Forms for Integrating Process and Product Information

The structuring of the forms is an iterative process, which involves the product and process information being categorised into a hierarchy from general to specific (i.e. the first form defines the most general top-level process activities or product elements, and are further decomposed into greater levels of detail). Thus, all other forms serve to define the information needed to convert a structured set of activities and resource elements (actors, equipment, specifications, components, etc) into the completed top-level process, in order to produce the specified output or end product; the 'constructed house substructure'.

6.3.4.1 Inputting Information in Top-Level Process Forms

An example of the kind of form used for entering, viewing, and editing of graphical modelling data about the top-level process activities is illustrated in Figure 6.3. The labels and object control buttons are designed to be self-explanatory (for example, when a user moves the mouse pointer over a control in the form, a message appears specifying a control tip, which has been attached to that particular label or control). Also, users are provided with further instructions or explanations on the status bar at the bottom of the form, whenever a text box or control button receives the focus. This feature applies to all forms in the HIPPY application.
At the top of the form are text fields for entering the process ID, process name, node, description, and process model data. The top-level process identification number (in this case ‘1’) for the example under consideration is automatically generated when a new entry is made in the other fields (process name, node, etc). However, because of the relationship between the top-level process and the other fields, data entry in at least one field will need to be specified, otherwise a new entry for process ID will not be saved (this also applies to all other forms in the HIPPY application). In addition to entering new data that best describes a process activity name of the proposed house facility, process names can also be selected from the list in the drop-down combo box; when clicked, all related data is automatically revealed in the associated fields. There are also provisions for the user to describe what the process activity is about and a direct field link to view or create the activity diagram.

Figure 6.3: Top-level process information form
Clicking on the process model frame (Figure 6.3) activates the OLE automation mechanism (described in Section 5.6.4) between the different applications of the prototype system. This enables the user to edit, change, update, delete or develop a new diagram within the software modelling tools original source file. It also enables the user to access a variety of product and process information from various other data sources, on account of the multiple hyperlinks connected (attached) to the diagram components (see Section 4.4.2.7). It is appropriate to mention that when the user activates this activity by clicking in any of the model fields, the diagram automatically opens the original source file, which also contains hyperlinks to other applications and information source files. This enables the user to view or edit various product and process information from a number of application sources. However, any changes made to these files, will automatically change (or update) the connected data fields within the HIPPY application.

The subform, illustrated at the bottom defines the decomposed main activity stages (or sub-activities) and the diagrams associated with the top-level process. Data entry is carried out in a similar way (in accordance with above), although much of this information can be generated automatically when data is entered in the second-level process forms. For example, the user enters data in the subform process '1' record (highlighted in Figure 6.3), clicks on the 'review process activities' button, which automatically links the user to the second-level sub-process form (Figure 6.4), where the sub-process activity information (highlighted in the subform record in Figure 6.3) can be viewed or edited as appropriate. Double clicking on the activity model frame also allows the user to edit the diagram such that when the diagram is modified, the embedded diagram is also modified automatically. This rule or function applies to all forms and subforms in HIPPY with diagrams, or hyperlink source functions connected to them.

All forms also contain navigation controls, print, save, add, delete or find facilities, and various command buttons allowing the user direct access to enter or view product data forms. Entry for the next process is initiated by clicking on the 'next record' button, or by using the Access Windows menu at the top of the form. To proceed with the processing (or data entry) of process information, the user will have to open the second-level sub-process form, illustrated in Figure 6.4. This enables process information to be decomposed and entered in finer levels of detail.
6.3.4.2 Inputting Information in Top-Level Product Forms

In order to automate data exchange between the top-level product and process information forms, clicking the ‘review product data’ button (Fig 6.3) automatically activates the event procedure, linking the user to the top-level product form. An example of the kind of form used for entering, viewing, and editing graphical modelling data about the top-level products is illustrated in Figure 6.5. Inputting product information follows a similar approach to that described for the process forms, whereby the first form defines the most general top-level product, while the remaining forms are decomposed into finer levels of detail, and define the information needed to convert building elements, components and sub-components into a completed product. The labels and object control buttons are also designed to be self-explanatory, and users are provided with similar facilities for adding, deleting, saving, printing and navigating between records and process information forms.
Figure 6.5: Top-level product information form

The relationships between text fields are generated in a similar way to that described for the top-level process forms. Provisions are made for the user to describe what the product is about and a direct field link to create or edit the diagram relating to that product. There is also a facility where the user can view a picture or image of the product. This is activated when the user clicks in the picture hyperlink text field, allowing the user to gain insight into what the product looks like. Double clicking on the product model field also provides the user with similar benefits to that described for the process form (e.g. access to a variety of product and process information from various other data sources), on account of the multiple hyperlinks, which have been connected to the diagram components.

To enter, edit or view more detailed product information, the user selects a sub-product record from the sub-form (Fig 6.5) and clicks on the ‘review sub-products’ button. The
associated sub-product data of the top-level product (defined earlier in the sub-form) is displayed in the top part of the ‘sub-product’ information form, all connected building elements are also listed in the sub-form below, as illustrated in Figure 6.6. The features, functions, and object command buttons enable the user to automate data exchange between software tools and applications, document files, Web pages, and to integrate between “all” product and process information forms in a hierarchical manner.

Figure 6.6: Form for structuring sub-product information

The structuring of product information follows a similar pattern to that described for the process information forms, whereby the product forms describe the building elements and components (substructure and foundation) needed to construct the house substructure foundation described by the process forms. The process for entering data is carried out in a hierarchical manner, until all the product and process information is entered in the appropriate forms.
6.3.5 Inputting Information in other Product and Process Forms

The following sections describe the various other forms within the HIPPY application, with a view to describing how information is entered, editing, and represented, by means of the 'construct substructure' example. However, it has been established that clicking on the appropriate 'view' or 'link' buttons, activates data exchange between product and process forms; thus, providing the user with options for entering, viewing and editing information in a hierarchy manner. Therefore, for simplicity, a demonstration will be carried out separately, so as to enable a consistent representation of the forms to be made.

6.3.5.1 Sub-process Activity Forms

These forms are used for entering, viewing or editing information in more detail in relation with respect to the top-level process. When activated, the user can enter data or click on the top process 'combo box' to view a list of available top-level processes, which are automatically generated from entering information in the previous top process form. Inputting sub-activity data is similar to the procedure described for the top process form, although, the relationship this time is between the sub-activity ID (this also applies to all ID phrases) and the name or node text labels, as illustrated in Figure 6.4. The five main stages of the top process 'construct house facility' shown in the activity model frame (and also highlighted in the process information subform) are entered or viewed in the decomposed function box sub-form. When the user selects an activity (e.g. the highlighted function box A3 record) in the subform, the colour of the function box displayed in the activity model field automatically changes. This function allows the user to identify the information and hyperlinks associated with this activity, so as to enables further decisions (if needed) to be made on whether to change or update existing information links. Clicking on the 'next record' bottom automatically navigates the user to that function box, highlighted in the activity model frame, and also illustrated in Figure 6.7.

This procedure allows for the selection of sub-activity information, which the user can either view or edit as appropriate. The user can also click in the function model field and enter (or hyperlink) a new diagram file name or edit the activity diagrams in the original source file. Clicking on any of the diagram connections within the original source file enables the users to access, view and edit product and process information from a variety of application sources. However, changes made to these files will automatically change the
connected fields within the HIPPY application. Clicking on the ‘view product data’ button also enables the user to enter or edit product data relating to the sub-activities on this form.

6.3.5.2 Structuring Functions and Activity Durations

The sub-activity form, illustrated in Figure 6.7, exhibits the selected ‘construct substructure’ record, highlighted in Figure 6.4, and activated by clicking on the ‘next record’ navigation button. Repeating the same process (e.g. selecting the ‘construct foundation’ A33 record highlighted in Fig 6.7) allows data entry for the ‘construct foundation’ function box, displayed in the function box information form (Figure 6.8), and also activated by clicking on the ‘review function box’ button, or by using the Access menu bar at the top of the form.

![Figure 6.7: Sub-activity form activated by clicking the ‘next record’ button](image)
When the user enters or selects a function box (e.g. construct foundation), the 'third-level' function box forms display this information in such a way that the 'construct foundation' function model is further decomposed to only show a single function box diagram. All associated information (e.g. ICOM attributes, arrows, data links, etc.) is entered, viewed or edited in the function box ICOM subform located at the bottom of the function box information form, so as to identify the requirements in accordance with the design constraints. The function box information form (Figure 6.8) is designed to allow the user more freedom in entering, viewing and editing process information.

![Figure 6.8: Function box information form](image)

The process activity name at the top of the form is generated automatically through a parent/child relationship, which all forms contain. Previously defined process activity names are also displayed in the same process activity drop down list box, together with the function box ID number and other associated function box information, which is displayed in the function box subform. Data entry is similar to the procedure for the previous forms,
whereby the user selects from the available drop down combo list boxes, or enters the appropriate phrases that best describes the function boxes. The user also has the option of entering the start and finish times of each activity function. This extra feature provides additional information with respect to the duration of activities. As a result, generating primary knowledge (e.g. possible size or importance of activities) for when decisions or changes have to be made at a later stage. The ICOM attributes (arrow names) describe the information needed to perform the top-level activities (previously described in the other forms), and are entered in the ICOM information sub-form at the bottom (Fig 6.8). The user can also view graphical or document files containing additional product and process information from the 'data link' fields, or enter a new hyperlink to a diagram or document file (this also includes links to other sources, such as Internet and Web files).

It is also important to mention that other third-level forms have been developed specifically for inputting, finding or replacing activity ICOM information. These forms allow a profile relating to the diagrams interconnections and hyperlinks to be assembled, which are stored in the application database and accessible from the appropriate form.

6.3.5.3 Structuring of ICOM Information

The third-level forms in Figure 6.9 and 6.10 are used for structuring ICOM attributes (i.e. resources, controls, mechanisms, and outputs), arrow interconnections, and hyperlinks to associated relationships with regards to the function box models (illustrated in Fig 6.8), and have been specifically designed to allow information to be entered in more detail. The most general attributes are the ICOMs, and these are further decomposed into finer levels of detail with respect to resource materials, construction team, equipment, plant, and activities performed or completed. Clicking on the 'find record' button can access additional features, which might be useful in structuring the ICOM information. This option helps the user to find or replace a record stored within the application. However, once a record or phrase is replaced, all connected information is altered to reflect the change. These forms can be opened by clicking on the appropriate 'view ICON' button, or from the Window menu bar at the top of the forms. Clicking in the hyperlink field box (Fig 6.10), also allows the user to view or edit valuable document files with respect to the ICOM attributes, stored within other applications.
Chapter 6

Figure 6.9: Form for ICOM attributes

Figure 6.10: Form for structuring activity ICOM arrows (connection relationships)
The final fourth-level form (Figure 6.11) has been designed for input, viewing and editing ICOM attributes, entered previous in the ICOM forms. This form is the last in the series of process forms, and combines the ICOM information from the third-levels forms. The ICOM type ID and attributes of the selected diagram (Fig 6.8), and defined earlier, are displayed in the top part of the form, as illustrated in Figure 6.11. The form also contains two sub-forms for the input, viewing and editing even more detailed information about the ICOM attributes.

Figure 6.11: Form for structuring types of ICOM attributes

The first sub-form provides for ICOM attributes generated in the ‘ICOM information form’ (Figure 6.8), and is directly linked to the second sub-form, which provides for the entry of more detailed ICOM arrow names and relationships, such as a specific plant item, or team member. To facilitate the structuring process, new information can be introduced in the ICOM fields, or existing ones rephrased, and stored for reuse at a later time in the processing function. The decomposition from general (ICOMs) to specific (ICOM arrows)
is based on the ability of the lower sub-form to enhance the implementation of the higher sub-form. On the other hand, the first sub-form provides justification for the second sub-form. For example, when a user selects an ICOM record field in the first sub-form (highlights in Fig 6.11), a list of the ICOM arrows required to perform the selected activity function is generated automatically in the second sub-form. The user can then view or edit the ICOM arrows and other data links associated with the selected record. The user can also preview the selected activity function fields by double clicking the 'preview report' button or by printing the record currently displayed in the form.

6.3.5.4 Sub-product Element Forms

These forms are used for structuring building elements and components in more detailed, and follow a similar approach for entering information to that described for the process forms. The types of building elements form (Figure 6.12) shows the decomposed sub-product information 'substructure', and is opened by clicking on the 'view element type data' button (Fig 6.6), or from the function box process form (Fig 6.7).

![Figure 6.12: Form for structuring building element types](image-url)
The top section of the form contains previously defined sub-product information (defined in Fig 6.6). It also provides hyperlink fields to allow the user to view or edit pictures/images (types of substructures or foundations for example) of the sub-products (building elements), which can be used to determine the dimensions of the building element types, illustrated in the top subform. The top subform provides the text fields for entering the building element dimensions (e.g. height, length, width), and is directly linked to the second sub-form, which enables the user to enter more detailed information with respect to the various building material components that are associated with the building element. Entry for the next building element (e.g. footing, foundation slab, etc) is initiated by clicking on the ‘next’ button, or by using the Access navigation controls at the bottom of the form. Selecting a hyperlink field record, or clicking on the appropriate view buttons, also provides access to other supplementary information, such as links to product and process forms, document files and images or pictures. To enter, view or edit the element quantities, and building components records in more detail (Figure 6.13), the user clicks on the ‘review building element’ button at the bottom of the element types form (Fig 6.12).

