School of Information Systems

The Development of the Business Rules Diagram

Donald Chisholm McDermid

This thesis was presented as part of the requirements for the
award of the Degree of Doctor of Philosophy
of the
Curtin University of Technology

July, 1998

Abstract

This thesis concerns the development of a diagramming technique which assists in the specification of information systems requirements. The technique is called the **Business Rules Diagram (BRD)** although earlier versions were given different names. The term **development** in the title of this thesis is defined here to include both the work involved in **designing** the BRD as well as **testing** its usefulness. So, the scope of this research covers research activity starting from the original idea for the diagram through to testing its usefulness.

Action research was the research method used. In all, two major action research studies were undertaken. The first involved working with an analyst only. The second involved working with an analyst and users.

Dedication

This thesis is dedicated to my mother, Winifred, whose love, patience, wisdom and stamina have been a constant source of inspiration to me.

19th July, 1998.

Acknowledgements

There are a number of people I should like to thank for their support throughout this project. Firstly, I'd like to thank Peter Marshall, my supervisor, for his help and for his unshakeable belief in me and in my ability to undertake this work. Trying to do this work on top of a full-time academic load was a challenge to say the least, but was made a lot easier by Peter's flexible yet constructive approach. I'd also like to thank Judy McKay for her willingness and ability to explore a whole range of issues with me - literally everything from past participles to epistemology! Still on the academic side I'd like to thank Paul Swatman, especially for a strong intellectual idea he provided and also John Gammack for his timely advice and his uncanny ability to get to the heart of a matter.

Thanks also go to my co-researchers who, though un-named, are VIPs as far as this research is concerned.

Last, but not least, I'd like to thank my wife Marion who once again has endured considerable sacrifice to allow me to do what I wanted to do. Thank you Marion, you deserve any accolade more than I do.

Table of Contents

The Development of the Business Rules Diagram 1
Dedication ii
Acknowledgements iii
Table of Contents iv
List of Figures and Tables x
Chapter 1 Introduction 1
1.1 Overview 1
1.2 Scope of Research 1
1.3 Brief History of the Diagram 2
1.4 The Use of Action Research 4
1.5 Cohesion of Thesis 5
1.6 Presentation Style 6
1.7 Remainder of thesis 7
Chapter 2 The Business Rules Diagram (version 3) 11
2.1 Overview 11
2.2 Context of the BRD 12
2.3 What is a Business Rule? 13
2.3.1 Data Models 15
2.3.2 Process Models 16
2.3.3 Behaviour Models 16
2.4 Theoretical Bases of the Business Rules Diagram 21
2.4.1 Flowcharts 21
2.4.2 Finite State Diagrams and Petri-nets 24
2.5 Constructing the Business Rules Diagram 28
2.5.1 Identify Candidate Business Rules (Policy Rules) 28
2.5.2 Identify Candidate Business Events and Signals 29
2.5.3 Identify Candidate Objects in Problem Situation 30
2.5.4 Construct Object Life Histories for each Object Identified 32
2.5.5 Construct User Business Rules Diagrams 35
2.5.6 Construct the Business Rules Diagram 40
2.6 Validate Business Rules Diagram 46
2.6.1 Structured Walkthrough 46
2.6.2 Examining Abnormal Lives 47
2.6.3 Analysing Amendments 48
2.7 Summary 49
Chapter 3 Research Question and Significance of the Research 50
3.1 Overview 50
3.2 Research Question 51
3.2.1 Specific Constructs of State, Event, Condition, Trigger and Message 54
3.2.2 Policy Rules 54
3.2.3 Context of a Business Rule 54
3.2.4 Direct Modelling of Business Rules 54
3.2.5 Construction Process 55
3.2.6 Notational Review 55
3.2.7 Event Specification Table 55
3.2.8 Modelling of Static Rules 55
3.2.9 Clarifying OLHs (Object Life Histories) 55
3.2.10 UBRD (User Business Rules Diagram) 56
3.2.11 Structured Walkthrough 56
3.2.12 Abnormal Life Analysis 56
3.2.13 Amendment Analysis 56
3.2.14 Modelling at the Instance Level 56
3.2.15 Significant Modifications 57
3.3 This Research demonstrates how to Design Diagrams using Action Research 57

3.3.1 Limitations of Previous Research 57
3.3.2 The Contribution of This Research 58
3.4 Structural Aspects of the BRD 59
3.4.1 Structured Representation of a Business Rule 60
3.4.2 Comprehensive Definition of a Business Rule 63
3.5 Construction Aspects of the BRD 63
3.5.1 A Third Generic Approach to Systems Development 64
3.6 The BRD has Features considered useful in resolving Difficulties in IRD 64
3.6.1 Features of the BRD which aid IRD 65
3.6.2 Problems in IRD 67
3.7 Summary 71
Chapter 4 Literature Review 72
4.1 Overview 72
4.2 Extended State-based Models 74
4.2.1 Booch 76
4.2.2 Shlaer and Mellor 76
4.2.3 Rumbaugh et al. 76
4.2.4 Petri-net+ 77
4.2.5 Evaluation of Extended State-based Models 77
4.3 Condition-based models 79
4.3.1 Herbst 79
4.3.2 Tsalgatidou and Loucopoulos 80
4.3.3 Sandy 80
4.3.4 CPM 80
4.3.5 Evaluation of Condition-based Models 81
4.4 Extended Static Models 82
4.4.1 Kappel and Schrefl 82
4.4.2 Kung and Solvberg 83
4.4.3 Evaluation of Extended Static Models 83
4.5 Trigger-based Models 84
4.5.1 Joosten 84
4.5.2 Flynn and Davarpanah Jazi 85
4.5.3 Evaluation of Trigger-based Models 85
4.6 Event-based Models 86
4.6.1 Dataflow diagram 86
4.6.2 Martin and Odell 86
4.6.3 Evaluation of Event-based Models 87
4.7 Summary 87
Chapter 5 Research Methodology 90
5.1 Overview 90
5.2 Consider Research Paradigm in Relation to Research Question 91
5.3 Implications of Research Question and Interpretivistic Paradigm 93
5.3.1 Real-world Context 94
5.3.2 Social and Technical Domains of Interest 94
5.3.3 Iterative Learning 94
5.3.4 Explicit Learning 95
5.3.5 The Degree of Involvement of Participants 95
5.4 Create Short-list of Research Approaches 96
5.4.1 Laboratory Experiment/Study 96
5.4.2 Field Experiment/Study 97
5.4.3 Case Study 97
5.4.4 Action Research 98
5.4.5 Descriptive/Interpretive Research 98
5.4.6 Short-list 98
5.5 Confirmation of Research Approach 99
5.6 Summary 103
Chapter 6 Research Design 104
6.1 Overview 104
6.2 Framework for Learning 105
6.2.1 Chosen Intellectual Framework 106

6.2.2 Relationship of Intellectual Framework to Research Question ... 108
6.2.3 Learning Cycle ... 110
6.3 Conceptualisation of the Research ... 112
6.3.1 Types of Action Research ... 112
6.3.2 Process Models of Action Research ... 114
6.4 Structure of the Design ... 118
6.4.1 First Action Research Study ... 118
6.4.2 Critical Review of Research Approach ... 120
6.4.3 Second Action Research Study ... 122
6.5 Data Collection and Analysis Techniques ... 123
6.5.1 Semi-structured Interview ... 124
6.5.2 Content Analysis ... 125
6.5.3 Documentary Analysis ... 126
6.5.4 Field Notes ... 127
6.6 Summary ... 128
Chapter 7 Action Research Study 1 The Business Event Diagram ... 129
7.1 Overview ... 129
7.2 Structure of Action Research Study 1 ... 129
7.3 Client-System Infrastructure ... 135
7.3.1 The Analyst ... 135
7.3.2 The Role of Researcher and Systems Analyst ... 136
7.4 The Business Application ... 136
7.4.1 Portbilling Application ... 137
7.4.2 Invoicing Application ... 137
7.5 Preparation Step of Developing the BED ... 138
7.5.1 Event Focus (1P1) ... 138
7.5.2 Object-oriented basis (1P2) ... 138
7.5.3 Group Event Notation (1P3) ... 139
7.5.4 Delay Analysis Step (1P4) ... 139
7.5.5 Event Notation Refinement (1P5) ... 140
7.5.6 Evaluation of the Preparation Step against Intellectual Framework ... 140
7.6 Application Step of Developing the BED ... 142
7.6.1 Flaw in Prerequisite Table Notation (1A1) ... 143
7.6.2 The Graphical Implications of Union and Intersection (1A2) ... 144
7.6.3 Alternative Point of Entry Problem (1A3) ... 145
7.6.4 Fan Out Dependency (1A4) ... 146
7.6.5 Amendment to the Notation for Contingent Dependency (1A5) ... 147
7.6.6 Simultaneity Problem (1A6) ... 147
7.6.7 Evaluation of Application Step against Intellectual Framework ... 148
7.7 Review Step of Developing the BED ... 150
7.7.1 Information Systems Events versus Business Events (1R1) ... 150
7.7.2 Need for Rule Completeness (1R2) ... 151
7.7.3 States versus Events (1R3) ... 152
7.7.4 Evaluation of Review Step against Intellectual Framework ... 154
7.7.5 Discussion in terms of Wider Literature ... 156
7.8 Decision Step of Developing the BED ... 156
7.8.1 Change Notation (1D1) ... 158
7.8.2 Change to State (1D2) ... 158
7.9 Summary of Progress with the BED ... 159
Chapter 8 Action Research Study 1 The Business State Diagram version 1 ... 161
8.1 Overview ... 161
8.2 Preparation Step of the Development of the BSDv1 ... 162
8.2.1 The Simultaneity Problem (2P1) ... 162
8.2.2 Evaluation of the Preparation Step against Intellectual Framework ... 163
8.3 Application Step of the Development of the BSDv1 ... 163
8.3.1 The Simultaneity Problem (2A1) ... 163
8.3.2 The Overkill Problem (2A2) ... 165
8.3.3 Evaluation of Application Step against Intellectual Framework ... 166
8.4 Review Step of the Development of the BSDv2 ... 167
8.4.1 Readability (2R1) ... 169

8.4.2 The Need for Rule Completeness (2R2) 169
8.4.3 Manyness (2R3)…... 170
8.4.4 Evaluation of Review Step against Intellectual Framework 171
8.4.5 Discussion in terms of Wider Literature…. 171
8.5 Decision Step of the Development of the BSDv2 171
8.5.1 Extensions to the Prerequisite Table (2D1) 171
8.5.2 Complex Objects (2D2) 172
8.5.3 The Harel blob (2D3) 172
8.5.4 Evaluation of Decision Step against Intellectual Framework 172
8.6 Summary of Progress with the BSDv1 173
Chapter 9 Action Research Study 1 The Business State Diagram version 2 175
9.1 Overview 175
9.2 Application Step of the Development of the BSDv2 175
9.2.1 Readability (3A1) 175
9.2.2 Evaluation of Application Step against Intellectual Framework 179
9.3 Review Step of the Development of the BSDv2 180
9.3.1 Complexity (3R1) 180
9.3.1 Simultaneity (3R2) 181
9.3.3 Rule Completeness (3R3) 184
9.3.4 Construction from Complex Objects (3R4) 185
9.3.5 Evaluation of Review Step against Intellectual Framework 185
9.3.6 Discussion in terms of Wider Literature…. 185
9.4 Decision Step of the Development of the BSDv2 186
9.4.1 Modelling in both States and Events (3D1) 186
9.4.2 Evaluation of Decision Step against Intellectual Framework 187
9.5 Summary of Progress with the BSDv2 187
Chapter 10 Action Research Study 1 The Business Rules Diagram version 1 189
10.1 Overview 189
10.2 Application Step of the Development of the BRDv1 189
10.2.1 The Simultaneity Problem (4A1) 192
10.2.2 Readability and the Simple Event (4A2) 192
10.2.3 Evaluation of Application Step against Intellectual Framework 193
10.3 Review Step of the Development of the BRDv1 193
10.3.1 Complexity (4R1) 193
10.3.2 Rule Completeness (4R2) 194
10.3.3 Information System Events versus Business Events (4R3) 196
10.3.4 The Manyness Problem (4R4) 197
10.3.5 Direct versus Indirect Construction (4R5) 198
10.3.6 Evaluation of the Review Step against Intellectual Framework 198
10.4 Decision Step of the Development of the BRDv1 199
10.4.1 Show Conditions and Triggers on the Diagram (4D1) 199
10.4.2 Prerequisite Table (4D2) 199
10.4.3 Refine Definition of Rule (4D3) 199
10.4.4 Direct Approach (4D4) 200
10.5 Summary of Progress with the BRDv1 200
10.6 Preparation of the Business Rules Diagram version 2 201
10.6.1 Business Rules versus Information Systems Rules (5P1) 202
10.6.2 Trigger Selection (5P2) 203
10.6.3 Harel blob for simple conditions (5P3) 203
10.6.4 Sub-States and Parallelism (5P4) 205
10.6.5 Event Specification Guide (5P5) 205
10.6.6 Evaluation of Preparation Step against Intellectual Framework 206
10.6.7 Discussion in terms of Wider Literature…. 206
10.7 Summary of Progress with the BRDv2 207
Chapter 11 Critical Review of Research Approach 208
11.1 Overview 208
11.2 Review of Framework F 210
11.2.1 Mapping (M) 214
11.2.2 Complexity (C) 216
11.2.3 Process (P) 221