![Figure 6.13: Decomposed building elements and their measurements](image-url)
The top part of the ‘Element Form’ (Figure 6.13) displays the product information with regard to the element types (generated automatically from Fig 6.12), the interrelated dimensions and calculated quantity of the element name, and other information associated with the building element (e.g. foundation). The user can either view the existing information or enter new values in the appropriate dimension fields. However, changes made to any of the numbers in the dimension fields will automatically change the information previously entered in the element type sub-form, and also the total quantity displayed in the ‘total amount’ number field. Although, once the user saves this information, changes to the data will automatically be updated and saved to the original database source. The top sub-forms contain a list of the decomposed building components (entered in the previous form), and is directly linked to the second subform, which contains a list of the different types of each material component that make-up the ‘element name’ (e.g. strip foundation), which in turn, is part of the element type (foundation). Clicking the ‘next record’ button or selecting the ‘brickwall footing’ element from the drop down list box, initiates entry to the next building element. The building element information form also provides the user with direct links to other product forms (Fig 6.14 & Fig 6.15), which have been specifically designed for entering building components and types of components. Inputting information follows a similar procedure to that described for the third-level process forms (Section 6.3.5.3), whereby the user enters information regarding the building components in more detail.

![Building Component Date Entry Form](image)

*Figure 6.14: Forms for inputting building component information*
The form in Figure 6.14 is specifically used for inputting and finding or replacing building components, whereby the form in Figure 6.15 is used for structuring the components types, and is interconnected with the building element form. That is, the information entered in this form is displayed in the associated form.

![Image](image_url)

**Figure 6.15: Forms for structuring types of building components**

### 6.3.6. Generating Product and Process Reports

Other forms for entering, viewing and editing product and process information provide similar features and user interaction. Figures 6.16–6.18 show the reports for: integrated relationships of product and process form, process activity function boxes (activity, ICOM, arrow connection, name, etc), and top-level building product, sub-product, and the building elements dimensions associated with the product ID. These reports can be printed or previewed by clicking on the ‘report’ button on the appropriate forms, or by highlighting the records in any sub-form, and clicking the ‘go to main interface’ button and previewing the information selected from the preview reports section.
Chapter 6

Integrated Relationships of Product and Process Forms

Figure 6.16: Integrated product and process relationships report

Activity Function Boxes Report

<table>
<thead>
<tr>
<th>ID</th>
<th>Function Box ID</th>
<th>Function Box Name</th>
<th>Start Date</th>
<th>Finish Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>M</td>
<td>Construct Organization</td>
<td>12/1/2001</td>
<td>6/1/2001</td>
<td>Resources are converted into completed elements of the designed superstructure activities</td>
</tr>
<tr>
<td>10</td>
<td>TAO</td>
<td>Construct House Facility</td>
<td>12/1/2001</td>
<td>6/1/2001</td>
<td>Top-level activity box defining the scope and external ICOM interface connections of the top-level construction process.</td>
</tr>
</tbody>
</table>

Figure 6.17: Report structure for the top-level function box
Figure 6.18: Top-level product information report

Figure 6.16 presents the relationships between the product and process forms into a hierarchy from general to specific structure. Clicking on the ‘preview relationship’ button in the main interface opens the report. The report in Figure 6.17 shows the top-level building product, sub-products, and building elements associated with the ‘product ID’, and can be opened by clicking the ‘preview print report’ button (Fig 6.13). The report in Figure 6.18 displays the top-level function boxes information (Fig 6.3), and can be viewed by clicking on the appropriate process ‘report’ button. Only a few reports have been generated here, so as to demonstrate how the Product and process information can be reviewed. Complete lists can also be previewed or printed by following the same procedure previously described (clicking on the appropriate report functions).

6.3.7 Integrated Product and Process Information Reports

The report in Figure 6.19 displays both process and product information. It was specifically designed for integrated product and process information, by attaching VBA code to specific product and process data fields (highlighted in black) ‘only’. It is generated by selecting product and process information from the appropriate subforms, clicking on the ‘go to main interface’ button and previewing the integrated report from the report section. Alternatively, the user can follow a similar procedure by selecting the
product and process data, and clicking on the 'preview integrated data' button, illustrated in Figure 6.12, or by opening the Access Windows control menu at the top of the forms. However, it is appropriate to mention that only a few selected forms and data fields have been developed (using macros and VBA code) to support the integration of product and process report building, and are used here only to demonstrate the output (integrated product and process reports) of the HIPPY prototype. A full report generation module could be introduced, which would not only integrate text but also diagrammatic representation of the selected data. In addition, all product and process forms could contain facilities for previewing and printing the displayed data; users can also print the report directly from the previewed window.

<table>
<thead>
<tr>
<th>Function/Box Name</th>
<th>Start Date</th>
<th>Initial Date</th>
<th>Element Type</th>
<th>Element Name</th>
<th>Length (m)</th>
<th>Height (m)</th>
<th>Width (m)</th>
<th>Amount</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contract Foundation</td>
<td>03/12/91</td>
<td>19/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Contract Foundation</td>
<td>03/12/91</td>
<td>19/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Contract Turnout Fac</td>
<td>03/12/91</td>
<td>19/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Contract Substructure</td>
<td>28/1/91</td>
<td>1/2/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Contract Substructure</td>
<td>28/1/91</td>
<td>1/2/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Contract Supervisors</td>
<td>28/1/91</td>
<td>1/2/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Report &amp; Approve Work</td>
<td>14/12/91</td>
<td>19/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Report &amp; Approve Work</td>
<td>14/12/91</td>
<td>19/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Lay Underground Service</td>
<td>09/12/91</td>
<td>14/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Lay Underground Service</td>
<td>09/12/91</td>
<td>14/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Organic Site Resources</td>
<td>13/1/91</td>
<td>18/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Organic Site Resources</td>
<td>13/1/91</td>
<td>18/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Plan of Work Schedule</td>
<td>28/1/91</td>
<td>31/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
<tr>
<td>Plan of Work Schedule</td>
<td>28/1/91</td>
<td>31/12/91</td>
<td>Foundation</td>
<td>Strip Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
<td>Mfbd.</td>
</tr>
</tbody>
</table>

Figure 6.19: Report displaying the integration of product and process information

This additional feature provides the end-users (construction managers) with a 'new' facility for integrating, viewing, editing and printing product and process information concurrently, and therefore, helping to clarify their vision of the facility to be constructed. It also provides an important step towards integration of the heterogeneous product and process information sources within the construction industry. By integrating product and process information the user stands to benefit from a consistent representation of the on-site construction process. The generation of product and process data reports concludes the processing of product and process information for on-site construction.
6.4. DISCUSSION

The previous sections have described the way in which product and process information can be generated to facilitate the integration of product and process reports using HIPPY. The tedious calculation of building element quantities is performed automatically by attaching VBA code. The user is provided with adequate guidance throughout the example by means of text labels and messages specifying control tips on how to enter and process information without the need to have mastered the methodology on which HIPPY is based.

Although the product and process information for a house-building was used as a test bed, HIPPY can also be used for other projects types, as there is enough flexibility for the end user (i.e. construction managers) to input information (e.g. diagrams, graphic images, document files, etc), which is not necessarily geared towards a house-building. Furthermore, the way in which the models have been integrated with the enabling technology provides a template, allowing end-users to enter their own constraints, parameters, etc. Also, most of the information can be reusable for other projects, given that the data has been stored within the database application.

However, since the information was generated within a research environment, it would be necessary to validate the practical application of HIPPY on a real construction project. As this was not possible, due to research constraints, a selection of researchers and industry practitioners, who represent some of the potential end-users of the prototype software, were invited to evaluate the prototype system.
6.5 SYSTEM EVALUATION

This section describes how industry practitioners and researchers evaluated HIPPY. The prototype system described in the previous section was used for the evaluation. An analysis of the evaluation questionnaire was undertaken. The suggestions, recommendations and results of the evaluation by practitioners are reported here.

6.5.1 Evaluation Objectives

The objectives of the evaluation were:

1. To assess the effectiveness of HIPPY in integrating product and process information.
2. To demonstrate that the prototype has achieved the aim of this study as outlined in Chapter 1.
3. To demonstrate that the prototype system effectively provides the functional and non-functional requirements described in Section 5.2.
4. To assess the ease with which HIPPY can be used.
5. To obtain comments and recommendations to guide further developments.

6.5.2 Constraints on the Evaluation

The major constraint on the evaluation was 'time'. It would have taken too long to travel round several industrial practitioners, but one session at the University would reach several at once. It was therefore decided that industry practitioners should be invited to observe the operation of HIPPY at the university. This limited the choice of evaluators (to local practitioners) who were available to participate in a demonstration of the prototype system. Therefore, to assist in the evaluation, a selection of researchers who have a good working knowledge of the construction and manufacturing industries were also invited to participate in the evaluation.

6.5.3 Evaluation Approach

Although various evaluation techniques are available, for example: heuristic methods based on reason (such as, white box and black box testing), test case design or control structure methods (such as, condition testing, data flow and loop testing), and methods for
specialised environments and applications, such as testing GUI, client/server, and real time systems (Pressman, 1997; Davis, 1995; Beizer, 1990; Myers, 1979). However, as computer software systems become more complex, the need for specialised testing approaches has grown. Typical evaluation in the area of integrated product and process systems has often focused on *ad hoc* user studies, since most integrated applications are for specialised areas (for example, integrated project databases for assisting project managers with the building services, costing and planning, procurement and detailed design stages, or applications assisting facility managers or managing construction changes). The HIPPY application is also in a specified field and not a wide spectrum of expertise is readily available to evaluate the prototype within the given time-frame. Therefore, the evaluation of HIPPY focused on techniques that were related to prototype testing and specialised computer environments and applications, since the prototype system has been structured to include various requirements and features associated with this domain, such as GUI and end users.

However, a good evaluation strategy should include tests conducted by both the designer and end-users. Evaluations conducted by the users of the system provide good feedback, whereas, designers of the system can also conduct various tests that help in identifying usability and reliability problems. This is the strategy that was considered to offer the best approach for achieving the evaluation objectives. Therefore, it was decided that potential end-users of the software prototype needed to see a live demonstration of its use. They would then be requested to complete a questionnaire, which will allow them to indicate their opinion on the various aspects of the software being evaluated. The procedure adopted is outlined below.

### 6.5.3.1 Evaluation Procedure

The evaluation proceeded through the following stages:

1. **Initial Evaluation:** Initial user feedback at any early stage of the systems implementation was important, this was obtained through comments and suggestions from other researchers and academic staff involved with the research. It was not always practical and possible to get industry practitioners to be involved in the initial assessment of the developing prototype; that would have required them to make frequent visits to the University.
2. **Self Evaluation**:- A self-appraisal of the prototype system was also performed by the researcher, using the available features and facilities of the software development tools, which contributed to the actual development of the HIPPY application. A more detailed discussion of the system from a research perspective is included at the end of this chapter.

3. **Final Evaluation**:- This evaluation was carried out at the final stage of the system implementation and involved a group of eighteen participants from industry. Seven industry practitioners representing the construction and manufacturing industries, and eleven researchers who had been selected because of their industrial experience. Each evaluator attended a short presentation about the development, implementation and operation of HIPPY. This was followed by a live demonstration (similar to the example previously discussed in the operational section) to show how the system works. The evaluators were then asked to complete a questionnaire relating the demonstration. To ensure that they knew what to look for, the questionnaire was placed on an overhead projector (acetate) after the demonstration, and an open discussion (comments) about the system and questions followed. It should be mentioned that several researchers were individually taken through the operation of HIPPY; this was due to the time and date of the demonstration. They therefore had the opportunity to see a more detailed demonstration on how the system operated.

### 6.5.3.2 Background of Evaluators

The evaluators consisted of two groups: 7 respondents who are currently working in the industry and 11 University researchers who have previous working experience of the industry. Tables 6.2 and 6.3 present the details of the evaluators and some information relating to their organisations and current status.

The participants from industry consisted of one construction software developer, two civil engineers, three construction managers, and one manufacturing engineer. Between the industrialists they have a total of 91 years with an average of 13 years. The organisations they represent have an average of 8,886 employees, and an average annual turnover of £38m.

In contrast, the researchers consisted of a broader range of respondents to include two project and three construction managers, one software engineer, one civil engineer, one
transport engineer, one lecturer of construction management, and two researchers in construction management. The researchers had a total of 113 years working experience with an average of 10 years, and a research experience average of 4.4 years.

**Table 6.2: Details about Industrial Evaluators and their Organisations**

<table>
<thead>
<tr>
<th>Position</th>
<th>Experience (years)</th>
<th>Type of Firm</th>
<th>Number of Employees</th>
<th>Annual Turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co'tion Software Developer (1)</td>
<td>15</td>
<td>Manufacturing Practices (Consultants)</td>
<td>25,000</td>
<td>£70m</td>
</tr>
<tr>
<td>Pro/Construction Manager (3)</td>
<td>37</td>
<td>Construction (Contractors) &amp; (RD &amp; T)</td>
<td>26,000</td>
<td>£100m</td>
</tr>
<tr>
<td>Manufacturing Engineering (2)</td>
<td>10</td>
<td>New Product Development (Suppliers)</td>
<td>400</td>
<td>£10m</td>
</tr>
<tr>
<td>Civil Engineer (2)</td>
<td>37</td>
<td>Consulting Engineers (Consultants)</td>
<td>1,800</td>
<td>£85m</td>
</tr>
<tr>
<td>TOTAL Experience (years)</td>
<td>91</td>
<td>TOTAL Employees/Turnover</td>
<td>62,200</td>
<td>£265</td>
</tr>
<tr>
<td>Average Experience (years)</td>
<td>13</td>
<td>Average Employees/Turnover</td>
<td>8,886</td>
<td>£38m</td>
</tr>
</tbody>
</table>

* RD & T = Research Development and Technology

**Table 6.3: Details about Researchers and their Previous Organisations (if applicable)**

<table>
<thead>
<tr>
<th>Background</th>
<th>Experience (years)</th>
<th>Type of Firm</th>
<th>Research Area</th>
<th>Research (years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Management (2)</td>
<td>25</td>
<td>Building/Construction</td>
<td>Construction</td>
<td>12</td>
</tr>
<tr>
<td>Construction Managers (3)</td>
<td>28</td>
<td>Civil Eng/Construction</td>
<td>Civil Engineering</td>
<td>14</td>
</tr>
<tr>
<td>Software Engineer (1)</td>
<td>10</td>
<td>Manufacturing Software</td>
<td>Construction</td>
<td>5</td>
</tr>
<tr>
<td>Civil Engineer (1)</td>
<td>9</td>
<td>Consulting Engineers</td>
<td>Civil Engineering</td>
<td>3</td>
</tr>
<tr>
<td>Transport Engineering (1)</td>
<td>15</td>
<td>Civil Servant (council)</td>
<td>Transport Studies</td>
<td>3</td>
</tr>
<tr>
<td>Teaching &amp; Education (1)</td>
<td>12</td>
<td>Teaching &amp; Research</td>
<td>Construction</td>
<td>4</td>
</tr>
<tr>
<td>Research Assistant (2)</td>
<td>14</td>
<td>Academic (only)</td>
<td>Construction</td>
<td>7</td>
</tr>
<tr>
<td>TOTAL Experience (years)</td>
<td>113</td>
<td>TOTAL Academic Research Experience (years)</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>Average Experience (years)</td>
<td>10</td>
<td>Average Research Experience per person (years)</td>
<td>4.4</td>
<td></td>
</tr>
</tbody>
</table>

Although the evaluators were not selected at random, they are however, fairly representative of the potential end-users of the system, in that they have management experience in, or have been involved with manufacturing or construction projects. The experience of the evaluators, and their nature of their positions, was also considered to be adequate to enable a fair assessment of the prototype system.

**6.5.4 Test Case for Evaluation**

To enable the evaluators to get a better understanding of the integrated information modelling system, they were first given a short presentation on the concepts, system requirements and integrated information architecture of HIPPY. After the presentation, a test case that shows the use and structure of the system was demonstrated. The actual
building products and processes for a house project were used, as an example to demonstrate the system's operation and functionality. This was followed by a discussion relating to the system, whereby the evaluators openly asked questions relating to the demonstration.

6.5.5 Review of the Evaluation Questionnaire

After the demonstration session, a questionnaire was then presented to the evaluators to complete. The questionnaire was designed so that HIPPY could be evaluated against the requirements for integrating product and process information based on CE principles to support the management of information during on-site construction, as well as the specific requirements for the generic information models. It was divided into three sections as follows:

1. Section A requested information about the evaluator's organisation, and about their position and experience.
2. Section B contained 25 questions about various aspects of HIPPY, grouped into the following five sub-heading:
   - definition of product and process information;
   - applicability to the construction industry;
   - management of the system;
   - efficiency of teamwork; and
   - general section.
3. Section C requested comments on ways to improve the system, and for any other general comments.

For each question, respondents were asked to express their opinion on the effectiveness of HIPPY on a scale of 1 (poor) to 5 (excellent). A sample of the questionnaire is provided in Appendix 1. The sub-heading of the questionnaire reflect the aim and objectives of the research by focusing the evaluators on what is the importance of the study and where they should focus their attention. By so doing, other areas of interest by the evaluators were considered under the general headings, such that the contributions of the evaluators would also be assessed against the overall performance (effectiveness) of the prototype environment. The results to the questions, suggestions, comments, recommendations, strengths and limitations are presented below.
6.5.6 Overview of Results

The performance of HIPPY was adjudged to be satisfactory. The rating of the questions in the questionnaire showed that HIPPY could adequately perform the function for which it was designed. All the respondents were generally satisfied with the effectiveness for integrating product and process information, in being applicable to the needs of the industry, facilitating communication and teamwork, and in its general operation.

Table 6.4 presents the average ratings (results) on the effectiveness of HIPPY with respect to the specific questions in the questionnaire. The table also shows the individual results between the industry respondents 'I. Avg.', the researchers 'R. Avg.', and the combined total 'T. Avg.'. The rational, was to enable a comparison to be made between the evaluators responses to specific questions, (for example, how confident are the evaluators with computer in general, how easy is the system to use, how convinced that the system will be used in industry), and to provide justification of their experience and current positions. Furthermore, it would help in facilitating a fair assessment of the software, to validate the actual structure (weighting) of the questionnaire, and to monitor any biasness or misrepresentation. That is, to act as a check against the evaluators responses to questions.

Most of those who took part in the evaluation were also impressed with the amount of information contained within the prototype, with respect to the representation of product and process information, diagrams, links to other application sources, and the overall structure of the system.

It is evident that most areas (e.g. facilitate the definition of product and process information, integration of product and process information, communication among team members, speed of information flow, and the overall rating of the system), the effectiveness of HIPPY was given an average rating of at least 3.6 out of 5. This basically means that the effectiveness total 'T%' rating of HIPPY was 75% or over, with a highest total average rating of 4 out of 5 (80%), and a lowest rating of 3.12 out of 5 (62%). Although, the individual results of the researchers responses to questions did reveal slightly higher averages than the industry practitioners. This was probably due to fact that many researchers had the opportunity to see a more detailed demonstration on how the system operated because of the time constraints.
### Table 6.4: Responses to Questions

<table>
<thead>
<tr>
<th>HIPPY Evaluation Questions</th>
<th>Ranking (out of 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition of Product and Process Information</strong></td>
<td></td>
</tr>
<tr>
<td>1. How effectively does the system facilitate the definition of product and process info?</td>
<td>4.00 4.00 4.00 80%</td>
</tr>
<tr>
<td>2. How effectively does the system help in understanding the flow of processes?</td>
<td>3.57 4.20 3.94 79%</td>
</tr>
<tr>
<td>3. How effectively does the system help in understanding the product components?</td>
<td>3.86 4.00 3.94 79%</td>
</tr>
<tr>
<td>4. How well does it ensure that the product and process information is represented?</td>
<td>3.71 3.80 3.76 77%</td>
</tr>
<tr>
<td>5. How effectively does the system support the integration of product and process info?</td>
<td>3.86 4.00 3.94 79%</td>
</tr>
<tr>
<td>6. How well does the system help in clarifying the browser interfaces between components?</td>
<td>3.29 3.60 3.47 70%</td>
</tr>
<tr>
<td>7. To what extent are the defined product and process info unambiguous?</td>
<td>3.43 3.40 3.41 68%</td>
</tr>
<tr>
<td>8. How effectively is the product and process information structured within HIPPY?</td>
<td>3.43 4.10 3.82 76%</td>
</tr>
<tr>
<td><strong>Applicability to the Construction Industry</strong></td>
<td></td>
</tr>
<tr>
<td>9. How appropriate are the application development tools used in the system?</td>
<td>3.71 3.90 3.82 76%</td>
</tr>
<tr>
<td>10. How well does the system architecture support the flow of modelling information?</td>
<td>3.71 4.10 3.94 79%</td>
</tr>
<tr>
<td>11. How useful is the system to the overall on-site construction process?</td>
<td>3.43 3.80 3.65 74%</td>
</tr>
<tr>
<td>12. How useful is the system to the construction team?</td>
<td>3.57 3.70 3.65 74%</td>
</tr>
<tr>
<td>13. How effectively will the system increase the speed of the info flow during on-site?</td>
<td>3.71 3.80 3.76 76%</td>
</tr>
<tr>
<td>14. To what extent does it represent an improvement of info systems (suitable is it)?</td>
<td>3.57 3.70 3.65 74%</td>
</tr>
<tr>
<td>15. How convinced are you that the on-site info systems will be accepted (useable)?</td>
<td>3.29 3.00 3.12 62%</td>
</tr>
<tr>
<td><strong>Management of the System</strong></td>
<td></td>
</tr>
<tr>
<td>16. How well organized (designed) is the system architecture?</td>
<td>3.86 4.00 3.94 79%</td>
</tr>
<tr>
<td>17. How effectively are changes to the info managed?</td>
<td>3.86 4.00 3.94 79%</td>
</tr>
<tr>
<td>18. How well integrated are the components of the system?</td>
<td>3.86 4.10 4.00 80%</td>
</tr>
<tr>
<td><strong>Efficiency of Teamwork</strong></td>
<td></td>
</tr>
<tr>
<td>19. How well does the system facilitate communication among members?</td>
<td>3.57 3.60 3.59 72%</td>
</tr>
<tr>
<td>20. How effectively will it facilitate a common understanding of the info among team?</td>
<td>3.43 3.90 3.71 74%</td>
</tr>
<tr>
<td>21. To what extent would the output of the system facilitate design or creativity?</td>
<td>3.00 3.60 3.35 67%</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>22. How easy is the system to use?</td>
<td>3.57 3.70 3.65 73%</td>
</tr>
<tr>
<td>23. How generic do you consider the system to be?</td>
<td>3.71 4.00 3.88 78%</td>
</tr>
<tr>
<td>24. How confident are you with computer info systems in general?</td>
<td>4.00 4.40 4.24 84%</td>
</tr>
<tr>
<td>25. What is your overall rating of the prototype system?</td>
<td>3.57 3.90 3.76 77%</td>
</tr>
</tbody>
</table>
The results to the evaluation questions are as follows:

The 'total' highest average rating of 4 (i.e. 76–85% rating) was assigned to HIPPY for the following areas:

1. the extent to which the definition of product and process information is facilitated;
2. the extent to which the system helps in understanding the flow of processes;
3. the extent to which the system helps in understanding the flow of product components;
4. does it insure that the product and process information is represented;
5. the extent to which the system supports the integration of product and process information;
6. the extent to which the product and process information is structured within HIPPY;
7. the extent to which the application development tools are used in the system;
8. the extent to which the system architecture support the flow of modelling;
9. the extent to which the system increases the speed of the information flow;
10. the extent to which the system architecture is organised;
11. the effective management of changes to information;
12. the extent to which the components of the system are integrated;
13. the generality of the system;
14. the confidence with computer information systems in general;
15. the overall rating of the prototype system.

The total average rating of 3.5 or 3.75 (i.e. 70–75% effectiveness) was assigned to the following areas:

1. the effectiveness in clarifying the interfaces between products and processes;
2. the importance of the system to the overall on-site construction process;
3. the value of the system to the construction team;
4. the suitability of the system to existing on-site information systems;
5. the efficiency for communication data amongst members of the construction team;
6. the effectiveness for generating insight to information among the construction team;
7. the ease of use of the system.
The lowest average rating of 3 or 3.25 were in the following areas:

1. the extent to which defined product and process information is unambiguous;
2. the extent that on-site construction professionals will use the system;
3. the extent to which the output of the system would facilitate construction creativity.

The low rating in these three areas was probably due to less-than appropriate wording, or varying interpretation of the question. For example, although ‘will people use the system’ received a relatively low rating, the previous questions in the same category and the management category received the highest rating of 4. Also, the ‘facilitate team creativity’ received a rating of 3.35 (67%), but the ‘facilitating common understanding’ and ‘communication’ received a rating of 3.60–3.90 respectively. This can be explained by the fact that although HIPPY was considered to be of ‘benefit’ to construction team, its ‘usefulness’ was more pronounced to the processing of construction information, which is usually generated by site management. However, relating to the ‘creativity’ question, only the industry practitioners shared this view, because they have a more hands-on view of the construction process.

The results also reflected the differences between both groups’ current positions. For example, the researchers were more confident with information systems, and the development of the system architecture in general. This could be for a number of reasons, such as education background of respondents or the level of information technology on construction sites. Nevertheless, the overall average rating between the two groups was only 0.2 to 0.25 (i.e. 5% effectiveness), this was probably due to the fact that researchers are constantly involved with information technology of some form or another, whereas the industry practitioners are not.