11.2.4 Abstraction (A) ... 223
11.2.5 Notation (N) ... 229
11.2.6 Summary of Framework Review ... 230
11.3 Review of Methodology M ... 231
11.3.1 Distinction between Methodological and Ontological Ideas ... 232
11.3.2 Reflections on the Application of the Methodology ... 232
11.4 Review of Area of Application A ... 235
11.5 Summary ... 237
Chapter 12 Action Research Study 2 ... 239
12.1 Overview ... 239
12.2 Goals and Structure of Action Research Study 2 ... 239
12.3 Client/System Infrastructure ... 243
12.3.1 Analyst Y ... 244
12.3.2 User A ... 245
12.3.3 User B ... 245
12.3.4 The Business Application ... 245
12.4 Preparation of the Business Rules Diagram version 3 ... 246
12.4.1 Categorising Events, Triggers and Messages (6P1) ... 246
12.4.2 Flowchart 'Look and Feel' to the UBRD (6P2) ... 247
12.4.3 The Abolition of Sub-states (6P3) ... 249
12.4.4 The One to Many Business Rule (6P4) ... 250
12.4.5 The Abolition of the Simple Event (6P5) ... 253
12.4.6 Refining the Harel Blob (6P6) ... 253
12.4.7 Changes to the Event Specification Guide (6P7) ... 254
12.4.8 Evaluation of the Preparation Step against Intellectual Framework ... 254
12.5 Application Step of Action Research Study 2 ... 257
12.5.1 User-led UBRD Construction (6A1) ... 258
12.5.2 Flowchart Loops in the UBRDs (6A2) ... 261
12.5.3 Processing Rules in the UBRDs (6A3) ... 261
12.5.4 Evaluation of Application Step against Intellectual Framework ... 261
12.6 Review of Step of Action Research Study 2 ... 263
12.6.1 The need for guidance (6R1) ... 263
12.6.2 The Need for Validation (6R2) ... 266
12.6.3 The level of abstraction problem (6R3) ... 268
12.6.4 Evaluate Review Step against Intellectual Framework ... 277
12.6.5 Discussion in terms of Wider Literature ... 278
12.7 Summary ... 279
Chapter 13 Analysis of Findings ... 284
13.1 Overview ... 284
13.2 Introduction to what has been learnt about Business Rules ... 285
13.3 A Review of the Business Rule. ... 287
13.4 A Review of the Policy Rule ... 292
13.5 A Review of the User Business Rules Diagram ... 297
13.5.1 The Need to have Separate User Diagrams ... 297
13.5.2 The Need for a Flexible Approach with the UBRD ... 300
13.5.3 The Need for Guidance (6D1) ... 303
13.5.4 The Need for Validation (6D2) ... 304
13.6 The Business Rules Diagram ... 307
13.6.1 Notational Issues ... 307
13.6.2 Harel blob ... 310
13.6.3 Sub-states ... 313
13.6.4 Construction Issues ... 315
13.6.5 Complexity ... 318
13.7 The Event Specification Table ... 321
13.7.1 The Emerging Role of the EST ... 322
13.7.2 Manyness ... 325
13.7.3 Simultaneity ... 327
13.8 Critique of Action Research and Intellectual Framework ... 330
13.8.1 Arguments against using a Framework ... 331
13.8.2 Arguments for using a Framework ... 334

13.8.3 Synthesis of a Revised Framework ... 337
13.8.4 Conclusions from this Critique ... 341
13.9 Summary ... 342
Chapter 14 Future Research ... 345
14.1 Overview ... 345
14.2 Pre-BRD Issues ... 348
14.3 The User Business Rules Diagram ... 349
14.4 The Business Rules Diagram ... 350
14.4.1 Parallel states ... 351
14.4.2 Sequentiality in a business rule ... 351
14.4.3 Harel blob ... 352
14.4.4 Simultaneity ... 352
14.5 The Event Specification Table ... 353
14.6 Validation ... 353
14.7 Post-BRD ... 355
14.8 Framework of Ideas ... 355
14.9 Ontological Ideas ... 357
14.10 Methodology ... 358
14.11 Area of Application ... 359
14.12 Conclusion ... 361

Bibliography ... 364

Appendices:

1. Versions of Diagram
2. Transcript of 1st Semi-structured Interview
3. Documentation of Portbilling Application
4. Documentation of Invoicing Application
5. Transcripts of 2nd Semi-structured Interviews
6. Minutes of Meetings from Action Research Study 2
7. Description of Order Processing Case Study
8. Preliminary Data Analysis of Action Research Study 2

List of Figures and Tables

Figure 1.1 Map of Studies in this Research to Versions of the Diagram ... 9
Figure 2.1 Levels of Abstraction of Business rules ... 18
Figure 2.2 Example of a Flowchart ... 22
Figure 2.3 Finite State Diagram for Customer Scenario ... 25
Figure 2.4 Petri-net for Customer Scenario ... 26
Figure 2.5 Comparison between Finite State Diagram and Petri-net ... 27
Table 2.1 Candidate Business Rules for Sample Case ... 29
Table 2.2 Candidate List of Business Events and Signals. ... 30
Figure 2.6 OLHs for Candidate Objects ... 33
Figure 2.7 Illustration of Harel Depth ... 34
Figure 2.8 Notational Constructs for Business Rules Diagram ... 36
Figure 2.9 Accept Order UBRD ... 37
Figure 2.10 Create New Order from Outstanding Order UBRD ... 38
Figure 2.11 Invoice UBRD ... 39
Figure 2.12 Credit Note UBRD ... 39
Figure 2.13 Payment UBRD ... 40
Figure 2.14 The Business Rules Diagram ... 42
Table 2.3 Event Specification Table ... 45
Figure 3.1 Cognitive Map summarising Significance of Research ... 51
Table 3.1 Summary of Features of BRDv3 ... 53
Table 4.1 Feature Analysis of BRD and Extended State-based Models ... 77
Table 4.2 Feature Analysis of BRD and Condition-based Models. ... 81
Table 4.3 Feature Analysis of BRD and Extended Static Models ... 83
Table 4.4 Feature Analysis of BRD and Trigger-based Models ... 85
Table 4.5 Feature Analysis of BRD and Event-based models ... 87
Figure 5.1 Cognitive Map indicating Steps in Research Methodology and Design ... 91
Table 5.1 A Review of Action Research Characteristics from the Literature ... 100
Figure 6.1 Cognitive Map indicating Steps in Research Methodology and Design ... 104
Figure 6.2 The Organised Use of Rational Thought, after Checkland, 1985 ... 106
Figure 6.3 Mapping of Scripts from Real World to Machine World ... 107
Figure 6.4 Five Scenarios in Ontological Framework, after Wand and Weber (1993) ... 109
Figure 6.5 Flowchart of Steps in Research Process, adapted from Checkland (1985) ... 111
Table 6.1 Types of Action Research Method ... 113
Figure 6.6 Lewin's Cyclic Model, adapted by Burns (1990) ... 115
Figure 6.7 Sociotechnical Action Research Cycle, after Susman and Evered, 1978 ... 116
Figure 6.8 Kolb's Experiential Learning Cycle, after Kolb et al. (1979) ... 116
Table 6.2 Mapping of Process Steps in Different Action Research/Learning Models ... 117
Figure 6.9 Framework F, using one of M1, M2, M3 or M4 applied to Application Area A in the First Research Study ... 119
Figure 6.10 Learning Cycle in Action Research Study 1 ... 120
Figure 6.11 Changes resulting from Critical Review ... 121
Figure 6.12 Feedback Paths in Second Action Research Study ... 123
Figure 7.1 Main Steps in Action Research Process ... 130
Figure 7.2 Research Progress Map for Action Research Study 1 ... 132
Table 7.1 Codes in Research Process ... 133
Figure 7.3 Revised Mapping showing Transitive Mappings ... 141
Figure 7.4 Attempt at Showing Alternative Construction Routes ... 141
Table 7.2 Prerequisite Table for BED ... 142
Figure 7.5 First-cut Business Event Diagram – Portbilling System ... 143
Figure 7.6 Two Heterogeneous Event Situations ... 144
Figure 7.7 Situation B Resolved ... 145
Figure 7.8 Straightforward Dependency and Fan Out Dependency ... 146
Figure 7.9 Contingent Dependency ... 147
Figure 7.10 Final Business Event Diagram – Portbilling System ... 149
Figure 7.11 Separation of Business and Information System Events ... 154

Table 7.3 The Bunge-Wand-Weber Representation Model, after Wand and Weber (1995)......156
Table 8.1 Prerequisite Table for BSDv1 164
Figure 8.1 Business State Diagram version 1 – Portbilling System 165
Figure 9.1 First Attempt at using the Harel Blob........172
Figure 9.2 Business State Diagram with Harel Depth........174
Table 9.1 Prerequisite table for Portbilling Application 181
Figure 9.3 Business State Diagram version 2 with Codes........ 179
Figure 10.1 Business Rules Diagram with Harel Depth........185
Table 10.1 Prerequisite Table for Portbilling Application........ 191
Figure 10.2 Trade-off in achieving an acceptable Level of Complexity........ 197
Figure 10.3 The Business Rules Diagram........ 199
Figure 10.4 Extension of Mappings from the Ontological World........ 206
Figure 11.1 Research Progress Map for the Critical Review 210
Figure 11.2 Extension of Wand and Weber's Model........ 210
Table 11.1 Allocation of Issues into Categories 214
Table 11.2 Research Codes Classified under Mapping (M) 214
Table 11.3 Research Codes Classified under Complexity 216
Table 11.4 McCabe's Complexity Factor for each Version of the Diagram........ 217
Table 11.5 Rossi's Complexity Factor for each Version of the Diagram 219
Table 11.6 Research Codes Classified under Process (P)........ 221
Figure 11.3 Structural Decomposition in Glasson's Model 222
Figure 11.4 Development Sequence in Glasson's Model 222
Table 11.5 Research Codes Classified under Abstraction 225
Figure 11.5 Two Different Business Rules for Deleting a Message 226
Figure 11.6 Extended Model showing Elaboration Problem 228
Table 11.7 Research Codes Classified under Notation........ 229
Figure 12.1 Research Progress Map for Action Research Study 2 241
Figure 12.2 Singularity within an Object........ 251
Figure 12.3 Multiplicity between Objects........ 252
Figure 12.4 Bar on Multiplicity between Objects 252
Figure 12.5 UBRD from Case Study 255
Figure 12.6 Harel Blob Equivalent of UBRD 256
Table 12.1 Generic Complexity Factors for UBRD........ 256
Figure 12.7 Cognitive Map supporting the Need for Guidance 264
Table 12.2 Characteristic Excerpts suggesting the Need for Guidance 265
Figure 12.8 Cognitive Map supporting the Need for Validation........ 267
Table 12.3 Characteristic Excerpts suggesting the Need for Validation........ 268
Table 12.4 Specific Statements from Interview in which Omissions are identified 269
Figure 12.9 State-event Network for Levels of Abstraction Problem 270
Table 12.5 Feedback from Interviews on Level of Abstraction Problem 271
Figure 12.10 Alternative BRD for the Registration Sub-system 273
Figure 12.11 Pure BRD for the Registration Sub-system........ 274
Figure 13.1 Model of Main Components of the BRD Technique 286
Figure 13.2 Research Progress Map leading to Need for Four Constructs in the BRD........ 289
Figure 13.3 A Business Rule in the Form of a UBRD 292
Figure 13.4 Research Progress Map leading to Consolidation of Policy Rules 294
Figure 13.5 Summary of History of Decision leading to a Separate User Diagram........ 298
Figure 13.6 Simple UBRD from Portbilling Application 299
Figure 13.7 Complex UBRD from Order Processing Case Study 300
Figure 13.8 Research Progress Map for Modelling Issues associated with UBRD 301
Figure 13.9 State-event Network for Levels of Abstraction Problem 302
Figure 13.10 Cognitive Map supporting the Need for Guidance 303
Figure 13.11 Cognitive Map supporting the Need for Validation........ 305
Figure 13.12 Notational Issues associated with the BRD 309
Figure 13.13 History of Issues associated with the Harel Blob 311
Figure 13.15 Construction Issues associated with the BRD........ 316
Figure 13.16 Cognitive Map supporting Conclusion that a correct partial BRD could be drawn.... 317
Figure 13.17 History of Main Issues associated with Complexity 319
Figure 13.18 Cognitive Map supporting the Conclusion to retain pure BRD 320
Figure 13.19 History of Issues associated with the Event Specification Table 322

Figure 13.20	History of Issues associated with Manyness	326
Figure 13.21	History of Issues associated with Simultaneity	329
Figure 13.22	Cognitive Map showing Arguments against using a Framework	332
Figure 13.23	Cognitive Map showing Arguments for using a Framework	334
Table 13.1	Mapping Issues identified during Research	335
Figure 13.24	Synthesised Framework for Action Research Study	339
Figure 14.1	Headings for Further Research	346
Table 14.1	Summary of Sections in this Chapter	347
Table 14.2	List of Research Topics outlined in this Chapter	362

Chapter 1

Introduction

1.1 Overview

The aim of this introductory chapter is to provide sufficient background on a number of issues relating to this research project so that the reader can obtain the best advantage from the remaining chapters. Necessarily an introduction should be brief, yet at the same time it should to serve as an overview for what is to come in the thesis. Consequently, there is a danger of providing perceived insufficient treatment of topics while attempting to give an impression of the breadth of the thesis. Also in this introduction some of the sections deal with matters which need to be stated at the outset so that the reader is aware of any standard or style that has been adopted.

Preview topics covered in this introduction are the scope of research, a brief history of the diagram and the use of action research. Topics which require some explanation in terms of standards or style adopted are writing style and issues to do with cohesion in the presentation of this thesis. This chapter ends with a section which signposts the remaining chapters in this thesis.

1.2 Scope of Research

This thesis concerns the development of a diagramming technique which assists in the specification of information systems requirements. The technique is called the **Business Rules Diagram (BRD)** although earlier versions were given different names such as the Business Event Diagram and the Business State Diagram.

The term **development** in the title of this thesis is defined here to include both the work involved in **designing** the BRD as well as **testing** its usefulness. This latter activity is referred to as testing rather than validation, since the term validation is used elsewhere in this thesis to describe a standard activity in systems analysis, ie the act of confirming the correctness of the diagram with users. So, the scope of this research covers research activity starting from the original idea for the diagram through to testing its usefulness.

1.3 Brief History of the Diagram

In 1990, I published a paper in the Software Engineering Journal entitled the 'State Dependency Diagram' (McDermid 1990a), the major advantage of which was that it enabled a fuller specification of an information system by identifying temporal precedence between states in different entities. One weakness of the paper was that the arguments presented were what Galliers (1990) describes as 'argumentative/subjective'. Perhaps this was somewhat typical of many papers published in that era regarding new techniques. For instance Fitzgerald (1991, p663-4) remarks while referring to the research literature on new techniques that *'They were usually simply described and sometimes argued to be powerful for a variety of reasons, but this was stated rather than proved'.* In other words, authors would often speculate for example about the usefulness of their new diagram without having any real evidence to support such conjecture. Clearly, such a situation is far from ideal. Although the communication and generation of new ideas is important, it follows that the testing of ideas by some accepted and formal means is absolutely crucial. The original focus and intention of the research was therefore to test the State Dependency Diagram in a rigorous and thorough manner.

A major benefit of the State Dependency Diagram was to provide a more complete specification of the inter-relationship between states across entities as well as within entities. So the main thrust of the research was seen as one which would test how well this was achieved and to what extent this contributed to a more complete specification. Although I had expected that perhaps minor revisions to the State

Dependency Diagram might be required, I saw the work very much as a refinement of the original diagram. However as I began to use the diagram on real applications, I found that the original premises upon which the ideas were based were not rich enough to adequately express the (perceived) problem. This caused me to reconsider and sometimes change my initial views and assumptions. So, contrary to expectation, the research became developmental as well as involving a component of testing.