6.5.7 Suggestions for Improvement and other Comments

Table 6.5 presents the comments made by the evaluators on ways to improve HIPPY, and on what they generally felt about the system. The suggestions for improvement are mostly focused on the user interface, the end user of the system, and linking the system with a front-end 3D graphical or virtual reality system. This reflects the relatively low average rating the system received with respect to ‘will construction professionals use the system’ and ‘facilitate creativity’. On the other hand, ‘how easy was the system to use’ received a
reasonable score from both groups. This was probably owing to the size of monitor (14") used to develop the system, and for this reason the information presented on it appeared to be clustered in some cases.

Table 6.5: General Comments on HIPPY*

<table>
<thead>
<tr>
<th>Suggestions for Improvement</th>
<th>Other Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• at some stages the presentation/graphics appeared cluttered and complex;</td>
<td>• the system is full of useful data but needs more graphical representations such as visual reality;</td>
</tr>
<tr>
<td>• maintain simplistic inputs and data; make sure system is easy to use at all times;</td>
<td>• where does the system fit with other systems? Can it be set up for each project and price them using class representation (e.g. SMM7, IFC)</td>
</tr>
<tr>
<td>• perhaps clearer definition of user numbers (i.e. present and future);</td>
<td>• introduce a wizard or more held programs</td>
</tr>
<tr>
<td>• minor details with good representation, however would be better if linked to a virtual reality system;</td>
<td>• can it incorporated design decisions (e.g. what, who, where, and why);</td>
</tr>
<tr>
<td>• integrate a 3D front end with sexy user interface;</td>
<td>• thanks for letting me assess the software; I would like to try it on a new project perhaps later next year;</td>
</tr>
<tr>
<td>• ideally, to be fully operational, make it a bit more user-friendly and state the level it is aimed at;</td>
<td>• good initiative work;</td>
</tr>
<tr>
<td>• good for presenting information simultaneously but needs to be more user friendly for first time users, thus needs some simplifications in the way information is presented;</td>
<td>• good system;</td>
</tr>
<tr>
<td>• a useful extension of the system is to make it portable;</td>
<td>• is there a manual?</td>
</tr>
<tr>
<td>• make a clear difference between the use of information and knowledge base.</td>
<td></td>
</tr>
</tbody>
</table>

*Some statements have been slightly edited to facilitate ease of reading.

Additionally, 65% of the respondents stated that the system needed a 3D graphical front end, to enable 3D objects to be individually linked to the information within the system. This could also be explained by the fact that before the presentation of HIPPY, another presentation demonstrating a 3D virtual reality modelling system took place, and many of the comments actually specified the name of the 3D system (VISCON). This could also be considered to reflect the ‘unambiguous of information’ question, whereby the system was originally developed to integrate and manage the flow of product and process information during the on-site construction stage, and not 3D graphical CAD models; although, this is on the agenda for further research. However, other comments by evaluators (for example, ‘nice system’, and ‘good initiative work’) indicated that they were generally pleased with HIPPY. In fact, some of the respondents expressed their desire to use the system on a live project in the near future.
6.5.8 Strengths and Limitations

The key strengths of the prototype system, as revealed from the evaluation include its ability to define and integrate various heterogeneous information sources, to facilitate an understanding of graphical and non-graphical data (e.g. processes and the products needed to support the processes) and to generate links and integrated reports of the various information sources, within a Windows environment. The limitations of the system, as revealed from the evaluation indicated that the prototype would need to incorporate added features (such as links to virtual 3D graphic representation) if construction practitioners would consider using it in real test cases.

6.6 REVIEW OF THE OBJECTIVES FOR HIPPY

The development of HIPPY was aimed at integrating product and process information during the on-site construction stage of SMsBP. The integration of heterogeneous information using concurrent engineering principles (in accordance with the research undertaken in this thesis), can be enhanced by modelling information with respect to the following:

- precisely defined to remove any ambiguities;
- structured in such a way as to make it easy to understand, and correlate changes within the original source files and forms;
- stated in a format that can be understood by different disciplines during the construction stage of a project; and
- be a true representation of the on-site construction process.

It was also determined that the process for achieving these outputs, should:

- maintain focus on the construction managers in the construction team;
- manage IT applications and the representation of products through static modelling, and the dynamic process for describing a proposed facility; and
- incorporate, where possible, CE principles (e.g. team work, upfront consideration of life-cycle issues).
The extent to which these objectives were satisfied by HIPPY is presented in Table 6.6, which links the questions in the questionnaire to the requirements for the effective integration of product and process information to support construction manager during the on-site construction planning and operations stages of the construction process. A calculation of the averages for the ratings for each objective (as illustrated in table 6.6), shows the following scores (out of 5):

- Precise definition of information to remove ambiguities 3.78
- Structured in such a way as to make it easy to trace and correlate changes within the original file source and forms 3.80
- Stated in a solution-neutral format that can be understood by different disciplines during the construction stage of a project 3.72
- Also, reflect a true representation of the building process 3.72
- Maintain focus on the site manager’s construction team 3.39
- Manage IT applications and the representation of products through static modelling, and the dynamic process for describing a proposed facility 4.00
- Incorporate, where possible, CE principles (e.g. team work, up-front consideration of life-cycle issues) 3.77
- Represent an improvement of the existing integration management of product and process information 3.74

\[
\text{Average for all objectives} = 3.75
\]

This score indicates that, although there is room for improvement, HIPPY does provide for the effective management of integrating product and process information to support CE implementation during the on-site construction stage. The evaluation also suggests that HIPPY, and the information models on which it is based, are of benefit to the end-user; that is, the on-site construction managers and the project team, and by implication, to the overall building delivery (construction) process.
Table 6.6: Hippy and the Objectives for the Information Models

<table>
<thead>
<tr>
<th>OBJECTIVES</th>
<th>QUESTIONNAIRE QUESTIONS</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precise definition of information to remove ambiguities</td>
<td>How effectively does the system facilitate the definition of product and process information?</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>How well does it insure that the product and process information is represented?</td>
<td>3.76</td>
</tr>
<tr>
<td></td>
<td>How effectively does the system support the integration of product and process information?</td>
<td>3.94</td>
</tr>
<tr>
<td></td>
<td>To what extent are the defined product and process information unambiguous?</td>
<td>3.41</td>
</tr>
<tr>
<td></td>
<td><strong>AVERAGE</strong> 3.78</td>
<td></td>
</tr>
<tr>
<td>Structured in such a way as to make it easy to understand and correlate changes within the original file sources and forms</td>
<td>How well does the system help in clarifying the browser interfaces between product components and the processes?</td>
<td>3.47</td>
</tr>
<tr>
<td></td>
<td>How effectively is the product and process information structured within HIPPY?</td>
<td>3.82</td>
</tr>
<tr>
<td></td>
<td>How well organized (designed) is the system architecture?</td>
<td>3.94</td>
</tr>
<tr>
<td></td>
<td>How effectively are changes to the information managed?</td>
<td>3.94</td>
</tr>
<tr>
<td></td>
<td><strong>AVERAGE</strong> 3.80</td>
<td></td>
</tr>
<tr>
<td>Stated in a solution-neutral format that can be understood by different disciplines during the construction stage of a project; Also, reflect a true representation of the building process</td>
<td>How effectively does the system help in understanding the flow of processes required for constructing a building?</td>
<td>3.94</td>
</tr>
<tr>
<td></td>
<td>How effectively does the system help in understanding the product components needed to support the processes?</td>
<td>3.94</td>
</tr>
<tr>
<td></td>
<td>How useful is the system to the overall on-site construction process?</td>
<td>3.65</td>
</tr>
<tr>
<td></td>
<td>To what extent would the output of the system facilitate design or construction team creativity?</td>
<td>3.35</td>
</tr>
<tr>
<td></td>
<td><strong>AVERAGE</strong> 3.72</td>
<td></td>
</tr>
<tr>
<td>Maintain focus on the site managers construction team</td>
<td>How useful is the system to the construction team?</td>
<td>3.65</td>
</tr>
<tr>
<td></td>
<td>How convinced are you that on-site construction professionals will accept (or use) the system?</td>
<td>3.12</td>
</tr>
<tr>
<td></td>
<td><strong>AVERAGE</strong> 3.39</td>
<td></td>
</tr>
<tr>
<td>Manage IT applications and the representation of products through static modelling, and the dynamic process for describing a proposed facility</td>
<td>How appropriate are the application development tools used in the system?</td>
<td>3.82</td>
</tr>
<tr>
<td></td>
<td>How well does the system architecture support the flow of graphical (modelling) information?</td>
<td>3.94</td>
</tr>
<tr>
<td></td>
<td>How confident are you with computer information systems in general?</td>
<td>4.24</td>
</tr>
<tr>
<td></td>
<td><strong>AVERAGE</strong> 4.00</td>
<td></td>
</tr>
<tr>
<td>Incorporate, where possible, CE principles (e.g. team work, up-front consideration of life-cycle issues)</td>
<td>How effectively will the system increase the speed of the information flow during the on-site construction stage?</td>
<td>3.76</td>
</tr>
<tr>
<td></td>
<td>How well integrated are the components of the system?</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>How well does the system facilitate the communication among members of the construction team?</td>
<td>3.59</td>
</tr>
<tr>
<td></td>
<td>How effectively will it facilitate a common understanding of the information among the construction team?</td>
<td>3.71</td>
</tr>
<tr>
<td></td>
<td><strong>AVERAGE</strong> 3.77</td>
<td></td>
</tr>
<tr>
<td>Represent an improvement of the existing integration management of product and process information.</td>
<td>To what extent does it represent an improvement (or help) existing on-site information systems (how suitable is it)?</td>
<td>3.65</td>
</tr>
<tr>
<td></td>
<td>How easy is the system to use?</td>
<td>3.65</td>
</tr>
<tr>
<td></td>
<td>How generic do you consider the system to be?</td>
<td>3.88</td>
</tr>
<tr>
<td></td>
<td>What is your overall rating of the prototype system?</td>
<td>3.76</td>
</tr>
<tr>
<td></td>
<td><strong>AVERAGE</strong> 3.74</td>
<td></td>
</tr>
</tbody>
</table>
6.7 SELF-APPRaisal OF HIPPy

This section briefly relates the author's own work to early classifications of the proposed models, standards, and approach used for developing the prototype system (HIPPy). It presents a self-appraisal of HIPPy from a research perspective, so as to provide a comparison with other system in use.

The research conducted in this thesis proposed a hybrid methodology, exploiting ideas from the structured design analysis and object-oriented analysis techniques. The basic concepts with respect to the HIPPy methodology followed a similar approach to many of the integrated information projects and models discussed in Chapter 3. The basic stages include: strategic and domain analysis, integrated activity and object model realisation (or modelling development theory), and application production (implementation and demonstration realisation). These functions provided an iterative approach for simplifying the development of information systems, and have been proven (in one form or another) to be suitable for the development of information models and integrated systems (Ackoff, 1971; Karhu, 2001; Avison & Fitzgerald, 2001). The domain analysis and strategic modelling techniques were used to produce an activity breakdown of the domain (on-site construction), in order to identify domain perspectives (activities), which are aligned with the underlying information architecture of the construction industry, rather than reflecting historical groupings.

Two modelling approaches are used to build the information models that support the task of integrating the information from a number of domain perspectives into the integrated conceptual model. The approach followed is a combination of top-down and bottom-up modelling approaches. The top-down was performed starting with the domain analysis of the perspectives (activities) and the definition of the basic functions that the information models have to support. This was achieved by modelling the construction activities with IDEFØ and the on-site construction processes using scheduling charts or networking techniques (i.e. project planning the construction processes to be decomposed into operational units). These methods proved to be very effective in modelling the information at the on-site planning and construction stages, in addition to identifying the information process flows (e.g. time dependency input), which earlier classification models such as Sanvido et al. (1990), Karhu & Lahdenperä (1999), and Stevenson (1999) failed to
demonstrate. The combined use of these methods overcomes their individual limitations, as described in Chapter 3 and 4.

The notion of a perspective is central to the HIPPY methodology. A perspective is an object-oriented model describing the information needed to support a single well-defined activity (e.g. construct_foundation) through the use of Use Cases and sequence diagrams. An object-oriented analysis technique, utilising the concepts offered by UML, was used to analyse the various ways of exploiting an integrated information system by specifying the proposed Use Cases, and detailing them through a sequence description of the interactions (messages) between the object classes (or artefacts). The UML methodology was used to build the information models and to integrate them into a single, multi-faceted (flexible) integrated conceptual model of the domain.

The HIPPY methodology used for developing the information structures followed a hybrid approach and exploited both the IDEF0 and UML standards. The shortcomings of one are the indicators that reflect the strengths of the other. While IDEF0 models are inclined to have some restrictive quantities, such as the type of arrows permitted and time dependencies, UML models tend to have fewer limitations. For example, UML provides a visual modelling communication tool, which manages complexity, promotes reuse (e.g. reusable components), multiple models to capture system objects and logic from a user's perspective (although no process is currently defined), and/or be used alone to represent both product and process information, independent of implementation language; in particular encompassing information for software design. Furthermore, the hierarchy that is essential in IDEF0 models is not fundamentally required in UML diagrams.