Over the research project a number of variations of the diagram were developed. The first three of these were called the Business Event Diagram (BED), the Business State Diagram version 1 (BSDv1) and the Business State Diagram version 2 (BSDv2) respectively. As the names suggest the main modelling focus in these diagrams was either the event or the state. However, each variant seemed to be lacking in some significant way. The fourth diagram attempted to address this by modelling both states and events as a dual focus. The combination of both states and events on one diagram began to model what was considered the essence of a business rule and so this version was called the Business Rules Diagram version 1 (BRDv1). At this juncture, I speculated that by using object-orientation in a simple way with users I could capture 'business rules' at a very early stage in the life cycle, indeed even before identifying the attributes and operations of objects. So the steps in constructing a 'Business Rules Diagram' could be quite different from the route taken in earlier versions. It was also clear that what I was really exploring were rules which govern a business rather than the specification of the inter-relationship between states or events in an information system and so the activity of Information Requirements Determination (IRD) became a far more central focus in this research than anticipated. Thus this represented a major shift in focus from the original intent of the thesis. The two subsequent versions of the diagram, namely the Business Rules Diagram version 2 and version 3, therefore tried to reflect these changes in focus and in development route which in turn determined how the research was then pursued.

In summary, what began as essentially a confirmatory exercise evolved into a developmental project involving further new ideas and then testing them out.

Moreover, a major shift in emphasis of the project occurred when it was decided to focus on business rules.

1.4 The Use of Action Research

Action research has been used as the vehicle for this research. Action research has had a somewhat chequered history both within the information systems community and in the social sciences generally (Baskerville and Wood-Harper, 1992). However, I have found it a flexible approach facilitating insight of a quality which is difficult to imagine obtaining from other approaches. For instance, I doubt whether I would have arrived at the conclusions which caused me to shift the emphasis to business rules if action research had not been used.

Metaphors such as 'journey' and 'vehicle' I believe are appropriate descriptors for action research. For example the metaphor of a journey can be seen in the way that I designed different versions of the diagram. Feedback on the effectiveness or otherwise of one version was used as input to creating a different version. This iterative cycle was felt to be particularly powerful for generating and testing improved versions of the BRD and moving from one version to the next, there was a definite feeling of a journey or exploration being undertaken.

Like any research approach there is no single panacea for all problems and action research is no exception. However, if executed effectively in a suitable context I believe that action research has an inherent flexibility which makes it an attractive choice for researching new diagrammatic techniques. This position is summed up in Baskerville and Wood-Harper (1992) who state that *'We cannot study a newly invented technique without intervening in some way to inject the new technique into the practitioner environment, ie "go into the world and try them out" (Land quoted in Wood-Harper, 1989). This leads us to conclude that action research is one of the few valid research approaches that researchers can legitimately employ to study the effects of specific alterations in systems development methodologies. It is both relevant and rigorous.'*

In this research I have shown how to use action research for designing new diagrams. In this respect the research may be regarded as a unique contribution to knowledge as I can find no evidence in the literature in which this has been previously undertaken. In addition, Checkland strongly recommends the use of an intellectual framework with action research (Checkland 1985). This means that the action research activity is made more explicit by identifying a framework of ideas and then using these ideas to frame the research and later evaluate it. I have used ideas based on the work of Wand and Weber (1993) in this regard. Their ideas, I shall demonstrate, can be used by designers of diagrams to help guide diagram development. Later chapters will discuss such matters related to action research in more detail.

1.5 Cohesion of Thesis

The presentation of this thesis requires clarification for two reasons. Firstly, the shift of emphasis to modelling business rules causes a presentational difficulty in the sense that for example there are two literature reviews that could be presented. One literature review would mainly focus on the literature relating to the State Dependency Diagram and its potential role in the systems development life cycle. This would reflect the position and intent at the beginning of this project. A second literature review would focus on business rules modelling and more generally information requirements determination. Though there may be some overlap, clearly two reviews could be presented. Further, the same problem occurs when discussing the significance of the research, research methodology and research design. Secondly, the process of research does not run serially, ie in practice progress does not happen in an orderly manner. Occasionally there may be a small discrete flash of insight; more often ideas gradually mature over time in an ad-hoc almost disorganised way. In my view it is debatable whether it is helpful to the reader to attempt to report accurately every detail of what actually happened in the order that it happened. Much of this would get in the way of showing what of value has been obtained from the research.

The following approach to presentation has been adopted. I believe the main task of a thesis is to ‘inform and explain’. By informing I mean that the reader is appraised of what was planned, what was executed and what was concluded. By explaining I mean how and why conclusions were arrived at. I consider it necessary therefore to provide sufficient background in order for the reader to be informed about the research and to have adequate explanations about how and why the conclusions were made. However, I do not believe it is helpful or prudent to submit every detail of the work undertaken. Consequently, I have deleted for example in the literature review, materials not directly relevant to business rules modelling but which were relevant to the original intent of this thesis and similarly I have focussed discussion in the chapters on significance of the research, research methodology and research design on the basis that business rules modelling is the central theme of this thesis. I undertake this in the hope that a cohesive thesis can communicate better what is important in this research project.

1.6 Presentation Style

There are two matters which come under this heading. The first is the use of the first person in sentences. Traditionally the academic literature has used the third person. Arguably this leads to a rather terse formal style; it is certainly the style that we have been accustomed to in the information systems literature generally. However, there is an alternative view. Myers (1994) makes the plea that since interpretive research by definition involves the reflections of the researcher, the use of the first person is desirable because it signals that subjective processes are at work much more so than does the use of the third person. Since the academic community is perhaps split on this matter, this leaves the writer of a thesis in a dilemma. Without knowing the predilections of one’s examiners, the writer is in danger of alienating the reader especially if the reader happens to hold a strong view on the matter. I have adopted the following approach in this thesis. Where I am working relatively objectively for example in the reporting of facts or events I shall use the third person as this is the most efficient mode for such work. Where I am reporting my observations, feelings or using my judgement in an interpretive way, I shall use the first person to signal to the reader its subjective or reflective nature.

The second matter in relation to presentation style is to do with referring to the various versions of the diagram. The following set of rules is used throughout the thesis. Firstly, there are a significant number of acronyms in the thesis. Any acronym used will be defined the first time it is used within a chapter. Thus the reader should have to look back no further than the preceding pages of that current chapter to obtain a definition. Secondly, specific versions of the diagram will be so annotated unless it is obvious from the context of the sentence which version is being referenced. For example to reference a particular version, say the first version of the Business Rules Diagram, the acronym BRDv1 will be used. However, often there is a need to refer to the genre of diagrams rather than a specific diagram. Here the term 'diagram' will be used to refer to the genre. Again it should be quite obvious from the context that the genre is what is being referred to rather than a particular version. This leads to a third point. The term Business Rules Diagram as it appears in the title of this thesis is strictly speaking not a single diagram. The Business Rules Diagram as it now stands is more correctly a technique involving the construction of three deliverables, ie two sets of diagrams and one table. Where the whole technique is being referred to, the term technique or the generic term BRD will be used.

1.7 Remainder of thesis

The remainder of the thesis will explore and analyse what has been introduced above in a more rigorous and detailed manner. In chapter 2 there is a detailed introduction to version 3 of the Business Rules Diagram, ie the last version of the BRD. I considered it important at an early stage to provide the reader with a statement of to where this research has led. Chapter 2 represents a snapshot of the diagram at the beginning of the last study in this thesis and as such is the last formal statement concerning the BRD at the time of writing. The chapter also contains some background material on for example finite state diagrams and Petri-nets to assist the uninitiated reader.

Chapter 3 describes the research question and also discusses the significance of this research. In particular it builds a case for why the research is significant by identifying those aspects of this research which are unique and discussing how this work relates to information requirements determination generally.

Chapter 4 reviews the technical literature relating to the BRD. For each main construct in the BRD, a feature analysis is conducted which involves comparing and evaluating the BRD against alternative techniques in the literature.

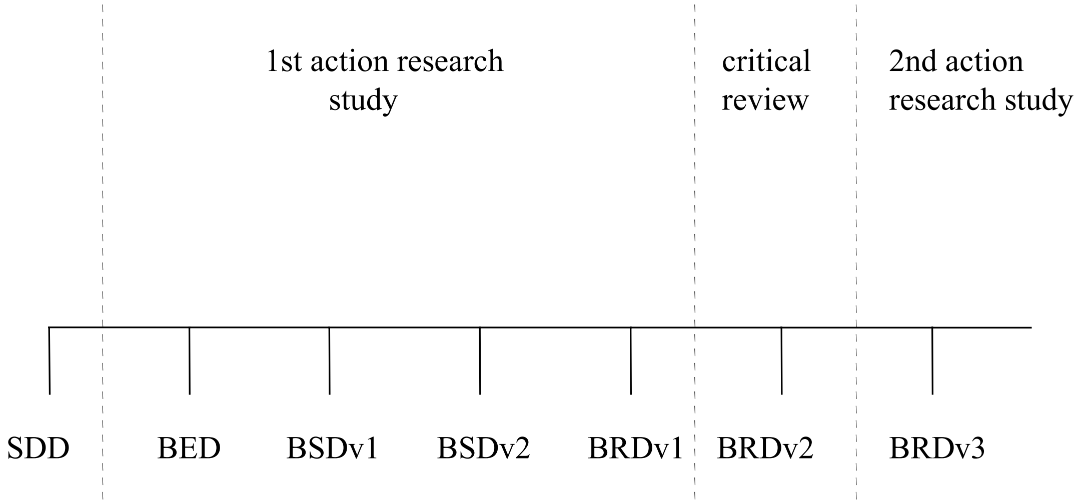
Chapter 5 deals with research methodology. In this chapter the rationale for choosing action research is justified. This involves considering positivist and interpretivist paradigms as candidate approaches for the research, constructing a short-list of candidates and then finally justifying the choice of action research.

In chapter 6 research design is discussed. Research design deals with the specific structure of the research in terms of identifying how learning was expected to take place, the actual studies undertaken and how these studies combined to form a cohesive strategy to address the research question. The chapter also outlines the data collection and analysis techniques used in each study.

The next six chapters describe the evolution of the diagram. Figure 1.1 indicates in which study each version of the diagram was developed.

In chapters 7, 8, 9 and 10 the first action research study is described in some detail. Within this study four versions of the diagram were developed in an iterative manner, ie chapter 7 describes the BED, chapter 8 the BSDv1, chapter 9 the BSDv2 and chapter 10 the BRDv1.

By the end of the first action research study a major shift in emphasis had been identified and so it was felt necessary to conduct a critical review at this point in order to regroup before proceeding with another action research study. Chapter 11 describes the detail of this critical review which took the form of reviewing the framework introduced against the current version of the BRD.



Legend:

SDD	State Dependency Diagram
BED	Business Event Diagram
BSDv1	Business State Diagram version 1
BSDv2	Business State Diagram version 2
BRDv1	Business Rules Diagram version 1
BRDv2	Business Rules Diagram version 2
BRDv3	Business Rules Diagram version 3

Figure 1.1 Map of Studies in this Research to Versions of the Diagram

In chapter 12 the second action research study is described. This study focussed on a different business application in a different organisation with a different analyst from the first study. This study also involved users.

Chapter 13 analyses the findings of this research in detail. Here a complete list of outcomes of the research are proposed and justified with appropriate argumentation. The analysis includes what the outcomes were, how these outcomes were arrived at, why it is claimed that these outcomes are valid and any qualifications or limitations concerning these conclusions. The chapter concludes with a critique of the intellectual framework used.

In chapter 14 future research directions are outlined. In this chapter a model identifying a number of areas and ways to extend the research is proposed.

The appendices contain a number of supporting materials to the main body of the thesis including detailed descriptions of each version of the diagram, transcripts of the interviews with analysts and users and other related documentation.

Chapter 2

The Business Rules Diagram (version 3)

2.1 Overview

The purpose of this chapter is to provide the reader with an overview of the Business Rules Diagram (BRD). This is done at this juncture so that the reader has an appreciation of the direction and approximate end-point of this research. The overview is a technical description of the diagram and in addition it provides some background to many of the technical ideas and concepts associated with this diagram. In this chapter such concepts are not justified; they are merely presented for the sake of providing the reader with sufficient context to appreciate the next few chapters. Later chapters will attempt to justify the rationale for adopting the specific concepts associated with the BRD.

This chapter contains a description of the Business Rules Diagram version 3 (BRDv3) which was the version used as input to the last study in this research. It should be noted that although the last study vindicated nearly all of what is introduced here as the BRDv3, some aspects were proven incorrect or require further research. At any rate, its technical details are identical to what appears in appendix 1 for the BRDv3 although some of the wording has been modified to assist its presentation at this point. Additionally this chapter contains a definition of a business rule which evolved over the course of this research and also it incorporates some background on flowcharts and Petri-nets since the BRD utilises ideas from these diagrams.

The remainder of this chapter proceeds as follows. The first section outlines the context in which the BRD is likely to be used. In doing so terms such as methodology and technique which are used as working definitions throughout this thesis are defined. The next section defines a business rule with respect to its use in this research. This includes some background discussion about business rules and a review of the extent to which other kinds of well-known model are able to capture business rules. This is followed by a section which describes the steps in the construction of the BRDv3 as well as its notation. After that there is a further section on how the model can be validated.

2.2 Context of the BRD

Working definitions of the terms **methodology** and **technique** are now introduced. An information system development (ISD) methodology can be described as a set of integrated techniques which support the development of an information system. More formally, Klein and Hirschheim (1991, p157) define a **methodology** as '*an explicit set of assumptions, beliefs and resources (tools and other means) which guide the analyst from his initial problem definition, to the final conclusion of the project.......A methodology not only guides the developer, but allows the project to move through its various stages, each delivering some form of intermediate result. Each stage is further subdivided into specific tasks with which methods and tools are associated*.' Examples of ISD methodologies include Information Engineering (Martin and Finkelstein 1981), ISAC (Lundeberg 1982), Multiview (Wood-Harper et al. 1985) and SSADM (Downs et al. 1988). Olle et al. (1991) define a **technique** as '*a part of an information systems methodology which may employ a well-defined set of concepts and a way of handling them in a step of the work*'. Examples of techniques include entity modelling, interviewing, structured walkthroughs, and black-box testing. Typically, techniques contain a relatively well-defined set of steps, which if followed, allow the developer to achieve a particular sub-goal in the development of the information system.

The BRD is considered a technique useful in information requirements determination (IRD). IRD, also known as requirements analysis, requirements elicitation and requirements engineering, is the first phase of the information systems development life cycle (Vitalari 1992). It involves taking the germ of an idea and elaborating it into a fully-fledged information system specification. Though Vitalari (1992) defines three steps of IRD (or requirements analysis), ie requirements generation, assessment and specification, a number of other authors prefer a two step definition involving requirements acquisition and requirements modelling (Valusek and Fryback 1987, Greenspan et al. 1994). It is not recommended that the BRD in itself would be used in requirements generation and assessment. Approaches such as the Soft Systems Methodology (Checkland 1981) or Cognitive Mapping (Eden 1988) would be more appropriate in establishing the nature and direction of requirements. However, once requirements are identified, there still remains the important question of how these are defined in terms of constructs in an information systems specification. Vitalari (1992) suggests that all requirements, at least initially, are actually propositions or **assertions** about some desired future information system. Such assertions, he submits, have to be 'tested' (in requirements assessment) before they would properly form part of the information system specification itself. The examination and to some extent the testing of assertions are what the construction and validation of the BRD address. The BRD highlights those constructs in an information system which depict its underlying logic or put another way, the rules by which the information system is governed. Clearly, such a model could be useful in assertion testing as described above. Further, it would permit discussion and perhaps debate about proposed changes to an existing information system and their consequences. Again, this is detailed in later sections.