The production of the activities (perspectives), and the integrated conceptual model is not intended to be in strict sequence. It is recognised that a top-down approach follows more naturally from the initial hierarchical breakdown stage, which usually comes first. A bottom-up approach such as UML, however, has the advantage that researchers modelling at the domain perspective level are not forced to compromise by adopting or considering a central model of the domain. This is evident in several existing approaches or project modelling systems, for example RATAS, MOPO, OSCON, COMMIT, CRPM, etc., which were inclined to follow this approach (see Chapter 3), using standards such as STEP, IFC-IAI, and CORBA, for the central model of the domain.
In practice, the modelling process using UML is iterative, whereby modelling the domain perspectives (activities) and classes are all continually evolving and contributing to each other. One further important point about the HIPPY modelling methodology is that it focuses on information required to support on-site construction processes (building information). The range of information modelled is therefore wider than just product data; in particular encompassing information about construction activities, operations, tasks, and building elements, components, classes, etc., and which is often unavailable when using standards such as STEP or IFC-IAI. The bottom-up approach therefore provided the necessary details to the general structure, detailing these elements and classes, defining their attributes and operations, and building the relationships with other classes.

Although there is no direct interaction between the information modelled in IDEFØ and the UML methodology, the IDEFØ modelled information can be used as a guide in the definition of the attributes and behaviour of the classes. The UML notation provides the advantage of supporting the process of migrating from one perspective to another, through the use of common elements (i.e. Use Case, Sequences, Classes, etc.). This has focused on the general structure of the information models, which provide major input for the prototype system (HIPPY). In the work described in this thesis, a distributed object based development database tool.

The development of the integrated information system for the implementation of the integrated conceptual model, has demonstrated the feasibility of the approach presented in this thesis, and also shown the usefulness of the application of both standards. However, from a researchers point of view, it has been found that UML needs extra support in the requirements capture stage (or domain analysis), mainly for the representation and definition of the structure of the information models. Therefore, while the HIPPY methodology provides a strong argument in terms of supporting information systems development, it may be limited in terms of providing an interpretation of the systems to the final end-user (construction managers). This interpretation might be better provided through exploiting tools and modelling standards, such as IDEF3, STEP, EXPRESS-G, IFC-IAI, and CORBA. Therefore, further investigation is necessary to check how such tools, e.g. IFC-IAI, CORBA, STEP, etc., can be used along with the approach presented to support the integrated information modelling system described in this thesis.
The main purpose of this study is to develop new approaches and information techniques that will contribute to improving the performance of on-site construction operations. This was to be achieved through the aim and objectives; that is, the development of an integrated information system (HIPPY), which was used for the implementation of the integrated conceptual model (i.e. product and process models) described in Chapter 4. The information models have been designed in such a way, so as to provide a template, allowing construction managers to enter their own constraints, parameters, etc. As a result, the modelling approach proposed in this thesis provides an flexible and easy to use methodology for integrating the heterogeneous information sources, which construction managers typically receives during the on-site construction planning and operation stages of SMsBP.

6.8 SUMMARY

This Chapter has described the use and structure of HIPPY. The actual building products and processes for a house project are used, as an example to demonstrate the system's operation and functionality. It has also provided the results of an evaluation carried out by selected researchers and industry practitioners, on its effectiveness in managing the integration processing of product and process information. The fact that the system was not validated on a real test case of a construction project, suggested that the actual building requirements illustrated in the information models, may not necessarily reflect the actual situation had HIPPY been used on a live project. However, within the constraints of the research (e.g. time and funding), the integration of product and process information retrospectively provided an adequate demonstration of how HIPPY could be used. The evaluation confirmed that, while improvements are required to make HIPPY fully operational, it does provide a good initiative for integrating product and process information to support CE during the on-site construction stage of SMsBP.

The next Chapter presents the summary and conclusions of the whole research project. It discusses the contributions made by this research; identifies limitations, and presents recommendations for practical applications and future research.
CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

7.1 INTRODUCTION

This chapter concludes the research project, which focused on the integration of product and process information during the on-site construction stage. The investigation resulted in the development of a product, process, and integrated conceptual model, and the associated prototype system (HIPPY). This chapter summarises the findings of the investigation, the development, implementation and evaluation of the resulting prototype system. It concludes that an integrated product and process information modelling system is an effective means for integrating the heterogeneous information sources, which construction managers have to contend with during the on-site construction stage. The chapter ends by making recommendations for practical applications and future work.

7.2 GENERAL SUMMARY

This section reviews the original aim and objectives of the project and compares it with the research undertaken throughout the project. This is followed by a general summary of the specific tasks undertaken in the development of the product and process models, integrated conceptual model, and the associated prototype software. The original aim and objectives of the research project are reproduced here from Chapter 1.

The aim of this research project was to develop a practical and easily adaptable integrated product and process information modelling system (HIPPY), based on, and compatible with existing information systems. On a more detailed level the aim was to be achieved through the following specific objectives:

- to investigate the role of system integration by reviewing product and process information;
- to review and evaluate the use of product and process models in the construction industry;
• to investigate current modelling practices and standards in the construction industry, in order to draw comparisons with similar practices from other industries, both in terms of product and process representation and model content;
• to review and evaluate application development tools and information system requirements to support a suitable integrated information structure;
• to develop a conceptual integrated product and process model for design and construction, based on concurrent engineering principles; and
• to implement a prototype information system capable of managing product and process information essential to construction management, and to evaluate the developed prototype system.

This research project has investigated the integration of product and process models within the context of concurrent engineering in construction. The general background and research problem was introduced and justified in Chapter 1. Government interest in improving productivity and performance of house-building practice (Egan, 1998) and the current lack of IT and integration between product and processes on construction sites (Construct IT, 1996), and possible solutions to these problems was discussed. It has argued that consistent use of current data structures (OLE), object-oriented modelling, and concurrent engineering, offers major scope for overcoming many of the enduring problems of the construction industry (Evbuomwan & Anumba, 1996), particularly those that are legacies of the age-old fragmentation problem. The investigation into the concepts of CE as practiced in the manufacturing industry, also established that CE could be adopted in construction. This is because of a number of factors, which include the following:

• The similarities between manufacturing and construction with respect to the repeated processes that are involved in the design and production of products; imply that management change developments (e.g. concurrent engineering), which has led to improvements in productivity (as a result of process re-engineering), can be used in construction.
• The goals and strategies of concurrent engineering directly address the problems in the construction industry.
• There are a number of recent related projects that have successfully developed models based on concurrent engineering principles in the construction industry, for example BEACON (Khalfan, 2001) and CRPM – ClientPro (Kamara, 1999).
• There are already practices in construction that have elements of concurrent engineering, as well as examples of limited implementation of CE concepts in the industry.

The research methodology was then described in Chapter 2, which included the use of both quantitative and qualitative approaches and procedures (also called triangulation method). A variety of methodologies and strategies were adopted to achieve the defined objectives of the research. These included: extensive literature reviews, discussions and semi-structured interviews with domain experts, industry surveys, participation at seminars, workshops, and conferences to interact with other researchers and professionals in similar research areas, and peer review of published work. Appendix 2 lists the reports and published work generated from this research.

A state-of-the-art investigation into the role of system integration, information modelling and object-oriented techniques, integrated developments, and associated research projects that address integration in construction were reviewed in Chapter 3. The literature review identified areas that had not been previously researched and surveys were formulated to investigate and evaluate these areas. The review revealed that present techniques do not adequately address product and process integration during the on-site construction stage, but are more concerned with productivity and efficiency issues at the planning and design stage.

The findings of the surveys, together with related research on the subject, also provided the basis for assessing the suitability of the current process for integrating product and process models in a concurrent engineering context. The investigation into the management of information during the construction stage, also led to the identification of various tools and techniques, which could facilitate the development of an integrated product and process model. Of the modelling methodologies reviewed, UML, IDEFØ and Scheduling were considered to be the most appropriate for representing the functional requirements and information perspectives for processing and managing product and process integration during the on-site construction stage. The utilisation of a top-down and bottom-up modelling approach was described in Chapter 4, in order to define the structure, and to determine the building levels necessary for the development of product and process models.
The development of the product and process models were based on an iterative process, to ensure their relevance to, and practical application in, the construction industry. This involved a number of tasks, such as the review of modelling techniques, tools, and related research journals, identification of requirements (scope and purpose), identification of on-site construction activities and building elements, feedback from academic staff involved with the research, and discussions with practitioners in the construction industry. The process for developing the integrated conceptual model also involved an iterative process, based on a number of assumptions and system requirements, which supported the hierarchy format of the product and process models, was also presented in Chapter 4.

The software comparison indicated that the available tools and techniques could support the modelling notations needed to represent the product and process requirements. However, only two provided the necessary facilities to support all the modelling notations. Therefore, Visio 2000 was considered to be the most appropriate software tool for graphically representing the functional requirements and information models perspectives. The limitation and benefits of the models and software were also discussed.

The construction entities (activities and components) incorporated in the building levels and stages of the information models are designed to ensure the proper definition of the system requirements, the incorporation of all the perspectives and information sources, and the relational structuring of product and process diagrams, and the adoption of a concurrent engineering approach to design.

Suggestions for the implementation of the integrated conceptual model with respect to the building levels in the construction process, as well as the objectives for developing a prototype system for the information models, were also discussed in Chapter 5. To demonstrate how the integrated product and process model could be implemented in practice, a paper-based example, using the requirements for a typical house-building was carried out. This led to the actual development of the prototype software (HEPPY). The functional and informational requirements and perspectives of the information models provided the basis (system analysis) for the development of HIPPY.

The prototype developed (HIPPY) is geared towards construction managers of small to medium sized building projects, who already have PCs as resource in their organisation.
The development and implementation (Chapter 5) were based on an iterative process that involved reviews and feedback from researchers, seminars, and discussions with practitioners in the construction industry. Its development was also influenced by the principles of software prototyping (Davis, 1995), and the need to realise the full benefits of concurrent engineering through the use of computer technology. The approach was based on a number of assumptions, using actual product and process information for a house project and modelling techniques within the construction sector, as well as adopting management change techniques from other disciplines (e.g. manufacturing).

Details of the resulting three-tier client-server operational system architecture, system components, implementation environment, and system development using Microsoft Access, a relational database management system, and MS Office were also described in Chapter 5. This provided a relatively quick, inexpensive and effective way of demonstrating how the product and process information can be processed in a computer environment, and its integration with other computer-based design and construction activities. The implementation stage, demonstrated through a potential practical application of the system, revealed that given the right framework, the integration of product and process information could be achieved.

Following its implementation, HIPPY was used to demonstrate the potential practical operation of integrating the information requirements for the house-building, used in the paper-based example, and a selected of researchers and industry practitioners, who represented some of the potential end-users of the system were asked to evaluate its effectiveness, were discussed in Chapter 6. Some of their comments and suggestions were incorporated immediately, while others may be implemented at a later date, in the future. The evaluation confirmed that, while improvements are required to make HIPPY fully operational, it does however, provide a good initiative for integrating product and process information to support CE during the on-site construction stage of small to medium sized building projects.

7.3 BENEFITS

The HIPPY prototype offers many direct benefits to construction managers in the proactive management of product and process information. These include:
• an interactive software tool that can be used to access a variety of heterogeneous information sources during on-site construction;

• the information models provide a dynamic graphical representation for understanding the on-site construction process;

• it helps construction managers to clarify their vision of the facility to be constructed, in that it ensures that product and process information are clearly defined at the early stage of the construction process;

• it facilitates communication, collaboration and a common understanding of the product and process requirements and information amongst construction team members. As a result, it helps bridge the technical terms and jargon gap that can present barriers to knowledge and understanding; thus enhancing better communication and collaborative working because of the common understanding of the construction information amongst members of the construction team;

• it effectively represents, describes, integrates and automates data exchange between various software tools and applications, document files, and Web pages;

• there is major scope or potential to use HIPPY as a training tool for teaching inexperienced site managers/engineers aspects of the on-site construction process;

• it effectively provides facilities for calculating the quantities of building elements, and generates activity durations; thus providing the basis for effective information management throughout the on-site construction life cycle phases;

• it represents a marked improvement to the existing process (computer aided engineering approaches) of integrating product and process information during the on-site construction stage;

• a common user-interface across program applications, information sharing among software applications to reduce data entry, and interaction between applications so that, one application immediately displays the effect of a change in another through the use of compound documentation such as object-linking and embedding;

• there is a facility for integrating, viewing, editing, and printing product and process information reports, and therefore helping to clarify their vision of the facility to be constructed;

• more generally, the user would enjoy the benefits of a more uniform, easy to use, comprehensive, flexible, and powerful approach to processing product and process information during the on-site construction stage;
the construction industry also stands to benefit from the associated benefits of adopting concurrent engineering principles. For example, the enhancement of teamwork with better co-ordination of the efforts of team members, reduced construction times, costs, claims & disputes, and quality improvements to the overall facility will result in more satisfied clients.

7.4 LIMITATIONS

The limitations of the integrated HIPPY information model include:

1. The implementation of the information models requires sufficient time that is allowed for generating product and process information. Since this is at odds with the current culture in the construction industry, where insufficient time is spent in defining actual real on-site information, this approach might not be welcome by some sectors of the industry. Furthermore, if a design team implements the models, they may be constrained by design deadlines, and may consider the approach represented by the models to specific or unnecessary.

2. The prototype has a limitation with resects to its applicability to real case studies due to the evaluation being carried out within a research environment with industrial practitioners. This does not invalidate the approach being generated within a research environment. It does however; raise questions to whether the scope of the information represents a real life project. More representative results could have been obtained by involving a substantially larger number of industrial practitioners.

3. The integrated model has been specifically developed to focus on the on-site construction stage and only shows the upper levels of product and process information. In order to implement the complete model, it would need to be developed to a more detailed level. However, this could potentially make the model vary large and confusing.

4. The model lacks a structure for representing the importance and complexity of performing the different processes, for example virtual 3D graphic representation.
Chapter 7

7.5 CONCLUSIONS

The conclusions from the research are summarised under the following headings:

- Prototype System (HIPPY);
- Integrated Product and Process Modelling; and
- Integration in Construction.

7.5.1 Prototype System (HIPPY)

The conclusions relating to the prototype system include:

1. The prototype system is an innovative tool that allows the user to view and edit a variety of product and process information from a number of different applications; generating a decision support tool for site managers that would aid them in making decisions about on-site construction processes.

2. The integration of product and process information within a computerised medium would assist the construction manager and give an insight into the vital areas that need to be addressed on site.

3. Hyperlinks and OLE data transfer methods provide a unique approach for ensuring that adequate focus on the on-site construction process is maintained, by means of connecting links to objects (product and process documents, text fields, Web files, etc) to various other applications, which can be automatically started and updated.

4. The generation of integrated product and process reports provides the site manager with a decision support tool that would aid him/her in making decisions about on-site construction processes. In addition, provides the site manager with a unique approach for viewing, integrated product and process information concurrently.

5. Information systems formulated on integrated product and process model approaches can be more powerful than paper-based approaches because they support professionals in co-ordinating work, and related information projects by making face-to-face discussions more effective.

6. Therefore, the construction industry needs to take advantage of the approach proposed in this thesis, as it represents an important step towards integration of the heterogeneous information sources within the industry.
7.5.2 Integrated Product and Process Modelling

The main conclusions with respect to integrated product and process modelling include:

1. Although, product and process models have been developed for several aspects of the life cycle of a constructed facility, there is much more benefit in integrating them. However, this research has shown that existing modelling methods for product and process models are often incompatible, making integration difficult. In particular:

   - many modelling efforts are not developed to the level of detail considered useful for developing practical applications specifically for the domain of on-site construction;
   - many modelling approaches are either not applicable to modern or on-site construction works (e.g. on-site construction process) or too specific to an organisation’s needs;
   - most of the information models are rather theoretical and too generic in nature, and have not yet reached a stage where software companies can implement them;
   - there is a growing need to produce less generic and more unambiguous proposals, possibly for a specific domain, such as construction management, because key finding from the research has shown that there integrated model support for the detailed design stage of a project’s life cycle, some supports for the early design, facility management, and client requirement/briefing stages, and limited support for the actual on-site construction stage, and to test them in real-life construction situations;
   - there is a need for new innovative approaches with wide applications that will facilitate effective integration; developing and applying new technologies, and optimising the benefits of these new computerised information and communication technologies in the construction industry; and
   - modelling the construction process has a tendency to focus on a particular viewpoint. A more flexible integrated information system can be developed by combining various activity and object-oriented modelling techniques that integrate multiple applications and viewpoints.

2. The integrated product and process model represents an innovative approach, and significant development in the understanding and representation of the on-site construction process in that it:
employs an analysis and design techniques (IDEFØ) to ensure consistency throughout the model;

employs object-oriented analysis and design techniques (e.g. UML) that examine requirements from the perspective of the classes and objects; offering rich semantics, and facilitating a description of the on-site construction process (illustrated in the process model) in terms of real world objects, and encourages the reuse of objects;

serves as an intermediate stage for the integration of product and process information, system analysis and mapping of product and process data, and to complete the basic structure of the information models; and

ensures the adequate focus on construction entities for providing an organisation and categorisation of information describing process activities, product elements, and components in terms of their attributes, operations and functions.

3. Product and process models have been presented as vital in this regard, and will enable members of a construction team to collaborate on the basis of an integrated conceptual ‘project’ model during the construction phase of small to medium sized building projects.

4. The integrated product and process model goes some way towards achieving a more powerful and semantically rich information environment.

7.5.3 Integration in Construction

A number of conclusions can be drawn with regard to integration in construction:

1. The integration of product and process information, amongst other measures, is one way in which the construction industry can improve its efficiency and respond better to the change factors affecting it.

2. Improvements in productivity and quality depend largely upon management change strategies, emerging IT technologies, and actions carried out by various individuals, struggling to mobilise resources, which are frequently inadequate and inappropriate and facing a formidable array of constraints and difficulties. It is to these techniques that the construction industry needs to target, and where necessary, provide an ‘integrated information framework’ that is capable of implementing these approaches, tools, and IT techniques to specific process issues in construction. In time the benefits
for improving the effectiveness of the project delivery process will enhance the productivity and quality of the construction industry.

3. The introduction of concurrent engineering in construction has the potential to contribute towards the effective 'integration' of the project delivery process, by improving quality, adding value, reducing cost, reducing construction scheduling, and overcome some of the industries current problems.

4. Clearly, there is great need for 'effective integration' (project team members, heterogeneous software tools, project stages, etc.) within the construction industry. The integration of product and process models within a concurrent engineering context will contribute significantly to this.

5. The on-site construction stage is important and an integral part of the project life cycle and should therefore, not be studied in isolation.

### 7.6 RECOMMENDATIONS FOR FURTHER WORK

The research project has revealed a number of areas for further work and development. These are discussed with respects to:

- Prototype System (HIPPY);
- Integrated Product and Process Model; and
- Integration in Construction.

#### 7.6.1 Prototype System (HIPPY)

The following recommendations can be made with respect to the prototype system:

1. Further work on improving the prototype software includes the evaluation of the system on a real test case in order to modify it according to changes within the construction industry.

2. Refinement of the prototype software is essential, if it has to be used on professional basis. This could be achieved by implementing it using the Microsoft Visual Basic programming language, or other languages (C++, Java) making it more user-friendly and including new tools and facilities. On the other hand, modifications could also be
made to make the information models and prototype software flexible enough for use within other industry sectors or other countries.

3. Enhancement of the knowledge base for the design environment to include in-depth process knowledge (project scheduling and resource scheduling), extensions to support design life-cycle information integration, capturing process requirement knowledge (client requirements, procurement strategies, site conditions and spaces, etc.) and other environmental factors need to be taken into consideration to present both the problem and the context for design (i.e. experience at all phases, machine learning and improvements to the knowledge base.

4. Another key development would be to make the system Web-Based; that is, structuring the user interface into a computer-network based system, so that it is accessible to members of a distributed virtual project team, allowing users to access the technology and the necessary knowledge to use it. Since its development, the evaluation of the prototype system has revealed that this is the main area for further research work.

5. The evaluation through the use of a questionnaire identified that the prototype system would benefit greatly if it was linked or integrated with other packages, such as a 3D/4D virtual reality modelling system. Therefore, further research into the development of a prototype system, for the use of computer visualisation will contain 3D information about the actual assembly of the building components, whereas HIPPY will contain the information about the locations of the building components and the required materials and their specifications to build it. Hypertext links that would lead to the actual activity or operation being performed would enhance the system. Thus communicating design information is essential for improving communication and collaboration in the construction industry.

6. The involvement of software vendors in research projects will increase the commercial take up of research prototypes.

7.6.2 Integrated Product and Process Modelling

Since the development of the information models, the evaluation of HIPPY has revealed that there would be benefits in restructuring the models to reflect the following:

1. Integrated models that cover the whole project life cycle from conception to demolition need to be developed.
2. The use of established and emerging interoperability standards (such as STEP, IFCs and UML) should be encouraged. This will have a positive impact on model reusability, longevity, and interoperability with legacy IT systems in the construction industry.

3. There is a need to ensure the reusability of integrated models to avoid duplication and waste of resources. In this regard, researchers should be required to make their models readily available on the Internet or from a central library.

4. A work breakdown structure that is recognised throughout the industry, such as the systems promoted by the IFCs and STEP, SMM7, CI/SfB Project Manual: Organising Building Project Information or the Unified Classification for the construction industry, would make the location of reference of sections (categories) of the models easier to identify by the members of the construction team.

7.6.3 Integration in Construction

Aspects of integration in construction that can be further investigated include:

1. Many enabling information and communication technologies (such as virtual and mixed reality, the internet, intranets, information exchange standards, etc.) are now available for integrating information; however, construction management capability presents a significant potential constraint on their ability to adopt or understand these emerging technologies. Therefore, training in these particular technologies need to be undertaken.

2. Further research should be conducted into integrating heterogeneous CAE or CAD tools that are being used by different disciplinary participants on construction projects. CAD systems are the source of most graphical data, whereas, CAE tools support paper based information. Many of the commercial CAD applications used within the construction industry are primarily geometry modellers and exploit several file formats (e.g. 3DS, DXF, etc.). These files formats do not solve the problems associated with file transfer, therefore further research should be carried out to search for the best ways in file format standardisation for both 2D and 3D information.
Chapter 7

7.7 CLOSING REMARKS

The effective integration of product and process information is a key component of the implementation of concurrent engineering in construction, and is a vital step towards the management of project information. However, neither the current process of integrating product and process information, nor current research efforts adequately address on-site construction in a concurrent engineering context. This research project has presented an integrated information platform (methodology) that integrates product and process information to effectively support collaboration and project team communications during the on-site construction stage of small to medium sized building projects. By integrating product and process models, a more powerful and semantically rich information environment has been created.

The benefits of this integrated environment lie in added data consistency, and improved life cycle support for design and construction information. The construction industry needs to take advantage of the approach proposed in this thesis, as it represents an important step towards integration of the heterogeneous information sources within the construction industry.
REFERENCES


References


References


References


Muller, R. J. (1999). Database design for Smarties using UML for Data Modelling, Morgan Kaufmanns Publishers, San Francisco, California, USA.


Appendix 1

DEVELOPMENT OF DATABASE OBJECTS FOR HIPPY

A1. Designing Tables

In a relational database, a table is the basic structure that models each entity set (a row of data values), corresponding to the identified entities in the conceptual and information models described in chapter 4, and presented in Table A1. The second level entities represent the lowest decomposition of the entities in the information models, and form the starting point for designing the tables.

Table A.1: List of Entities in the Integrated Product and Process Model

<table>
<thead>
<tr>
<th>First Level Entities</th>
<th>Second Level Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building Product Data</td>
<td>Products</td>
</tr>
<tr>
<td></td>
<td>Sub-Products</td>
</tr>
<tr>
<td></td>
<td>Building Elements</td>
</tr>
<tr>
<td></td>
<td>Building Element Types</td>
</tr>
<tr>
<td></td>
<td>Building Components</td>
</tr>
<tr>
<td></td>
<td>Building Resources</td>
</tr>
<tr>
<td></td>
<td>Building Component Types</td>
</tr>
<tr>
<td>Building Process Information</td>
<td>Processes</td>
</tr>
<tr>
<td></td>
<td>Process Activities</td>
</tr>
<tr>
<td></td>
<td>Function Boxes</td>
</tr>
<tr>
<td></td>
<td>Activity ICOMs</td>
</tr>
<tr>
<td></td>
<td>Activity ICOM Types</td>
</tr>
<tr>
<td></td>
<td>Activity ICOM Arrows</td>
</tr>
<tr>
<td></td>
<td>Tool Equipment</td>
</tr>
<tr>
<td></td>
<td>Mechanical Plant</td>
</tr>
<tr>
<td></td>
<td>Construction Team</td>
</tr>
<tr>
<td>Mixed Sources of Information</td>
<td>Product and Process Details</td>
</tr>
<tr>
<td></td>
<td>International Standards</td>
</tr>
<tr>
<td></td>
<td>Building Specifications</td>
</tr>
<tr>
<td></td>
<td>Activity Durations</td>
</tr>
</tbody>
</table>
Access allows the design of tables in a number of ways, including the use of the *Table Wizards*, an automated help facility that provides steps for the process of table creation by suggesting predefined layouts for tables. Using the Table Design View facilitates the design of tables in Access. Figure A.1 shows the design view of the ‘function boxes’ table, which is used as an example of how the tables in HIPPY were manually developed.

![Figure A.1: Design view of function boxes table](image)

The Table Design View is divided into three columns. The first column contains the field names, which correspond to the ‘attributes’ of the function box entities, and can hold up to 64 characters in length and include letters, numbers, spaces, or punctuation marks. In order to identify the records under function boxes a primary key, ‘FunctionBoxID’ is defined to uniquely identify each record in that field. Also, the ‘ProcessActivityID’ is a foreign key, which is used to establish a link or relationship between the process activity and the function box tables. After entering the desired field name, press Tab, and the insertion pointer moves to the Data Type column. This column specifies the data type for the data that will be stored in the field. By default, Access assigns a data type of text, which can be
changed by clicking the arrow next to the data type filed to cause the drop down list box to appear (Figure A.1). The final column provides a description of each field record, which can be viewed in the Form view window, or through assigning custom property text tags. The bottom part of the design window is used to define the field properties such as size, caption, validation, etc., and depends on the select data type.