So in summary, it is suggested that the BRD is a technique considered useful in IRD, particularly in the specification of information system requirements and also in the validation of assertions. It is further proposed that the technique could be formally incorporated into a methodology, but equally it could be employed as part of a contingency based approach as one of a federation of techniques.

2.3 What is a Business Rule?

There is a lack of consensus on what actually constitutes a business rule (Loosley 1992). Sandy (1994), for instance, notes that the diversity of opinion ranges from critical success factors and quality goals at the one extreme to database integrity constraints at the other. Lack of consensus presents an immediate problem not only for a researcher but for industry at large if it is accepted that business rules are central to the specification of information systems or in understanding a business.

The term 'business rule' is commonly used by analysts. Anecdotally, the following might be claimed to be typical business rules:

> customer orders may be received by telephone or post,
> part-payment for invoices is permitted,
> credit orders are granted if credit balance is greater than or equal to order value,
> only good customers may be granted credit orders and
> one customer may place many orders.

The problem with anecdotal evidence is that it may not be complete or assist in focussing on other important aspects of the problem. A theory or at least a definition of a business rule is required as a starting point for discussion. Appleton's definition (1988) for example describes a business rule as *'an explicit statement which stipulates a condition that must exist in an information environment for information extracted from that environment to be consistent with organisational policy'*. It is my view as well as others (Sandy 1994, Sandifer 1991, Appleton 1988) that the explicit modelling of business rules will do much to improve the 'state of the art' in information systems specification.

Appleton's definition emphasises the condition as an important construct which needs to be explicit in any definition of (and presumably in any diagram which attempts to model) the business rule. At this point, it might be reasonable to ask if existing models of information systems do not already capture business rules. This will be done by reviewing classes of models used in specifying information systems. These classes of model were identified from a series of conferences called the CRIS conferences. The CRIS acronym stands for Comparative Review of Information System Design Methodologies. Between 1982 and 1988, four conferences were held by the International Federation for Information Processing (IFIP). The first was a comparative review of methodologies (1982) which inspired a second conference the following year looking at the features of the methodologies in the first conference. The third conference was in 1986 and its theme was improving the practice. Lastly, the fourth conference was held in 1988 and concerned computer assistance in the life cycle. Overall, the CRIS conferences represented an important effort by a large part of the information systems research community to come to terms with methodological issues by comparing and contrasting approaches from academic as well as practitioner groups. Much of the work of these conferences was distilled into a book (Olle et al. 1991). From the point of view of this thesis, it is interesting to note that Olle et al. detected a trend in the use of behaviour (dynamic) models. Olle et al. (1991) identify three kinds of model which pervade information systems methodologies. These are data models, process models and behaviour models.

2.3.1 Data Models

The entity relationship model (Chen 1976) is an example of a data model. The purpose of a data model is to store data over time in a fashion which is accessible at some future point in time. It is a static model in that no explicit movement or change is defined. The data model can be used to infer some types of business rule. For instance, the existence of part-payments can be detected from a data model, thus implying that part-payment is at least some aspect of business policy. Also, where a ‘many’ relationship exists between say an order and invoice, the reader can conclude that many invoices may be generated for one order. Optionality and mandatoriness in relationships (Rumbaugh et al. 1993) further allow for the specification of aspects of business rule semantics. For example, if one order can have many invoices and the ‘many’ side is optional, this means that the order can exist independently of the invoice thus possibly implying that the invoice is generated after the order. Thus a static model can be used to make temporal inferences. However, although some business rules or aspects of business rules can be modelled using ‘static’ models such as the data model, other business rules cannot, such as anecdotal business rules three and four above.

2.3.2 Process Models

Process models such as the dataflow diagram (De Marco 1978, McDermid 1990b) can be argued to be dynamic in the sense that they show movement of data through an information system. However, the question here is to what extent can they model business rules. Again the answer seems to be only to a limited extent. Certainly, showing telephone and postal orders of customers (rule 1) could be detected from a dataflow diagram and by inspecting its details it could probably be deduced that part-payment for invoices is permitted and that one order generates many invoices. The word probably is used in the previous sentence because the reader would have to be asking the right questions in order to find the answers from the diagram. In other words, dataflow diagrams emphasise the flow of data in an information system and the processes in an information system. They do not necessarily lend themselves to, nor are they arguably designed for, highlighting the rules of an information system. Rules three and four above for example would be embedded in process specifications which are a complement to dataflow diagrams in many methodologies (Gane and Sarson 1979). Further, even if the part-payment rule could be deduced it would not be clear under what circumstances part-payment was permitted. For example, is it given to all customers, is there a minimum part-payment?

Another problem with dataflow diagrams is that they model the information system rather than business policy. In other words, it is likely to be difficult if not impossible by inspection to separate information system rules from business policy rules, ie those more specific rules which reflect the implementation of business policy rather than what is inherently part of the business. This point will be elaborated later.

2.3.3 Behaviour Models

Olle et al.'s third category of model was the behaviour model. Now the term behaviour can mean many things. However, the underlying theme behind the term seems to concern time and the modelling of time-dependent behaviour. For example they state that *"The most recent trends in information systems methodologies focus on the dynamic nature of the data and the need to analyse and understand events in the real world which impact data recorded in the information system ... this view concentrates on changes over time which may, and which are perceived to, take place ..."* (Olle et al. 1991, p54).

Many related types of computerised system such as real-time systems, command and control systems rely heavily on the passing and counting of time as a means of exercising control over themselves. Indeed the authors of many information systems books have borrowed embedded systems examples such as microwave ovens or washing machine systems to illustrate certain kinds of behaviour models (Shlaer and Mellor 1992). The use of such borrowed examples suggests that the information systems community is lagging behind its sister disciplines (such as real-time computer systems) in terms of its ability to concisely express requirements. In these sister disciplines, it is common to see Petri-nets (Peterson 1977, 1981) and finite state diagrams (Davis 1988) used to show behaviour in these systems. See Rosenquist (1982) for an example of finite state diagrams used in an information systems context. Petri-nets depict states and transitions (events) and the dependencies between them whereas finite state diagrams are a restriction of Petri-nets which only show states and their interdependencies.

However, it is debatable whether the states and events of the Petri-net on their own could completely model all the rules in the list of anecdotal business rules. Using only states and events only rules three and four could be straightforwardly modelled. While the finite state diagram and Petri-net are only two examples of behaviour models, it is argued that the emphasis on time and precedence makes it difficult for behaviour models generally to easily capture more 'static' rules such as rules one, two and five.

The basis of the model developed in this thesis is a behaviour model. It may be described as a set of extensions to a Petri-net, the extensions of which are designed to capture other types of business rule. However, in one important respect, the BRD contains a major restriction in terms of what others might regard as business rules. This is now discussed.

I have concluded that there is a need to distinguish levels of 'business rule'. It is proposed that is not necessary for an organisation to group all of its rules together as business rules. Instead, it is submitted that (at least) three types of rule in a business can be identified and that each might be better kept in separate repositories. Furthermore, it will be argued that the topmost level of business rule can be captured as a true conceptual model of the business in the sense that it can show rules which are unconstrained by how people in the organisation execute these rules or indeed how a computerised information system might implement these rules. Figure 2.1 depicts three levels of business rules.

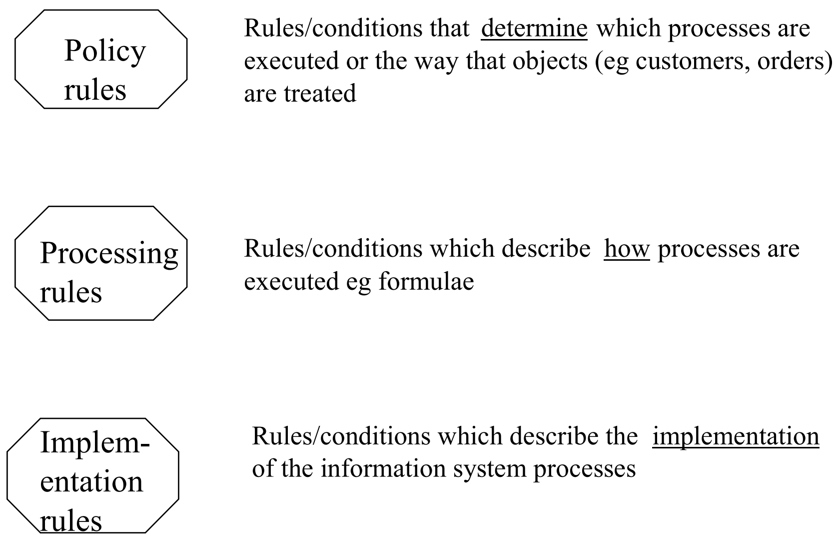


Figure 2.1 Levels of Abstraction of Business rules

At the highest level such business rules may be termed policy rules in the sense that they reflect the essence or core of what the business is about. For example, a bank would want to distinguish overdraft customers from customers with a positive balance. They will typically be treated differently and different processes and procedures would be likely to apply. Rules at this level would be considered fundamental to the business. In other words it is difficult to conceive of describing human activity systems without such policy rules. Also, observe that such rules would have to contain the pre-conditions which fire processes or activities in the business eg overdraft customers are given higher interest rates, reminder letters etc.

At the next level, what may be termed processing rules can be seen to exist. These rules arguably are also business rules. However, in comparison to policy rules they relate more to the detail of how the processes and procedures are actually carried out, rather than the condition(s) under which the business may choose to follow different processes. So, for example, a rule expressing the calculation of a sales tax or the detailed sequence of steps in accepting an order would be classified as a processing rule. Policy rules can be distinguished from processing rules in the following way. There is a qualification to this which will be described later but in essence a rule is a policy rule if any external observer (ie someone outside the human activity system but associated with it) can detect that the system has changed state. In other words, policy rules determine state changes (ie the external WHAT) whereas processing rules execute state changes (ie the internal HOW).

At the lowest level of abstraction in this figure sits the implementation rule. Here the distinction lies in that the rule relates to how the process is actually implemented in the **information system** eg the batch printing of invoices for efficiency's sake would be an implementation rule rather than a processing rule. Also, the rules of database integrity (such as that one entity occurrence must be deleted before another entity occurrence) would be considered implementation rules, though arguably a lower level yet could be identified for this.

Failing to distinguish these levels has arguably led to confusion and overloading. Confusion has arisen because various authors are working at different levels of abstraction. For instance in the practitioner literature some examples of business rules are provided which seem more concerned with describing a level of abstraction equivalent to implementation rules as in 'employee ID is a numeric field between 1,000 and 3,000' (Jones 1991, p9). The overloading problem is likely to occur because without these levels of abstraction any diagrammatic presentation of business rules would inevitably contain more than one type of rule if not all and therefore this would lead to a very busy diagram.

Throughout this thesis, the term **business rule** will refer only to those high level policy rules in a human activity system. Such a model would be a conceptual model of business policy and as far as possible therefore be unconstrained and independent of processing activity and information system considerations. The question of how to identify and express processing and implementation rules is not addressed in this thesis, although it is suggested here that such identification would naturally occur after the BRD has been drawn and agreed by (future) users of the information system.

I have concluded that a business rule requires four explicit components or constructs. These are states, events, conditions and signals. **States** reflect the status of an object of interest at any given time, so for example a manufacturing work order might occupy the states unstarted, in progress or completed. **Events** are actions carried out internally by the organisation. They are considered to be instantaneous occurrences which reflect the organisation's policy on what should happen in a particular circumstance eg cancel work order. One important role of the event is to avoid processing rules. Rather, such processing detail is best kept separate from 'policy' rules as discussed earlier. **Conditions** define the criteria by which objects of interest in the business move from one state to the next as events take place. Sometimes many conditions require to be met in order for an event to take place thus increasing complexity. It is argued that modelling conditions without the context of states and events (and vice versa) is far less powerful. Lastly, **signals** either enter or leave the human activity system. Signals which enter the system will typically initiate activity within the system and so these are called **triggers**. Triggers may be external such as a customer sending an order or internal such as one department sending a document to another department which then triggers off some activity. Further, a trigger may be a time trigger eg an activity beginning at the start of the day or the end of the month. Those signals which leave the system serve the purpose of informing those outside the system of what has occurred inside the system and therefore are referred to as **messages**. Thus, though some might argue that the idea of a condition is at the heart of a business rule (as in Loosley's definition (1988)), the related constructs of state, event and signal provide a context for the business rule. So, as an aid to memory we might say:

Business Rules are made up of States and Events and Conditions and Signals

The following is my definition of a business rule, though at this stage the reasoning for some of the specific wording may not be clear. *A business rule is an explicit state change context in an organisation which describes the states, conditions and signals associated with events that either change the state of a human activity system so that subsequently it will respond differently to external stimuli or reinforce the constraints which govern a human activity system.* It is claimed that the components of a business rule in the above definition permit the range of anecdotal business rules identified earlier in this section to be specified. Moreover, the definition has been so constructed as to convey other aspects of a business rule such as making explicit its constructs. The term human activity system in the definition stipulates that a business rule is an organisational artefact rather than an information system's artefact and thus is consistent with modelling only policy rules. The terms state, event, condition and signal are defined in the previous paragraph.

2.4 Theoretical Bases of the Business Rules Diagram

In the construction of the BRD use is made of concepts borrowed from flowcharts, finite state diagrams and Petri-nets. The section provides a brief introduction to these diagrams.

2.4.1 Flowcharts

Flowcharts have been in use for a long time. Within the information systems community flowcharts were in use from the earliest days. For example they were used to chart the logic of clerical tasks and computer programs in the 1960s. Flowcharts have been used for many purposes, for example as computer runcharts tracing the sequence of programs and files in a system and as a system flowchart identifying high-level processes in an information system (Daniels and Yeates 1971).

Over the years different sets of symbols have been experimented with. For example the ECMA (European Computer Manufacturer Association) standard has seven symbols including symbols which represent filing, manual operations, machine operations, testing and so on (Davis 1983). Perhaps one of the best known standards is the NCC standard for computer procedures which has only four symbols (Daniels and Yeates 1971). These are a rectangle for an operation, a diamond for a decision, a small circle for a connector and a softbox to show entry to or exit from a program. Variants of this basic standard also exist. For example Davis (1983) uses a rhomboid to depict input and output operations. Whichever set of symbols is used, they are combined together to define the control logic of computer procedures.

Three control constructs are used to describe logic within a flowchart. These are sequence, selection and iteration. Sequence involves describing operations serially thus defining the order in which they occur. Selection involves a binary condition. Depending on whether the outcome of the condition is true or false determines which path in the flowchart is then followed. Iteration involves looping back to an earlier decision point. Typically this looping back is continued until a criterion is met. Figure 2.2 is an example of a flowchart which contains all three constructs.

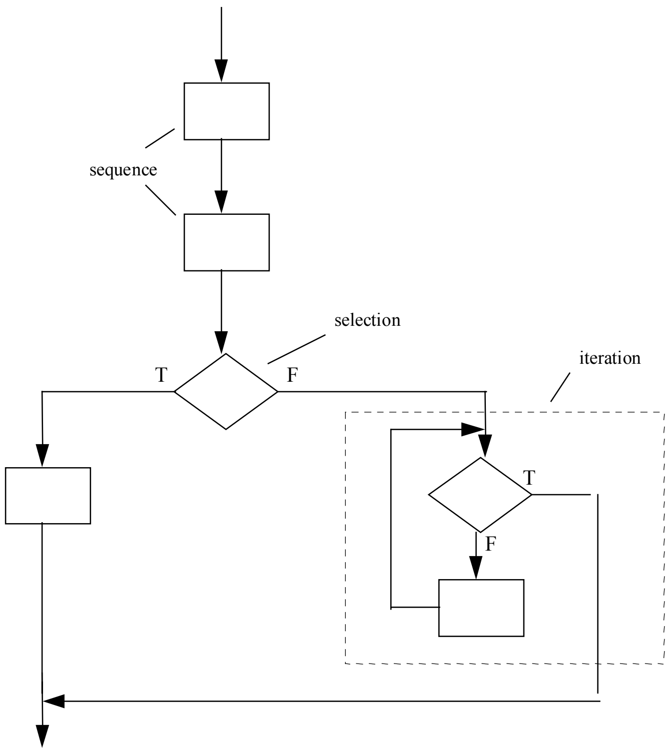


Figure 2.2 Example of a Flowchart

The BRD uses flowcharting concepts in two ways. Firstly, the BRD uses the diamond shape as a condition and also the rectangle, which is an operation on the flowchart, is used as an event. Further, the BRD has a connector symbol, though it happens to be a small square rather than a small circle. In the flowchart a softbox is used to denote entry to or exit from a computer program. In the BRD entry to or exit from a human activity system is called a signal but it is denoted by a thick arrow. Overall it can be seen that the BRD borrows heavily from the computer procedure flowchart in terms of what the basic symbols represent.

The second way in which the BRD uses flowcharting ideas is evident in what will later be defined as the User Business Rules Diagram (UBRD). Here the constructs of sequence, selection but not iteration are used. In other words, users are expected to work with a flowchart-like diagram which represents a business rule. Because of the level of abstraction consistent with the initial modelling of policy rules, it is claimed that any policy rule can be described by using a combination of sequence and selection constructs. In other words it is asserted that showing iteration is not helpful at this level of abstraction. Figure 2.9 is an example of a UBRD.

I have incorporated the condition symbol into the UBRD in a flowchart-like manner because I believe it is superior to alternative non-graphical approaches especially for use with non-computer people. In this context non-graphical approaches essentially imply the use of English, structured English (Gane and Sarson 1979, McDermid 1990b) or pseudo-mathematical notations (eg Kung and Solvberg 1986). Such alternatives may be considered linear notations. Fitter and Green (1979) provide some insight into the arguments surrounding the use of a graphical device over linear notations such as pseudo-mathematical notations or structured English. For example, they concluded that as far as analysts were concerned the flowchart was better for spotting mistakes than linear notation. Their position is summarised in '*Our conclusion is that if a graphical notation can reveal the structure inherent in the underlying data or the process by which entities are manipulated, then it will be superior to a linear symbolic language*' (Fitter and Green 1979, p255). They maintain that the flowchart has a process quality to it involving a 'test-action' pair and are positive about the ability of most beginners to take advantage of it. Further Bohl (1971, p53) notes '*The flowchart is an essential tool in problem-solving...The person who cannot flowchart cannot anticipate a problem, analyse the problem, plan the solution, or solve the problem*'. However, not all studies reveal such a positive outlook. For example, Schneiderman et al. (1977) found that flowcharts gave no advantage in modifying programs etc., but here the group analysed were programmers performing a maintenance activity rather than non-computer people specifying information requirements.

2.4.2 Finite State Diagrams and Petri-nets

Finite state diagrams and Petri-nets are reviewed more formally in the literature review. Here the concepts of these diagrams are introduced informally in a tutorial fashion so that the uninitiated reader may follow the remaining technical description in this chapter.

Suppose an organisation wishes to model aspects concerning its customers. For the sake of this example further imagine that the organisation identifies three kinds of customer. These are **temp** customers to whom no credit will be extended during a probationary period, **good** customers to whom normal credit facilities are extended and **bad** customers who have at one time been good customers but who defaulted on their repayments. Customers may move from one type to another as follows. Initially all customers are temp customers. After a successful probationary period they are moved to become good customers. If at some future stage they default on repayments they may be moved to bad customer status. Presumably as bad customers they are treated differently, eg they may be given no further credit. If at some future point they are deemed worthy of good customer status, for example by paying off their outstanding debt, they may be returned to good status from bad status. This movement from good to bad status and vice versa may occur many times with customers. Further, customers may never return to temp status in this example.

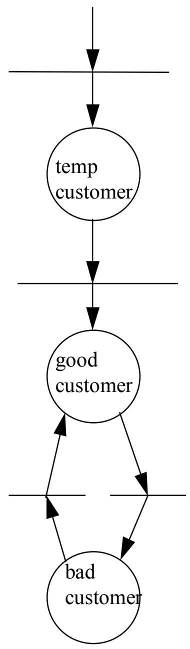
It is possible to model this scenario quite succinctly in a graphical fashion. Figure 2.3 is a finite state diagram which models the above scenario. The circles are considered states of the customer. Since there are only three states this is certainly a finite number. The arcs in the diagram depict what is permitted in terms of movement from one state to the other.

Legend:
circle: represents state of customer
arrow: represents precedence

Figure 2.3 Finite State Diagram for Customer Scenario

One limitation of the finite state diagram is that it only models the states of an object. For example it does not explicitly model the events which cause a state to move from one object to another neither does it model the criteria which determine the transition between one state and another. Figure 2.4 is a Petri-net for the same customer scenario. Unlike finite state diagrams, two constructs are modelled graphically. These are the state and the event. (In strict Petri-net parlance these are known as placeholders and transitions respectively.)

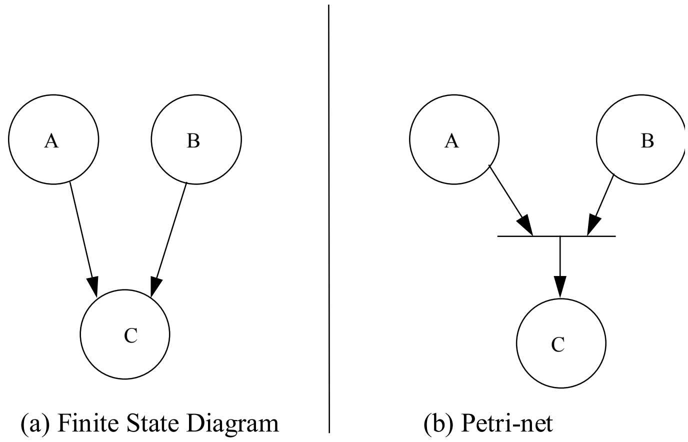
The Petri-net is superior to the finite state diagram in that a more complete picture of a scenario may be modelled, ie the events associated with transitions can be modelled. This can be helpful in modelling more complex situations because a limitation of the finite state diagram is that in any transition between one state and another there may only be one input and one output (Peterson 1977). For example suppose that both states A and B are prerequisite to state C, ie both must pre-exist for



Legend:
circle: represents state of customer
bar: represents event (transition)
arrow: represents precedence

Figure 2.4 Petri-net for Customer Scenario

state C to come into existence. Figure 2.5a shows an incorrect attempt at drawing the finite state diagram for this. Based on the commonly accepted rules used with this type of diagram, a reader looking at figure 2.5a would infer that either state A or B but not both in any one transition are prerequisite to C. On the other hand figure 2.5b which is the Petri-net for the same problem is much clearer since the intervention of the event symbol and the ability to show two inputs to it provides a more accurate picture of what is going on although of course the logic of the transition itself still needs to be defined. While it is possible to incorporate graphical tactics in a finite state diagram (as I attempted at one stage in this thesis) in an effort to ameliorate this problem, in general it is argued that Petri-nets provide more expressive power in such circumstances.



Legend:
circle: represents state
bar: represents event (transition)
arrow: represents precedence

Figure 2.5 Comparison between Finite State Diagram and Petri-net

Further it is possible to annotate Petri-nets so that for example the rules or conditions for events are made explicit. Such annotation may appear on the diagram itself or alternatively be specified separately with a cross-reference to the diagram. Such annotation increases the expressive power of the Petri-net, though it is argued that the use of explicit graphical constructs for showing conditions and signals as I have done, emphasises other important aspects of behaviour.

Both finite state diagrams and Petri-nets are utilised in the BRD. Finite state diagrams are used to model object life histories which will be discussed shortly. Perhaps more fundamentally, the BRD contains the constructs of state and event and as such the BRD can be considered to be an extension of the Petri-net. These extensions include particularly the graphical constructs of conditions and signals as well as other extensions.

2.5 Constructing the Business Rules Diagram

The steps involved in creating the Business Rules Diagram are:

- identify candidate business rules (policy rules),
- identify candidate business events and signals,
- identify candidate objects in problem situation,
- construct object life history for each candidate object identified,
- construct User Business Rules Diagrams and
- construct Business Rules Diagram.

These steps are now described in some detail.

2.5.1 Identify Candidate Business Rules (Policy Rules)

This is achieved by assembling groups of users together and brainstorming a list of candidate rules. After brainstorming, the list is reviewed and rules which are obviously not policy rules are removed. Table 2.1 contains a set of policy rules for an order processing case which will be used throughout this thesis (appendix 7). Though every reasonable attempt is made to identify all policy rules, later steps provide an opportunity to identify rules which have been overlooked here.

Orders sent by mail or telephone
Omission on order line leads to deletion of that order line
Credit balance >= order value to accept order, otherwise reject
Stock qty >= order qty for normal order, otherwise outstanding
One invoice for one order
Sum of payments = order value - sum of credit notes
One order may have many credit notes
Many payments per invoice possible
Overdue invoices occur 30 days after Statement
If product line not carried, reject item
If unobtainable multiples reject item
New order created for outstanding items
Only good customers may obtain credit orders
Credit balance reduced for all items on an order including outstanding items

Table 2.1 Candidate Business Rules for Sample Case

2.5.2 Identify Candidate Business Events and Signals

A first-cut list of events and signals is also achieved through a brainstorming/review process. At this stage, the distinction between an event and a signal may not be so clear in the minds of users. Both events and signals are considered instantaneous and further the exact boundary of the information system may still have to be clarified in detail. In the first instance it is considered more important to identify through the brainstorming process as complete a list as possible and then that list can be classified into events, triggers or messages. Triggers, such as the receipt of an order from a customer are coded with a ‘T’ for trigger. Events such as the creation of an outstanding order item (or back order) due to insufficient stock are coded with a ‘E’ and messages such as the sending of an invoice are coded with an ‘M’. Of course the act of coding may throw up omissions in the brainstorming process. For example, the sending of an invoice (ie a message) is a different activity from the creation of the invoice (which is an event) and thus the list may be added to as omissions are identified. Table 2.2 contains a list of candidate business events and signals for the same order processing case.

Receive customer order	T
Delete line	E
Reject order	E
Create new order	E
Send invoice	M
Generate credit note	E
Receive payment	T
Create outstanding item	E
Create new customer	E
Move to good customer	E
Move to bad customer	E

Table 2.2 Candidate List of Business Events and Signals.

Each event and signal at this stage is only a candidate. A business event should be a significant occurrence in the life of an information system (ie externally verifiable). The event therefore should have impact on how some component in the system is subsequently dealt with. Again, it is not vital that every event or signal is identified here, as subsequent steps may throw up more events. This step could be combined with the first step into a single session in which two lists are generated.

2.5.3 Identify Candidate Objects in Problem Situation

Object modelling is a way of partitioning a system into components (Coad and Yourdon 1991). Such partitioning is performed to enable individuals working with the system to deal with properties and aspects of the system at a more local and focussed level (ie at the object level) as opposed to working with the whole system. For example the object customer may be perceived to be a component of an order processing system. The ability to focus on aspects of a customer without reference to an information system which involves customers makes it easier for users to explicate requirements. Working with objects is considered by many to be natural and intuitive to users (Jacobson et al. 1992).

Objects may be simple or complex. A simple object is one in which each property is single-valued; objects whose properties are multi-valued or whose properties are themselves objects are considered complex (Atkinson et al. 1989). In modelling a business in terms of objects, there is a need to model complex objects (as is illustrated shortly) to allow working at the level that users regard as appropriate. See Coad and Yourdon (1991) for an example of a methodology which employs only simple objects.

The application of object-oriented ideas to the domain of information systems is a relatively recent one. For example, Jacobson et al. (1992) in tracing back the roots of object-orientation to the telecommunications industry comment that it is only now (ie in the nineties) being used in information systems, real-time systems, CASE tools and so on. Further, it is my observation that many authors of books and articles borrow examples from sister disciplines of information systems rather than using examples or case studies from the domain of information systems itself. For instance Coad and Yourdon (1991) use the example of a microwave oven to introduce a finite state diagram. However, this merely demonstrates that applying object-orientation to information systems, especially in requirements analysis, is a relatively new phenomenon. Since the beginning of the nineties a number of well-respected books have been published with the information systems community in mind. These include Rumbaugh et al. (1991), Jacobson et al. (1992), Coleman et al. (1994), Henderson-Sellers and Edwards (1995), and Martin and Odell (1995). More recently the UML (Unified Modeling Language) has been made available (UML 1997). The UML is not a methodology as such but rather a federation of mostly graphical techniques for abstracting a variety of aspects of an information system. The BRD could be incorporated as an individual graphical technique into the UML.

Jacobson et al. (1992) refer to the difference between the real world and a model as a semantic gap and claim that object-oriented models make for small semantic gaps. The advantage of this correspondence between real world and model is that users and analyst(s) then have a common framework in the model to clarify issues concerned with for example completeness and correctness. The analyst thus **learns about the human activity system** through a dialogue with users about the model.