After the tables were created in the design view window, the adding, editing and deleting of data records in the tables were facilitated in the Datasheet View, as shown in Figure A.2

---

**Figure A.2: The datasheet view of the function boxes table**

The list of tables in HIPPY is presented in Figure A.3. The entities for building standards and images/pictures are obtained from external sources, through the use of Hyperlink and OLE fields. Hyperlinks are used to store a combination of text and numbers that serve as a hyperlink address, or path connection to an object in the same or another Access database, to a document created in another Office program, or to a document on the Internet or local intranet. Hyperlink fields can be addressed to any Internet Web site; linked to any OLE or ActivityX application, or file names on hard disk or a network drive. The hyperlink field contains the address of the targeted object; when activated, the hyperlink automatically jumps to that object, and if the object is the product of another application, that application is automatically started.
In contrast, OLE objects can include any form of information, such as product and process models, pictures, graphs, or word documents that are created by an application outside Access and linked or embedded in another application; in this case, inserted into a database table using the OLE object field. The capabilities of OLE will be discussed later in this chapter, under the data transfer section. These tables were developed to facilitate their use in determining specific information and graphical data, and can be reused downstream (i.e. the information is constant, and can be reused in various building projects).

**A2. Creating Relationships Between Tables**

Before the relationships are established, it is worth mentioning that each table represents a type of object in the real world. For example, each record in Figure A.2 represents a real function box, and each entry in the construction team table represents a real person (builder) carrying out a particular function. Therefore, the first step in deciding what relationships is needed between the different tables, is to ask the question, ‘what types of relationships can exist between the real world objects?’

---

1 The term ‘relationship’ as used here has nothing to do with the use of the work ‘relational database’. It is used in this work to simply imply an association and/or interdependency, as described in preceding chapters.
As Access is a relational database, which provides the facilities for creating a finite collection of real world objects, in the form of tables, which provide the means to implement the integrated product and process model (described in Chapter 4). In order to associate or compare data between these tables, each table will often need to establish a relationship or link between them. In Access, there are two ways for establishing relationships: through queries, or by selecting the tool/relationship menus item on the database window (shown in Figure A.3). In Access there are four possible kinds of relationships between any given pair of tables, these include: one-to-many, many-to-many, one-to-one, and none.

The type of relationship defined, generally depends on how many records in each table are likely to have the same value. However, in order to relate these tables, they must each have a primary key field, a field that contains a unique value in every record. A primary key can also be a combination of two or more fields whose combined value is unique for all records. Figure A.4 displays the relationship window where the required tables are added to establish the relationships, and is used here to illustrate some of the relationships between tables in HIPPY. Each table is related to at least one other table, thereby creating a relational database.

![Figure A.4: Types of relationships between tables in HIPPY](image-url)
In the development of the HIPPY database, the most commonly used relationship (indicated by the line) is one-to-many (also referred to as a polygamous relationship), and signifies that a record in one table can be related to one or more records in another table. The table on 'one' side is often called the parent table and usually contains the primary key, and the other is called the child table. For example, the products table, shown in Figure A.4 has one record for each product, and is represented by a 1 (one) appearing at the end of the line, and attached to the table on the 'one' side.

In contrast, the sub-products table has more than one sub-product for the same product, and is represented by the ∞ (infinity) symbol appearing at the table of the 'many' side of the relationship. In general, the table containing the foreign key will be the many parts of the relationship. The implementation of many-to-many relationships involves a little more organising, and is not really permitted as such in a relational database. Many records in one table have the same values in the key field as many records in the second table. To implement this, a third table is created called a junction table, and is placed between both tables converting the many-to-many to two one-to-many relationships. This is illustrated in Figure A.4, were two one-to-many relationships have created between the Products and processes tables, and the Activity ICOMs and Function boxes tables. The one-to-one relationship is a form of lookup in which each record in one tables has a matching record in the other table. Neither table is designated as the parent. Thus, the key fields in both sides are the primary keys. Tables with no relationships, simply means that some objects do not have any relationship with each other. Neither of these two relationships were used for creating relationships between tables; they were only included for completeness.

In order to guarantee the validity between tables, enforcing a relationship's referential integrity was the best way to avoid any serious errors in the database. Figure A.5 presents the referential integrity window, an optional system, which provides a set of rules that guarantees that the relationships are valid and the database will remain intact, by means of preventing the entering, editing, or deleting of any records that violates the relationship between tables.
A.3 Design of Forms for HIPPY

The tables provided the foundation for creating the forms, which will be used for displaying, entering, and editing information into the tables. The information is entered in a form, and Access stores the data in the corresponding (underlying) table in the database. The product and process browser modules described in Section 5.4.3, and illustrated in Figure 5.1, correspond to the information captured in the information models, and basically involved the documentation of various types of information. The information that has been decomposed in a hierarchical manner, to facilitate the data breakdown and stored in tables, can be facilitated further by simply designing forms that are based on the tables storing the data (e.g. the list of tables illustrated in Figure A.3).

In Access, a form provides an easy way to enter and display data stored in a table, it also provides the means to link to any application (i.e. includes user-interactive elements for acquiring additional information and executive user choices). The design of a form can either be created manually by means of a Form Design window, which allows objects to be dragged on the form such as fields or text, or quickly by clicking the NEW OBJECT button on the toolbar, or by means of the Form Wizards, shown in Figure A.6.
The design of a new form starts by clicking the ‘New’ button on the database window; this opens the ‘New Form’ dialogue box, which displays a number of options for creating forms. These include:

- the Design View option creates a new blank form without using a wizard;
- the Form Wizard option automatically creates a new form based on the fields selected from one or many tables, as long as the selected tables have relationships;
- the AutoForm options automatically create the selected option;
- the Chart Wizard facilitates the development of a form that includes charts; and,
- the PivotTable Wizard creates a form that allows the embedding of Excel pivot tables (e.g. Excel spreadsheets). A PivotTable form is an interactive table that performs calculations, such as sums (the default for numeric fields) and counts (the default for text fields), based on how the data is arranged in the PivotTable form.

Figure A.6: Window for creating new forms and illustrating the underlying table.
Appendix I

As with tables, Access also offers different views for entering data in forms: Design View, Form View, and Datasheet View. The Design View is used to made changes to the design of the form or to change the properties of the controls and other data that are connected to the form objects. The Form View is the finished product (form) that is used to view, enter, and edit data, typically in a record-at-a-time format. The Datasheet View displayed the data in the familiar row-and-column format, identical to that of a table’s datasheet or a query’s dynaset. The design of forms in HIPPY utilised useful features from a variety of options and views, in order to establish an efficient controlled information flow. Figure A.7 shows the design view of the ‘building element’ information form, which illustrates how the forms in HIPPY were designed.

The form design is made up of elements called controls. All the objects placed on a form, and the information displayed is represented by these controls. A control is a graphical object that can be placed on a form to display data, perform an action, or enhance the appearance of the form. Examples of controls (e.g. text boxes, combo boxes, images, pictures, etc.) are boxes that display the contents of field values, field labels, calculations, lines and rectangles, and command buttons. Controls can also run Access micros or call programs written in VBA code, and also contain embedded OLE objects, such as an Excel spreadsheet, Web picture or video clips.

In Access, controls come in three basic types, depending on their relationship to the values in the tables: a bound control, an unbound control, and a calculated control.

- The bound control gets its value from a field in the table and as the data changes, the value of the bound control changes with it. The data fields that are added to a form are examples of bound controls, and can include bound OLE objects that display graphics or contain sound or other Windows data stored in an OLE Object field.
- The unbound control has no type to the underlying table data and retains the value that is originally entered; thus, the data is not stored in any particular field. Most data that is stored in an unbound field is usually activated by use of micros or by VB code. Examples of unbound controls are combo boxes, lines, rectangles, labels, images, and OLE Objects (word, Visio, Excel, etc.) that need to be changed or updated.
- Calculated controls are a special type of unbound control whose contents are based on calculations, which get their value from values in tables and are actually ‘expressions’ containing functions and operations, in addition to fields, that produce a result (i.e. the
expression used to perform the calculation generally is based on one or more fields of the underlying table or query).

The value illustrated in the calculated control, shown in Figure A.7 changes as the values in the underlying fields change. However, calculated controls cannot directly be edited.

![Diagram](image-url)

**Figure A.7: Design view of building element form showing properties sheet and fields**

Forms, and in many ways reports, as well as their controls share similar properties, such as record source, caption, width, and filter, but also have a few unique properties (Anderson, 1999). Each type of control has an appropriate set of properties such as name, caption, source, format, decimal places, colour, filter, position, and size. All the properties relevant to the current selected control are displayed in a window called a *property sheet*. Properties are used to establish the characteristics of forms and report design elements, and everything in a form or report design has properties. These properties set the structure, appearance, and behaviour of the controls, and can also determine the characteristics of the text and data contained in a control. Figure A.7 also displays part of the property sheet for customising or modifying the appearance and behaviour of the ‘building elements’ form.
A partial list of the 30 forms and subforms of HIPPY are also presented in Figure A.6. Some of these forms are based on tables with the same name, and in some cases, more than one form has been developed from a table. Also, a number of forms contained one or more subforms and were created using fields from several tables or queries; that is, a 'hierarchical' form that retrieve data from more than one table at a time.

A hierarchical form (sometimes referred to as a multiple form) usually consists of a main form and one or more subforms, and is created by embedded forms within forms. The main form, or master form shows data from records on the 'one' side of a one-to-many relationship, whereby the embedded form or subforms (sometimes referred to as the child or detailed form) shows data from records on the 'many' side. Figure A.8 illustrates an example of a hierarchy form for types of 'process activity ICOMs', displayed in the Form View window. In the example, the form shows information relating to activity ICOMs, which are fields taken from the 'process activity ICOM types' table. The subforms display the information relating to the ICOMs types, and are tables in their own right. Each field within the top subform when selected will display the information relating to the ICOM in the bottom subform. Thus, creating a hierarchical process.

![Figure A.8: Design view of a form that displays the hierarchy of component types](image)
A.4 Design of other Database Objects

Other database objects used in HIPPY were: queries, reports, macros, and modules. However, the manual development of macros using the event builder was initially limited to the early stages of the form development process, and as a working knowledge of VBA became more and more understandable, it also became clear that VBA code had fewer limitations in comparison with using micros. In addition, the use of modules was also limited, since macros, VBA code, and queries provided the functionality required at this stage in the development. The development of these objects followed procedures that were similar to those described for forms and/or tables.

The most common operation performed on the data in a table is to carry out a query. A query is simply a question, which can be asked of the data in a table. Using a graphical tool known as the Query By Example (QBE) Grid, illustrated in Figure A.9, develops a question, and the answer is displayed in a dynaset, which contains the record (10 m³ strip foundation) that satisfies the criteria specified in the query.

![Figure A.9: Query design view showing QBE grid for 10 M³ or less strip foundations](image)

The development of queries facilitated the design of the forms and was used to prioritise and find additional information. In general, the queries helped to:
- View data from multiple tables sorted in a specific way.
- Perform calculations on selected groups of records (scheduling dates and quantities).
- Find or display duplicated or unmatched records.
- Update data, delete records, or append new records in a table.
- Create a new table for records with one or more table.

Figure A.10 lists some of the queries used in HIPPY. It also displays the New Query dialog box (for generating new queries), which can be opened by clicking new object and selecting new query, or clicking the new command button. A description of some query records in database view is also displayed.

![Figure A.10: A list of queries used in the development of HIPPY](image)

In order to facilitate the automation in HIPPY, macros were designed. A macro is a set of one or more commands that perform a particular action, for example, navigating between form records, or opening and closing a form. A command button is created and linked to a macro and then used to run that macro, for example, a command button 'Save' would run a macro, which saves the added database file in Access. Alternatively, a command button 'Print' would allow the user to print the record displayed in the form or report. Macros were also used for:
• Trapping certain keystrokes throughout the application in order to make it user-friendly. For example, assigning frequently used actions to keystrokes, such as printing current records when users hit Ctrl+P.

• Carrying out a series of actions whenever the database is opened (this is done via the Autoexec macro). When Access opens an existing database, the first thing that Access does is to set any options that have been specified in the Tools/Startup dialog. After this, it checks to see if a macro called Autoexec is present. If it is, then Access executes it immediately. This handy feature allowed the writing of a record to a log file to indicate that the application has started up (e.g. ensured that the welcome screen was displayed first).

Although, macros have their uses for simple tasks, they do however possess limitations when developing buttons that are intelligent enough to allow users to click them only if they represented a valid choice (i.e. disable or enable macros according to where they were in the records behind the form). While this may not be a problem for some users, one of the requirements of HIPPY is to provide a certain degree of intelligence, facilitating a user-friendly interface that will appeal to the end-user (i.e. encourage people to use the system).

To achieve this, the Command Button Wizard (shown in Figure A.11) was used to generate VBA code. This enabled a more varied selection of actions to be executed than were available through macros. The use of VBA code also executes faster than the equivalent macro action, and since ‘speed’ is generally a critical factor in impressing the end-user, it was therefore practical (based on reason) to convert macros (except the macros used for display) into VBA code. The advantages and disadvantages of using VBA code and macros can be found in Smith and Sussman (1999).