A related issue here is to do with abstraction. Abstraction refers to the principle of emphasising some aspects of a system which are considered relevant to the exclusion of others (Jacobson et al. 1992). In terms of the BRD, the constructs of state, event, condition and signal are considered to be the important constructs as far as business rules are concerned. Further, the focus on policy rules to the exclusion of others is also an application of the abstraction principle. The principle of information hiding is also used in object modelling. Information hiding refers to the principle of hiding certain detail about the system so as not to obscure the main informational value of the model (Parnas 1972). In an important sense this is the dual of the previous characteristic about abstracting certain features.

By scanning the lists of events, signals and rules, a candidate list of objects can be identified. From the two tables above the candidate objects of **order**, **stock** and **customer** can be identified. Less clear is the existence of another object called **outstanding order item.** Assume that the way this case operates is that outstanding order items exist over some time and during that time quite a number may accumulate from across several orders for the same customer. Such a situation would justify identifying a separate object and importantly also lead to rewriting the business rules list and updating it with a new rule, ie that an order can be created from outstanding items from several different initial orders. Note also that in turn, this ought to lead to updating the events list so that outstanding items are at some point converted to normal order lines. At any rate, for the purpose of illustration here, there are four objects, ie order, stock, customer and outstanding order item.

2.5.4 Construct Object Life Histories for each Object Identified

Figure 2.6 contains the object life histories (OLHs) for the four candidate objects. The practice of modelling object or entity life histories is relatively well established in the information systems community (Downs et al. 1988, McDermid 1990b, Coad and Yourdon 1991, Shlaer and Mellor 1992) though it is by no means ubiquitous. However, this practice usually relates to constructing OLHs from simple objects rather than complex objects. Entities by definition are simple.

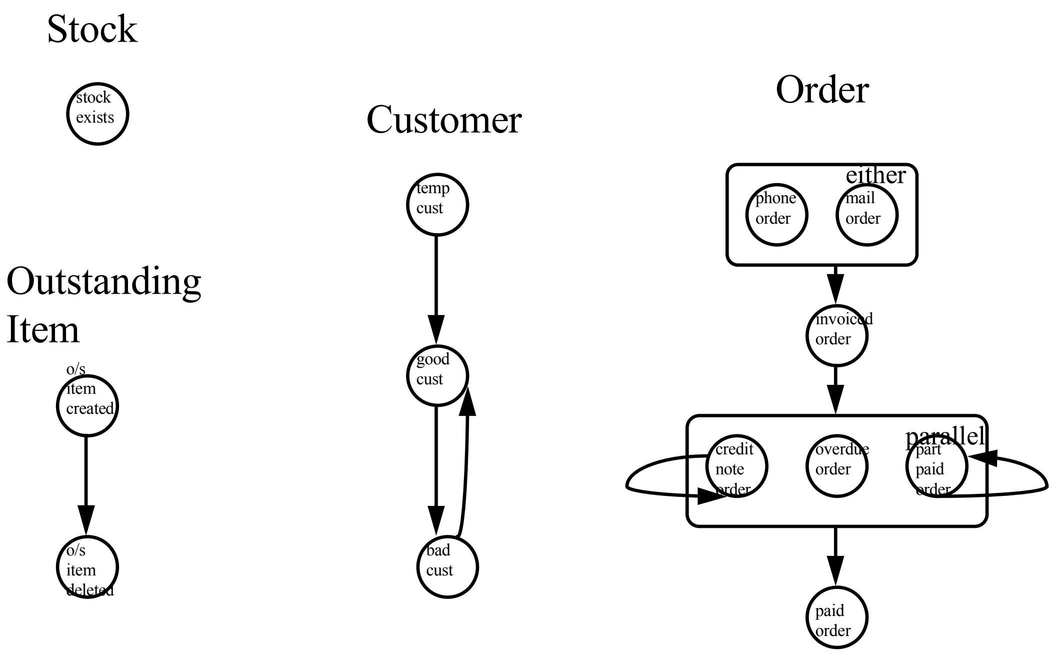


Figure 2.6 OLHs for Candidate Objects

An object life history is modelled by identifying the sequence of states that an object occupies over time. Ackoff (1971) defines the state of a system at a moment in time as '*the set of relevant properties which that system has at that time. Any system has an unlimited number of properties. Only some of these are relevant to any particular research.... The value of the relevant properties constitute the state of the system*'. Thus the state of a system (and therefore the state of an object which is some component of a system) is an abstraction of the system to meet a particular end. For the BRD, candidate states are those which are occupied for some period of time and which are **perceived as relevant for describing the business rules of the system**. While the OLH can show the sequence or route that any one instance of an object may take, it also is a template indicating all possible routes of all instances of an object (class).

States in the OLH are shown as circles and sequence (precedence) is shown by a single-headed arrow. Note the existence of the deleted state for outstanding order item. On first inspection this may seem strange in the sense that non-existence is being modelled rather than some aspect of existence. Also it might be questioned why this aspect is modelled for one object but not the others. The problem is that outstanding items are to be ‘converted’ to normal order lines at some point. While the logic for this has not yet been identified, identifying a deleted state is one way of flagging that this has to be dealt with later.

Ideas from the work of David Harel and his higraphs are used in the BRD. Variously referred to as Statechart (Harel 1987), Higraph (Harel 1988) and Objectchart (Coleman et al. 1992) his diagrams incorporate a powerful device known as Harel depth for reducing the visual complexity of certain diagrams. The basic principle involves using the concept of depth to delimit the scope of effect of an arc. In figure 2.7, an object order is shown which has two states - a telephone order state and a mail order state. Arc b represents a direct connection to the mail order state and arc d a connection away from the mail order state. Arcs b and d therefore allow the logic relating to a specific state of an object to be depicted. On the other hand arcs a and c respectively show connections to and from the whole object. In other words, these arcs apply equally to all states within the object, in this case the telephone order state and the mail order state. Though simple and intuitive this is a powerful mechanism for reducing the number of arcs necessary on a graph and thus makes the graph more readable.

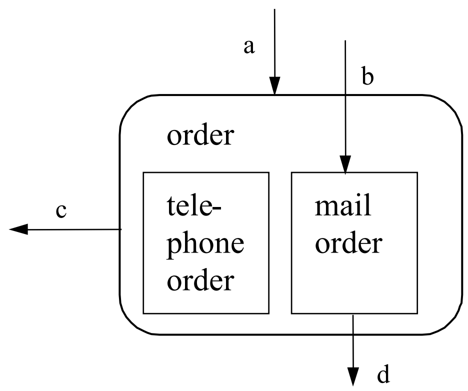


Figure 2.7 Illustration of Harel Depth

There are two ways in which Harel depth (sometimes called the Harel blob) is used in the OLH. The first way is essentially that described above, ie where it is desired to indicate selection. A softbox is drawn around, in this case, the two states of telephone order and mailed order and the label ‘either’ inserted. See figure 2.6. Arrows stop and start at the edge of the softbox indicating that either (but not both) state may exist for any one instance of an order and that any predecessor and successor states apply to both. The second way is where there is a need to indicate parallel states. Take the order OLH for instance. Here, credit-note order (which means that at least one credit-note has been generated for that order instance), part-paid order and overdue order may all be created provided that the instance is in invoiced order state. However, an important difference here is that these three states may co-exist, ie it is possible that credit-notes may have been generated while at the same time part-payments have been received and the order becomes overdue. The 'parallel' label indicates the propensity for parallel existence. See figure 2.6.

2.5.5 Construct User Business Rules Diagrams

The analyst scans the OLHs to identify use cases. Jacobson et al. (1992) use the term 'use case' to describe an episode of use of an information system. A use case represents one session of interaction with the system; such a session is a meaningful, whole task in its own right as far as a user of the system is concerned. For example, the episode of receiving an order and processing it to its acceptance is a use case. One user usually will perform one use case (though one use case type may be performed by many users). There are many possible outcomes to a use case. For instance, in processing an order the possible outcomes include outright acceptance, outright rejection, deletion of some ordered lines but acceptance of the rest and so on. For the most part, one state change (or creation) will equate to one use case though not always. For each use case identified, the analyst then assembles the user or users who will have responsibility for that use case. A User Business Rules Diagram (UBRD) is drawn for each use case.

The User Business Rules Diagram is constructed by elaborating the states in the OLHs diagram by adding conditions (ie the candidate business rules), events, triggers and messages from tables such as tables 2.1 and 2.2. The notational constructs used are defined in figure 2.8. Figures 2.9, 2.10, 2.11, 2.12 and 2.13 are the use cases (UBRDs) for the order processing case study described in appendix 7.

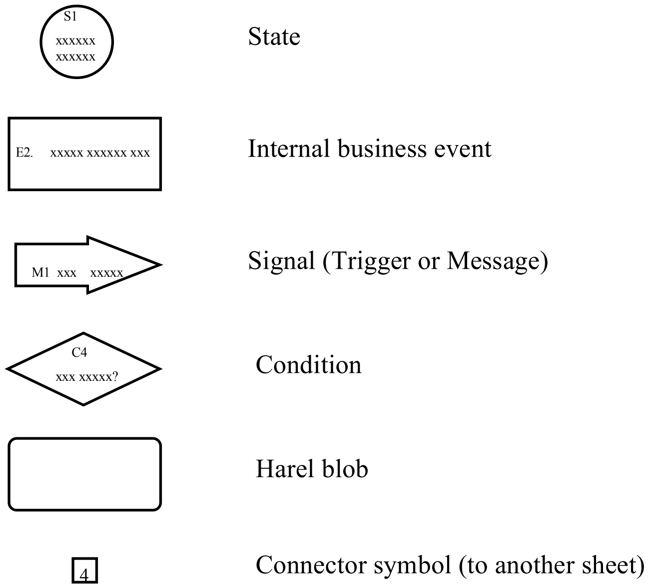


Figure 2.8 Notational Constructs for Business Rules Diagram

The purpose of drawing the UBRD is for the analyst to glean as much information as possible about the business rules concerning the use case. It is envisaged that the analyst will create the UBRD with the users present and that therefore it is important that the notational complexity of the UBRD is kept to a minimum.

Note the 'flowchart look and feel' of the UBRD in figure 2.9. This keeps the diagram intuitive for the users' benefit yet at the same time allows the analyst to build a comprehensive picture of the rules behind a use case. Notice that some outcomes such as the creation of the outstanding item state do not involve the sending of a message to the customer. The opposite situation also arises in figure 2.9 wherein a deleted line message is sent to the customer but no state change occurs. On yet other occasions such as the creation of a credit note in figure 2.12, both a new state is created and a message to the customer is sent. Overall this permits the range of business policy situations to be described in an efficient way.

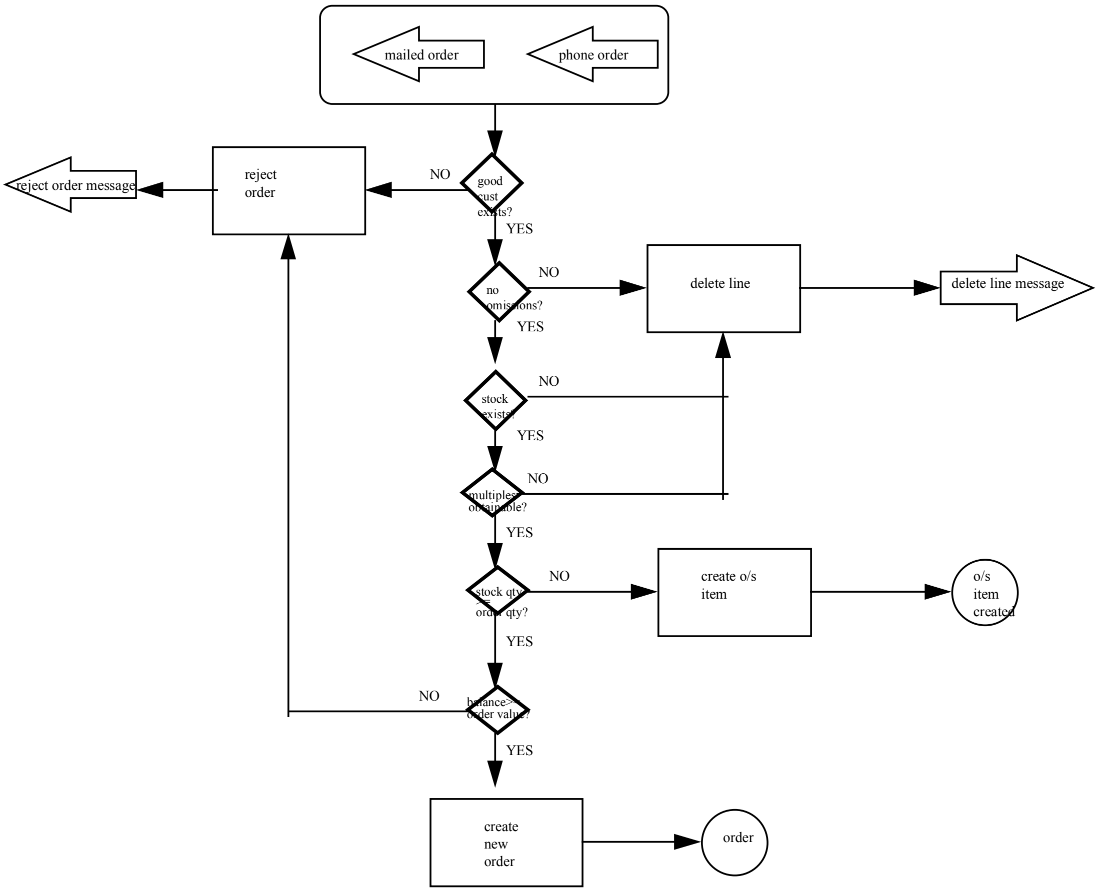
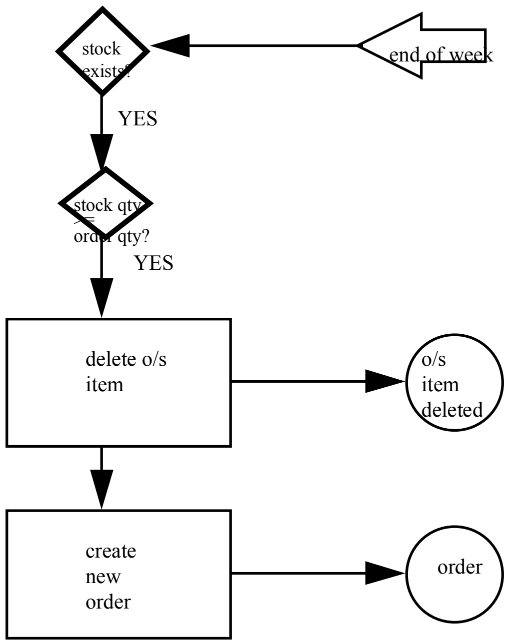


Figure 2.9 Accept Order UBRD

Sometimes a situation arises in which nothing is done in a certain circumstance. For instance, in figure 2.10 when scanning stock levels at the end of the week, if the ordered quantity is greater than the stock level nothing happens. Again, this allows current business policy to be depicted in an efficient way, but the analyst should always take care to confirm such phenomena. This illustrates the reasoning for including the phrase *'reinforce the constraints which govern a human activity system'* in the definition of the business rule because some executions of a business rule will not result in a state change. Note also the existence of a time trigger in figure 2.10.