![Command Button Wizard](image)

*Figure A.11: Generating event procedures through the Command Button Wizard*
Forms were developed for imputing data and viewing it on the screen. However, to enable users to view hard copies of this data, Access provides several methods for converting data to information by creating reports, performing calculations on those records, and/or changing the sequence in which the records are displayed. A report is a printed document that displays information (i.e. the finished product) from the data tables stored in a database. The reports listed in Figure A.12, were created and implemented in HIPPY with the New Report Wizard (also shown in Figure A.12), and were based on the underlying product and process tables or queries in the HIPPY database (Figure A.3 & Figure A.10). Access provides several different options for creating a new report, although the report wizard is the easiest way to design and customise a report in a professional way.

Report Wizard and list of reports implemented in HIPPY

Reports were developed in much the same way as forms and composed of similar design elements with event procedures attached to them. In order to facilitate the automation, a command button was created on the forms and linked to the event procedures, when initiated, the users can either preview or print the selected information in a more professional manner. Reports also allow users to select and preview multiple product and process information, including totals and summaries, as illustrated in Figure A13 & A14).
### Appendix I

#### Integrated Product and Process Information

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Start Date</th>
<th>Finish Date</th>
<th>Element Name</th>
<th>Length (m)</th>
<th>Height (m)</th>
<th>Width (m)</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contract Foundation</td>
<td>08/1/2001</td>
<td>10/12/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Contract Foundation</td>
<td>08/12/2001</td>
<td>10/12/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Contract House Facility</td>
<td>10/2/2002</td>
<td>24/12/2002</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Contract House Facility</td>
<td>10/2/2002</td>
<td>24/12/2002</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Contract Foundation</td>
<td>20/1/2001</td>
<td>13/12/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Contract Substructure</td>
<td>20/1/2001</td>
<td>13/12/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Contract Foundation</td>
<td>11/2/2001</td>
<td>09/2/2002</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Contract Substructure</td>
<td>11/2/2001</td>
<td>09/2/2002</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Inspect &amp; Approve Work</td>
<td>14/1/2001</td>
<td>20/12/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Inspect &amp; Approve Work</td>
<td>14/1/2001</td>
<td>20/12/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Inspect &amp; Approve Work</td>
<td>14/1/2001</td>
<td>20/12/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Inspect &amp; Approve Work</td>
<td>14/1/2001</td>
<td>20/12/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Lay Underground Service</td>
<td>09/2/2001</td>
<td>14/12/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Lay Underground Service</td>
<td>09/2/2001</td>
<td>14/12/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Organic Site Resources</td>
<td>10/1/2001</td>
<td>20/11/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Organic Site Resources</td>
<td>10/1/2001</td>
<td>20/11/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Plan of Work Schedule</td>
<td>20/1/2001</td>
<td>30/11/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Plan of Work Schedule</td>
<td>20/1/2001</td>
<td>30/11/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
<tr>
<td>Prepare &amp; Distribute Resources</td>
<td>30/1/2001</td>
<td>01/12/2001</td>
<td>Foundation</td>
<td>20</td>
<td>0.225</td>
<td>0.54</td>
<td>3.402</td>
</tr>
<tr>
<td>Prepare &amp; Distribute Resources</td>
<td>30/1/2001</td>
<td>01/12/2001</td>
<td>Wall</td>
<td>0</td>
<td>1</td>
<td>0.23</td>
<td>2.160</td>
</tr>
</tbody>
</table>

**Figure A.13:** Report ‘Design View’ for product and process information

**Figure A.14:** Report ‘Print Review’ of integrated product and process information
It is appropriate to mention that at this stage of the development process, reports were only used to demonstrate how the information could be implemented. Therefore, only a few forms have been programmed to generate product and process reports, and to integrate both information sources.

A.5 Development of the User Interface (Switchboard) for HIPPY

The basic principles behind an end-user application are the ease of use and a specific focus on the activities the user plans to undertake. When creating a application for a user who wants to concentrate on its purpose and the special tasks it requires, Access facilitates the adding of custom user interfaces, and provides a number of tools that help create a user-friendly, graphical, and customised system for those who have no need or time to study how it works. In the development of the HIPPY user interface, two special forms were added, which were used as switchboards for choosing activities, and as dialog boxes for acquiring user input.

- The main switchboard (user interface) offers a single point of entry into the application with a list of custom activities and, with a click on a switchboard item, the end-user branches to the operation.
- The dialog boxes offer a set of options from which to choose, such as which form or report to open, which filter to apply, or what action to take next.

The user interface is made up of multiple switchboards consisting of forms with customised menu items, and buttons. The buttons facilitate the viewing of records such as next, previous, last, and the opening and closing of forms, which are activated by pre-designed VBA code using the command button wizard, shown in Figure A.15. The switchboard system consists of a hierarchical arrangement of switchboard pages beginning with the main switchboard, and branching out to include six subordinate pages. The specific details of what these lead to, and control is described in Chapter 6. These include:

- Get Started Switchboard: reached by clicking How to Get Started;
- Product Data switchboard: reached by clicking on the View Product Modelling Data;
- Process Information Switchboard: reached by clicking on the View Process Modelling Information;
• Integrated Switchboard, reached by clicking the View Integrated Sources Item;
• The Reports Switchboard: reached by clicking Preview Reports, and
• The Administration Switchboard, reached by clicking the Administration button.

Each page contains a set of items with commands that carry out a specific activity. Most items also include an argument that specifies which form to open, which report to preview, which macro or event procedure to run and so on. To start the switchboard manager, choose Tools / Database Utilities / Switchboard Manager, and the mandatory default main switchboard page appears, as shown in Figure A.15. The Figure displays the main switchboard, the switchboard page, and the adding items to pages dialog boxes.

**Figure A.15: The switchboard manager windows**

Once the switchboards are created, a final option in the main switchboard is to exit the database without exiting Access. This was important to allow the user to close the database and return to the empty Access desktop to open another database or perform other work. The application also includes a welcome screen, called a 'splash screen', which appears for only a short period of time before giving way to the main switchboard (see Section 6.3).
A.6 Development of Welcome Screen and LOGO Form

A welcome screen and LOGO form were developed to provide a welcome to the user and an introduction to some brief information about HIPPY. To limit the length of time over which the welcome forms displayed, an event procedure was attached to the timer interval property and set to 7000 cycles (7 seconds). A CloseSplash macro was also attached to the On Timer event property, to enable the splash screen to close and return to the main switchboard. Finally, a function written in VBA code was attached to the ‘On Close’ property of the form to test the state of the control buttons and the actions of the form, as illustrated in Figure A.16.

![Visual Basic code for welcome screen](image)

Figure A.16: Visual basic code for welcome screen

In addition to the welcome screen, another form was developed to serve as an introduction to the user and provided a brief description of HIPPY. The LOGO form (shown in Figure A.17) is activated by clicking on the ‘How to Get Started’ button on the main switchboard. Once activated, it gives the user brief information about HIPPY, and allows direct entry into the object database window. The LOGO form was developed in a similar way to the welcome form using various macros and VBA code.
A.7 Transferring Data and Objects between Forms

The Windows environment provides a high degree of compatibility between applications and, as a result, the HIPPY platform provides several ways of transferring data between applications. Hyperlinking is one method by which different links can be created between applications and their objects. The Windows clipboard is another method by which different types of data: text (either standard text or DDE link data) and graphics (bitmaps, metafiles, etc) can be moved from one application to another. The drag-and-drop mechanism is another method that allows users to copy, link, or move objects by direct manipulation.

However, it is also possible for the application to communicate with another Windows application and ensure that project data is updated automatically. Two methods are available: dynamic data exchange (DDE) and object linking and embedding (OLE). The DDE method allows links to be created between applications with DDE capabilities. Most, if not all, of the applications used to create the HIPPY prototype had DDE/OLE built in.
The OLE method is an alternative form of data transfer that allows applications or components to be activated from within another program allowing several types of data to be processed. The HIPPY prototype supports two primary aspects of OLE: the compound document architecture and the OLE automation facility.

- The compound document (an aggregation class) is important in that it is the beginning of what is needed to fully enable the kind of environment that supports distributed object-oriented document models. In a document-oriented model, a compound document created by a client application may contain various data objects linked to different server applications (component classes that can be part of an aggregation class). For example, a product model describes the facility, whereby the facility is a house unit that has processes.

- OLE automation distinguishes between OLE clients or containers (programs that use the objects) and OLE servers or source (Window-based applications that expose or send their objects). An OLE server exports an interface that contains methods (an action that is performed by the server on behalf of the OLE client) and properties (characteristics of an object) that may be openly accessible from the OLE server object. The information may be stored in the object or be retrieved by the server object from another source (i.e., Web page, document (flat) file or database).

Many, if not all, of the forms and reports created in HIPPY were developed using objects that can give the HIPPY software remote control over other MS applications (e.g. Excel, Word, PowerPoint, Visio, etc.), allowing these applications to use OLE automation to access the HIPPY database via MS Access. This type of control could extend over just about any other type of application, and is another major discussion point for further research work into OLE-BD and ActivityX data objects (ADO) for transferring data via the Internet.
QUESTIONNAIRE

An Integrated Information System to support the Management of On-Site Construction Operations

HIPPY Evaluation Questionnaire

Could you please complete the following evaluation questionnaire after a presentation and demonstration of the prototype system?

Nature of Participants and Organisation Details

Your Position Please (e.g. project/construction manager, designer, engineer, consultant)

Type of Organisation:

Area of Experience:

Experience in the Construction Industry (years):

Number of Employees (if applicable):

Annual Turnover:
# The Prototype System (HIPPY)

<table>
<thead>
<tr>
<th>Questions</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td><strong>Definition of Product and Process Information</strong></td>
<td></td>
</tr>
<tr>
<td>1  How effectively does the system facilitate the definition of product and process information?</td>
<td></td>
</tr>
<tr>
<td>2  How effectively does the system help in understanding the flow of processes required for constructing a building?</td>
<td></td>
</tr>
<tr>
<td>3  How effectively does the system help in understanding the product components needed to support the processes?</td>
<td></td>
</tr>
<tr>
<td>4  How well does it insure that the product and process information is represented?</td>
<td></td>
</tr>
<tr>
<td>5  How effectively does the system support the integration of product and process information?</td>
<td></td>
</tr>
<tr>
<td>6  How well does the system help in clarifying the browser interfaces between product components and the processes?</td>
<td></td>
</tr>
<tr>
<td>7  To what extent are the defined product and process information unambiguous?</td>
<td></td>
</tr>
<tr>
<td>8  How effectively is the product and process information structured within HIPPY?</td>
<td></td>
</tr>
<tr>
<td><strong>Applicability to the Construction Industry</strong></td>
<td></td>
</tr>
<tr>
<td>9  How appropriate are the application development tools used in the system?</td>
<td></td>
</tr>
<tr>
<td>10 How well does the system architecture support the flow of graphical (modelling) information?</td>
<td></td>
</tr>
<tr>
<td>11 How useful is the system to the overall on-site construction process?</td>
<td></td>
</tr>
<tr>
<td>12 How useful is the system to the construction team?</td>
<td></td>
</tr>
<tr>
<td>13 How effectively will the system increase the speed of the information flow during the on-site construction stage?</td>
<td></td>
</tr>
<tr>
<td>14 To what extent does it represent an improvement (or help) existing on-site information systems (how suitable is it)?</td>
<td></td>
</tr>
<tr>
<td>15 How convinced are you that on-site construction professionals will accept (or use) the system?</td>
<td></td>
</tr>
<tr>
<td><strong>Management of the System</strong></td>
<td></td>
</tr>
<tr>
<td>16 How well organized (designed) is the system architecture?</td>
<td></td>
</tr>
<tr>
<td>17 How effectively are changes to the information managed?</td>
<td></td>
</tr>
<tr>
<td>18 How well integrated are the components of the system?</td>
<td></td>
</tr>
<tr>
<td><strong>Efficiency of Teamwork</strong></td>
<td></td>
</tr>
<tr>
<td>19 How well does the system facilitate the communication among members of the construction team?</td>
<td></td>
</tr>
<tr>
<td>20 How effectively will it facilitate a common understanding of the information among the construction team?</td>
<td></td>
</tr>
<tr>
<td>21 To what extent would the output of the system facilitate</td>
<td></td>
</tr>
<tr>
<td>Design? or construction creativity?</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td></td>
</tr>
</tbody>
</table>

### General

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>How easy is the system to use?</td>
</tr>
<tr>
<td>23</td>
<td>How generic do you consider the system to be?</td>
</tr>
<tr>
<td>24</td>
<td>How confident are you with computer information systems in general?</td>
</tr>
<tr>
<td>25</td>
<td>What is your overall rating of the prototype system?</td>
</tr>
</tbody>
</table>

### General Comments

1. Bearing in mind that the system is a prototype; in what way can the system be improved?

   
   
   
   

1. Further Comments:

   
   
   
   

Thank you for participating in this questionnaire.
Appendix 3

Reports and Publications Arising from the Research