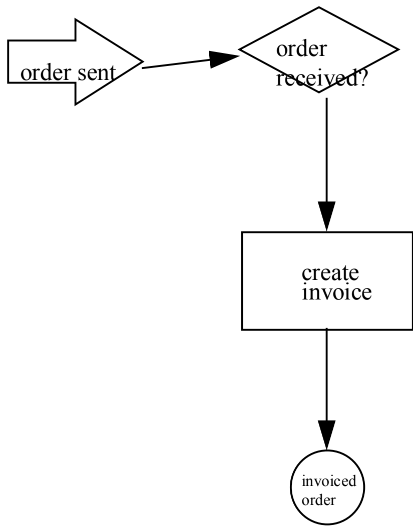


(In this UBRD, if stock is sufficient, order-lines are deleted from the outstanding item object and are used to create a new order object instance).

Figure 2.10 Create New Order from Outstanding Order UBRD

Figure 2.11 represents a very straightforward situation in which the event occurs after the order is sent to that department. In this example a different clerk is responsible for creating the invoice and this is done only once the order has been accepted. In a manual system, there will be a delay before this happens. Clearly, such situations are candidates for automation. However, the analyst should confirm that such circumstances are indeed automatic and that no exceptions are possible.

Figure 2.12 shows a common situation. Here an external trigger, in this case a complaint is presented to the system. The system will either accept or reject the complaint. If the complaint is accepted, there is a state change; if rejected there is no state change but a message is sent back to the customer. One part of this particular business rule is that 'if a complaint is accepted then a credit note is generated'. The other part of this policy is that 'if a complaint is rejected a customer will receive notification to that effect'. So, as defined earlier a business rule is not just one condition or one route but rather a whole context in which many possible state changes, conditions and routes through the business rule may occur. Thus a rich picture of the business rules in an organisation is obtained. The context includes the conditions, events, triggers, messages as well as state changes.



(In this UBRD, when an order is received a single invoice (invoiced order) is created from it).

Figure 2.11 Invoice UBRD

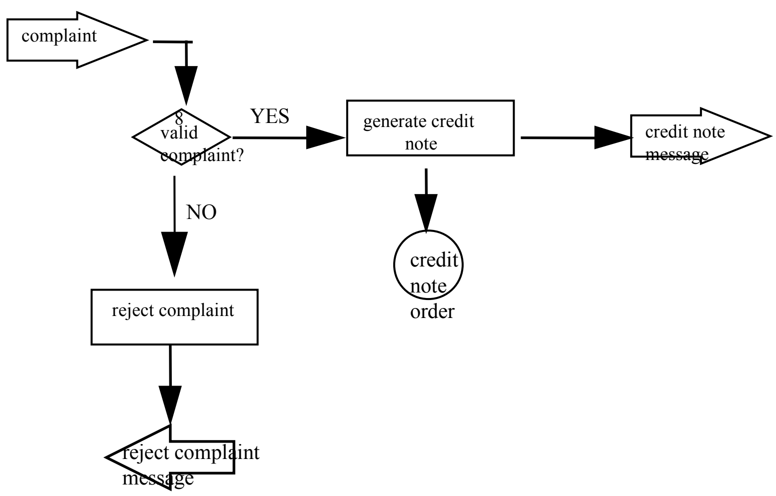


Figure 2.12 Credit Note UBRD

Figure 2.13 shows a structurally similar situation. However, here notice that there is no rejection situation as such unlike that in figure 2.13. Is the diagram implying that a payment can never be rejected? Again, the construction of the UBRD with users can elucidate such contingencies if they were overlooked. Having said that, there exists a lower level of business rule (ie that of the processing rules) at which basic validation criteria for the acceptance of any document might be located. This suggestion is made even though in figure 2.9, criteria for the acceptance of an order are specified including checking for the omission of certain fields on the order. This is because in figure 2.9 checking for omissions is part of business policy whereas in figure 2.13, the rejection of a payment would be a processing rule problem. In other words, figure 2.13 reflects a situation where an organisation has limited business rules for checking omissions or errors in a payment. Of course whether it ought to have a richer set is another matter.

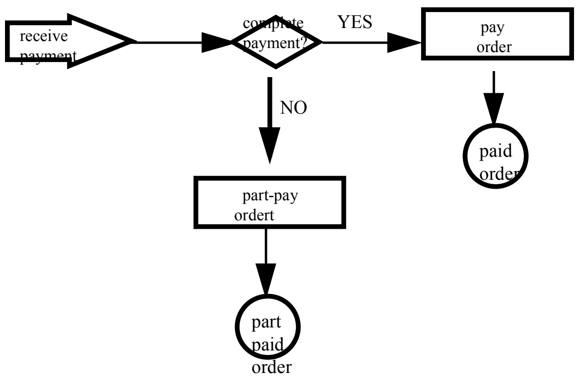


Figure 2.13 Payment UBRD

2.5.6 Construct the Business Rules Diagram

In this section the abbreviation BRD will be used to refer to the diagram rather than the whole technique. The BRD is constructed by amalgamating the UBRDs into a single diagram. The construction of the BRD requires dedication to sorting out the inherent logic of an information system and to describing the logic in a particular way. As such it is one of the most crucial steps in the process and it will probably require several iterations to get it right. For example, sometimes the collective knowledge about a business situation is not complete. This is illustrated in the left-hand section of figure 2.14 by the gaps between the states of the object customer. This highlights that there is insufficient knowledge about the conditions, triggers, messages and events between these states. As such it shows how the diagram can be used to detect omissions of business rules, events or signals from the original lists.

There are a number of additional notational features in the BRD which were deliberately not shown on the UBRDs. This reflects an attempt to keep the UBRDs as simple as possible for the users to read. Since only the analyst will work with the BRD a richer complexity is arguably feasible. Many (though not all) of the additional features use the Harel blob.

The method for numbering states, events, triggers, messages and conditions is a simple serial one which uses the first letter of the construct type eg S1 for state number 1. The lines on the diagram separate the different object life histories identified earlier. There is an important reason for adding this feature to the diagram. It is proposed that the BRD can be used to describe an important type of business rule eg rules of the type that 'one customer may place many orders'. The reading of the diagram should be interpreted as follows. Within an object it may be assumed that there is one-to-one correspondence between one instance of a state of an object and the other state instances within the same object. For example, it can be assumed that there is a one-to-one correspondence between state S3 of the object order, ie order created and its next state S6, invoiced order. (Please refer to figure 2.14 to cross-reference these state codes.) However, for relationships across object boundaries, the default should be read as a one-to-many correspondence. So, for example, S1, the good customer state may have many S3 states, ie orders created or indeed many S4 states, outstanding order items created. If the nature of an inter-object relationship is one-to-one, this can be enforced by including an additional condition which would preclude many instances. In similar fashion if there is a finite upper or lower limit to inter-object relationships this too can be enforced through the use of an appropriate condition. So, by inspecting dependencies which cross object boundaries, business rules of the type 'one X may have many Y' can be inferred where X and Y are different objects.

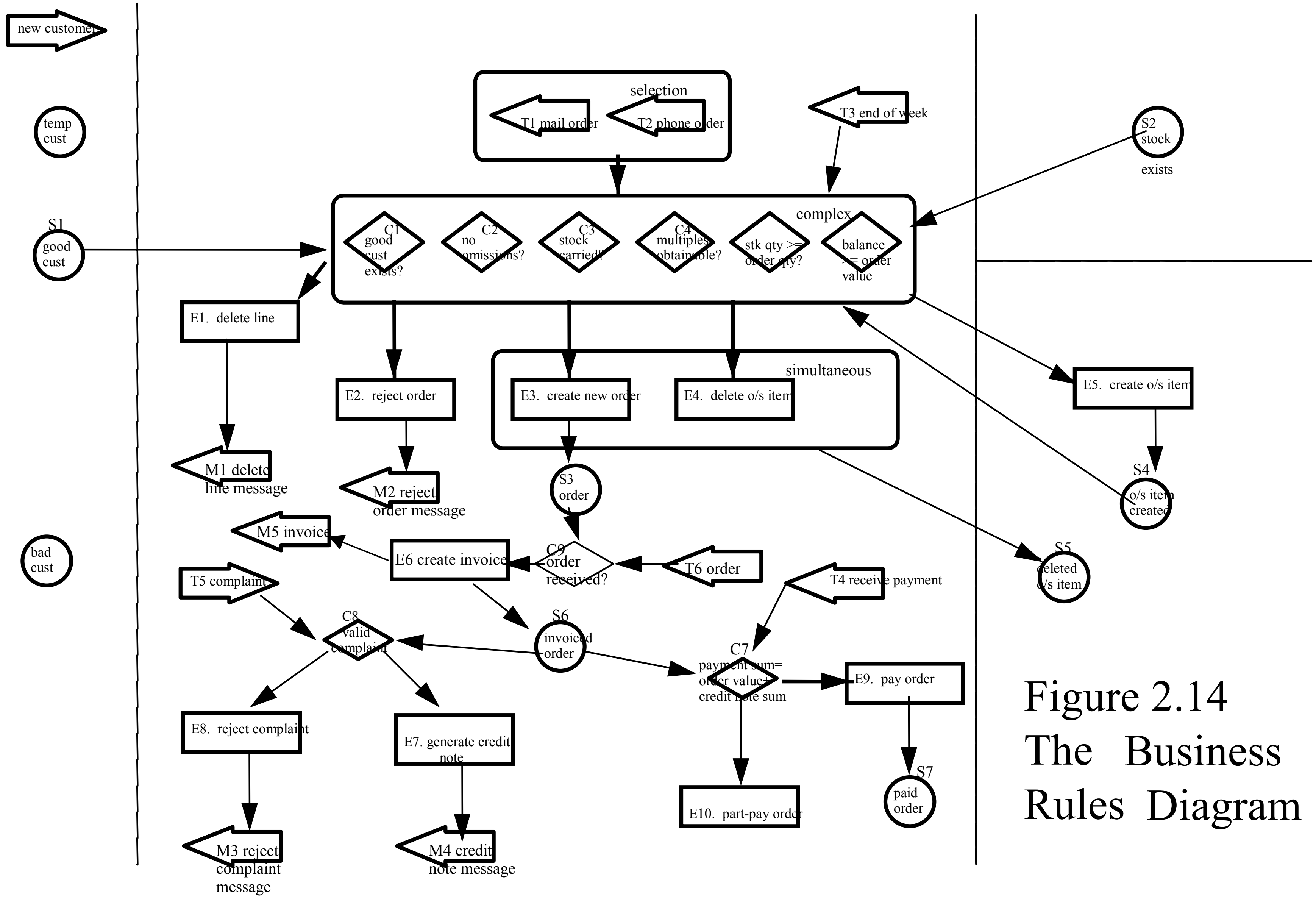


Figure 2.14
The Business Rules Diagram

In this example it may be assumed that the rules for processing telephone order and mailed order are identical and therefore only one state is required. However, it still may be helpful from a business policy perspective to model the business rule that 'orders may either arrive as telephone orders or mailed orders'. The Harel blob is used to depict this by encapsulating the two triggers, ie that either a telephone order or a mailed order will trigger the succeeding events. See figure 2.14. This allows rule one in the list of anecdotal business rules to be modelled using the BRD. Thus the BRD is arguably able to model all types of anecdotal business rule.

In some business rules many possible outcomes are possible. This was noted earlier in figure 2.9 where a number of conditions and indeed a number of different business events arose as possible outcomes of the answers to the conditions. This is known as a complex condition. This is why the definition of a business rule includes the term 'context'. The term context is taken to mean that over many executions of a business rule, different outcomes may eventuate depending on the values of states and the evaluation of conditions.

One additional feature in the BRD but not in the UBRDs, is that states which are used in testing a particular condition are connected to that condition (or complex condition). Notice that in figure 2.14 there are three predecessor states which influence the complex condition described here, ie stock exists, outstanding order item exists and customer exists. However, because this is a complex condition it is possible that any one of these states affects only some (but not necessarily all) of the decisions inside the complex condition. In fact, the event specification table which is discussed later details the linkage, but this is not indicated graphically. All that can be inferred graphically is that these three predecessor states each affect at least one of the decisions inside the complex condition. On the other hand, the advantage of this topological approach is that any condition, event or state only needs to be depicted once and once only on the diagram. This was a considerable advantage over other approaches which were considered. It also significantly reduces the number of connecting arcs required to link conditions, states, events and signals. Without this feature, the BRD quickly becomes much more difficult to read as it would contain many, many more arcs.

Events 3 and 4 in figure 2.14 have a surrounding Harel blob with the label 'simultaneous' included. This specifies that the two events are to be executed simultaneously (or effectively so). In this case when a new order is created from outstanding items, each outstanding item is deleted, so we have two events, ie the creation of a new order and the deletion of one or more outstanding items. This arises because of the different levels of abstraction involved. The BRD models the complete system, but because of the need to simplify and to work with the mental constructs of users, the system is broken up into components (or objects). In doing so however, this may lead to the need to specify a 'system' event which covers events in two different objects as is the case here.

Earlier with the OLHs for the sample case, credit notes, part-payments and overdue invoices were perceived as being parallel states in that they could occur independently of each other. This was a false perception and illustrates Vitalari's point (1992) about assertions and assertion testing. First of all it was decided that the overdue order state was unnecessary since it served no useful purpose, ie the event itself and the creation of the state did not impact on the business rules of the system (at least as they are specified within this case study). For example, no UBRD contained a condition which needed to reference an overdue state. Figures 2.12 and 2.13 contained references to a credit-note order state and a part-paid order state respectively, but when constructing the BRD it was found that these states were not predecessors to any other business rule. Consequently both states were removed from the BRD as they served no useful purpose also. This is a good example of where the elaboration of the OLHs into the BRD allows the modeller to examine and specify the problem in a more efficient manner.

Finally the notation for the event specification table is discussed. Considerable effort has been made to keep the diagram as simple and as intuitive as possible. Yet, at the same time there is additional information that can be recorded which would need to be defined in a full information system specification. For example, the ability to indicate the cardinal relationship between the number of instances of each input state to one event is important. Without this, an aspect of completeness of a business rule is omitted eg that several outstanding order items which now have sufficient stock are grouped together to form a **single** new order.

An example of the event specification table is set out in table 2.3. There are six columns representing events, triggers, messages, conditions, predecessor states and successor states. Each line/row within an event represents one scenario in which the

Event	**Trigger**	**Message**	**Conditions**	**Pre-states**	**Post-states**
E1	[T1/T2]	M1	C1, **C2** C1, C2, **C3** C1, C2, C3, **C4**	S1 S1, $S2^+$ S1, $S2^+$	-
E2	[T1/T2]	M2	**C1** C1, C2, C3, C4, **C6**	S1 S1, $S2^+$	-
E3	[T1/T2]	-	C1, C2, C3, C4, C5, C6	S1, $S2^+$	S3
E3, $E4^+$	T3	-	C5	$S4^+$	S3, $S5^+$
E5	[T1/T2]	-	C1, C2, C3, C4, **C5**	S1, S2	S4
E6	T6	M5	C9	S3	S6
E7	T5	M4	C8	S6	-
E8	T5	M3	**C8**	S6	-
E9	T4	-	C7	S6	S7
E10	T4	-	**C7**	S6	-

Legend for conditions:

C1 good customer exists?
C2 no omissions?
C3 stock carried?
C4 multiples obtainable?
C5 stk qty >= order qty?
C6 balance >= order value?
C7 payment sum = order value + credit note sum?
C8 valid complaint?
C9 order received?

Table 2.3 Event Specification Table

event may occur. If it is possible that more than one instance of a state is involved then the repetition superscript '$^+$' is used to indicate one to many instances. Under the conditions column, those condition numbers which have an influence on an event are positioned in the sequence in which they occur. If it is required that in order for the event to execute the answer to the condition is no rather than yes, then the condition is displayed in bold. Note the inclusion of a line in the table where the event is enumerated as 'E3, $E4^{+}$'. This indicates that the events are executed 'simultaneously' as discussed earlier and tells the reader that several (deleted) outstanding lines will make up one new order. Note also that event E3 is described separately for normal incoming orders.

2.6 Validate Business Rules Diagram

Once constructed, the Business Rules Diagram needs to be confirmed with the users some or all of whom will have assisted in the earlier process of construction. Clearly, there are many ways in which this could be done. It is strongly recommended that this is done involving users as far as possible. Discussion here reflects that philosophy. Three steps are identified and these are structured walkthrough, abnormal life analysis and amendment analysis.

2.6.1 Structured Walkthrough

It is proposed that the analyst would conduct a structured walkthrough to confirm the logic within the BRD. Typically the analyst will have prepared the BRD from material obtained at previous meetings and through discussions with individual stakeholders. Two options are possible. One is to show users the completed BRD and to live with the problem that the richness of notation and the overall complexity may confuse users. The other is to only use the UBRDs while working with users. This implies that the analyst will have to translate any questions to the relevant UBRD. The latter option is recommended, though it is quite possible that some users would take to working with the BRD. The main benefit obtained is that the logic behind the BRD is validated with the stakeholders. If errors or omissions exist, hopefully these can be incorporated into the model during the walkthrough. If not, the meeting can be reconvened once the BRD has been corrected.

2.6.2 Examining Abnormal Lives

Because the BRD is constructed initially from OLHs, the logic within the BRD is therefore based on and limited to the logic inherent in what may be termed 'natural' or normal lives as opposed to abnormal lives. An abnormal life is an exception to what would be the normal sequence of events. For example, if it is decided that a bad customer is to be removed because of non-payment of debts and therefore the debts are to be written off, this would be considered an abnormal life, because a natural deletion should only take place if all bills had been paid and therefore present no problem. Consider the customer OLH in figure 2.6. Initially a deleted state was not considered but the object life history could be extended to include a deleted state to accommodate the removal of 'bad' customers. Note that when this issue is examined using the BRD rather than the OLH, it is much more likely that the problem of existing debt will be seen. Further, arriving at a good resolution is more likely if the BRD is used.

Now the abnormal termination of the customer object would require that contingencies be identified for object customer states 'temp' and 'bad'. If the state is 'temp', then the customer has never been a good customer and therefore has never had any debts. This is what state 'temp' is taken to mean in this example. So, customer removal can proceed immediately. If the state is 'bad' however, then it is possible that several unpaid invoices exist, in which case perhaps these need to be removed before the customer object is terminated. The above illustrates the kind of discussion that could take place with a group of users and their analyst.

After the BRD is confirmed as correct, each OLH would be reviewed and using the stakeholders' knowledge, a list of abnormal life scenarios would be generated. Each scenario would be walked through with the group using the UBRD and a conclusion reached concerning policy on that issue. It would be the responsibility of the analyst to try to ensure that issues relating to elsewhere in the system were addressed. This could be done by the analyst showing one set of users the UBRDs from other groups where appropriate. As well as identifying new scenarios (and therefore potentially new requirements), examining abnormal lives also serves to reinforce the legitimacy of the natural life scenario and therefore increases confidence in the correctness of the BRD.

2.6.3 Analysing Amendments

The states identified in the OLHs are ones which are crucial to the basic underlying logic of the system. However, most information systems require other kinds of amendments, ie ones which do not or at least should not alter the business rules, eg changing a customer's address or more contentiously changing the quantity on an order line.

The BRD (and UBRDs) can be used to determine the impact of these amendments on the system. The example of a change in customer address is a relatively trivial one here save for the 'engineering change' decision as to when to implement the change of address. However, consider the other example in which the quantity on the order line has to be changed. What if the customer is allowed to request a reduction in the order by telephone after the order was sent? The BRD is examined to identify the states in the life history of that order to determine the impact on the rest of the system. What should be done if the order has become an invoice? It needs to be decided if it is not permitted to make such a change after the order becomes an invoice or if some other action must ensue, for example, the setting up of a credit note equivalent in value to the reduction in quantity. Clearly, if the latter route is chosen, this amendment has an impact on the system and therefore becomes a significant amendment which affects business rules. Without the aid of the BRD it would be more difficult to see the implications of amendments with respect to the rest of the system and to decide what action can be taken.

Once again, this activity should be undertaken with groups of users. The stakeholders' knowledge of the business will be indispensable in identifying the likely scenarios and also in suggesting possible solutions. The above is an illustration of the kind of dialogue that could ensue regarding one possible amendment. By examining each OLH in turn user groups could brainstorm possible amendment scenarios. The list generated could then be prioritised, so that later groups could focus on those which had important ramifications as requirements. Thus amendment analysis is a mechanism for identifying some types of business rule which may have been overlooked.

2.7 Summary

This chapter has described a technique for capturing business rules in a human activity system. The Business Rules Diagram has been presented as a practical tool for identifying and modelling assertions about business rules during information requirements determination. The early clarification of such assertions is considered vital to the subsequent development of an information system as is the ability of such a model to be used as a vehicle for reasoning about the consequences of assertions and their impact on the system as a whole. The steps in constructing the diagram with users have been described in a collaborative way, ie involving users at almost every stage and through that involvement recognising the importance of the 'people side' of information systems development.

The purpose of this chapter was to provide the reader with an appreciation of the general direction and approximate end-point of this research project. It is re-iterated that it was not the purpose in this chapter to explain or justify how technical ideas and concepts were arrived at nor was it the purpose here to compare the BRD with other published literature. That is the purpose of later chapters. However, it is hoped that sufficient background has been imparted for the reader to relate to discussion in the following chapters. For example, it is now possible for the reader to appreciate the detailed wording of the research question, but without the benefit of this chapter, the reasons for the detailed wording would not be so clear.

Chapter 3

Research Question and

Significance of the Research

3.1 Overview

This chapter has two purposes. The first purpose of this chapter is to introduce the research question. Since the diagram had evolved over the course of the research, it was decided to use the BRDv3 (Business Rules Diagram version 3) as the basis for presenting the research question. The BRDv3 contains a number of features of direct interest to the research at this point. These features are also considered part of the research question and are thus outlined to provide a complete description of the research question.

The second purpose is to discuss the significance of the research. It is appropriate to discuss significance immediately after formalising the research question as it is then possible to show the direct links between these two important aspects of the research. Figure 3.1 is a cognitive map summarising the main arguments with respect to significance. Throughout this thesis, cognitive maps or variants of them have been used as a means of summarising arguments which are relatively complex. Cognitive maps are used in the following way. Each bubble represents a fact, belief or concept that is considered important in a logical argument; arrows between bubbles indicate some connection possibly but not necessarily causal. For example, in figure 3.1 the assertion that this research demonstrates how to design diagrams strengthens the conclusion that the research overall is significant. This is a causal connection. On the other hand, in figure 3.1 the Harel blob is one kind of tactic to assist complexity management, so here the connection is not causal; rather one is a set of features, the other is a member of that set of features. In such situations the term ‘ISA’ appears to indicate 'is a kind of'. Cognitive maps are therefore a subjective device whose main purpose is to communicate the main threads of an argument to its readers.

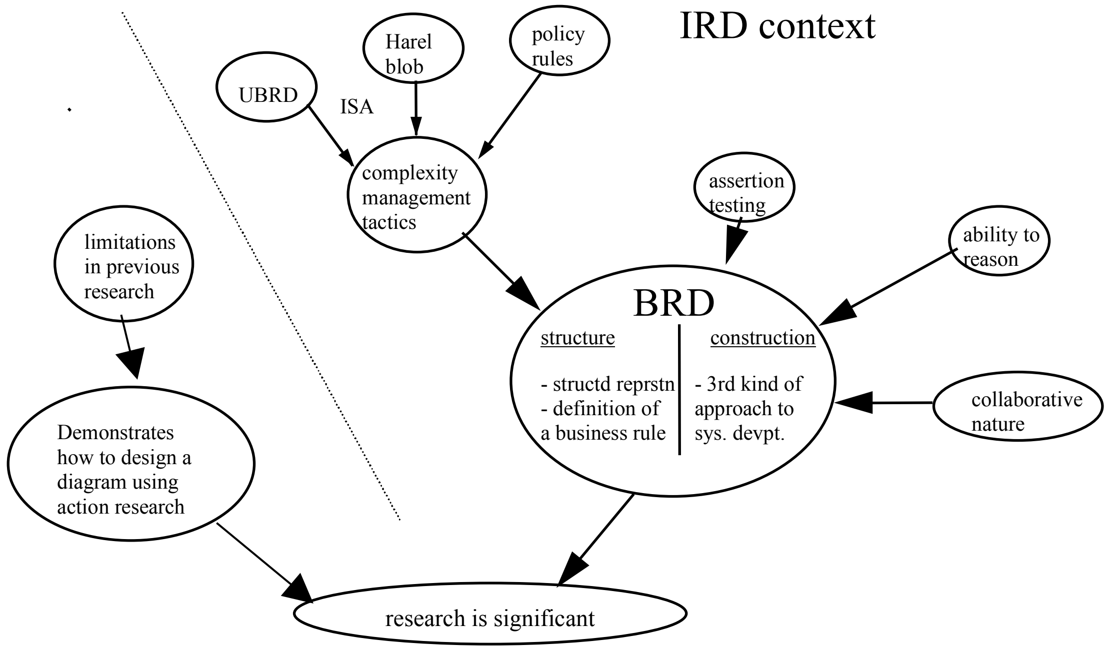


Figure 3.1 Cognitive Map summarising Significance of Research

The significance of the research is reported at two levels. On the left-hand side of figure 3.1, discussion takes place at the level of general research, ie in terms of this work as a contribution to the discipline. Specifically, I have shown how to design new diagrams using action research. On the right-hand side of figure 3.1, the significance of the research is discussed more specifically, ie in relation to the BRD and IRD (Information Requirements Determination).

The remainder of this chapter is divided into sections which address firstly the research question itself and then the significance in terms of the major bubbles on figure 3.1.

3.2 Research Question

One observation that can be made from the discussion in the last chapter is that this research is actually about developing a diagramming technique rather than a diagram. In other words, this research is not only about deciding which graphical constructs should appear on a diagram but also about how these constructs are identified and the steps that are involved in doing so. Indeed, later in this research it will be demonstrated that the two are deeply intertwined (chapter 12). Thus developing a technique includes developing a diagram. Having said that, I hold the view that deciding on the constructs on a diagram is a primary goal of this research.

The research question is made up of two related parts. The first part concerns the structure of the BRD. The term structure refers to the specific components or constructs which make up the diagram. In other words, the main aim of this research is to identify those characteristics of business rules which are to be shown graphically. The second part of the question relates to the steps involved in constructing the diagram. The process of constructing business rules is not trivial. As will be seen later, there are many issues and indeed many possible steps in such a process. So a second concern is to consider the construction of the BRD. Accordingly, the research question can be stated more formally:

'What is the structure and what are the construction steps of a diagrammatic technique for capturing business rules while determining information requirements in a human activity system?'

Over the course of this research there were many versions of the research question. In all six versions of the BRD were developed and each version contained a changed set of characteristics to be tested. This was true both in terms of structure and construction steps. At the beginning of the testing of each version a list of characteristics was generated. These are documented as templates in appendix 1 and were used to guide the interviewing process described later.

While stating the research question at this high level of abstraction is useful since it applies to all stages of the research (ie to all versions of the diagram notwithstanding the shift in emphasis discussed in chapter 1), it does not clearly indicate what specific matters of interest were the focus of the research. Of course as stated in the previous paragraph, such matters changed over the different versions of the diagram and so it would be impossible to define them specifically as one research question.

However, since these structure and construction issues evolved and matured over the course of the research, it is reasonable to consider that those characteristics tested in the last version of the diagram are the most relevant in terms of defining the nature of the research question. Table 3.1 contains is a list of characteristics of the BRDv3 that were used to question participants in action research study 2.

BRDv3
Modelling in states + events + conditions + triggers + messages (SECTM)
Policy rules vs processing rules vs implementation rules
Context of a business rule (ie SECTM)
Drawing BRD before objects are specified in detail
review of construction process: - policy rules - business events + signals - objects - object life histories - UBRD - BRD - validate BRD
notational review - Harel blob for selection - Harel blob for parallel states - Harel blob for complex conditions - Harel blob for simultaneous events - Asterisk to indicate repetition event
Event specification table - showing ‘manyness’ - showing logic of conditions
Inclusion of ‘static’ business rules eg one customer may have many orders
Clarifying OLHs eg some states no longer useful
UBRD vs BRD
Structured walkthrough
Abnormal life analysis
Amendment analysis
Modelling events or states at the instance level rather than object level
Significant modifications, ie ones which affect the processing of the system are only shown, ie non-significant ones not shown

Table 3.1 Summary of Features of BRDv3

It is important to note that table 3.1 was in effect a working list of the important ontological issues considered relevant at that point. Because it was used as a template for questioning participants during interviewing, these issues were therefore thoroughly embedded into the research process. All these features have been discussed and defined at some length in the previous chapter. However, for completeness sake, the following paragraphs summarise each of these features.

3.2.1 Specific Constructs of State, Event, Condition, Trigger and Message

A business rule was defined to include states, events, conditions, triggers and messages. The purpose of this first question was to explore whether each construct was needed.

3.2.2 Policy Rules

The technique captures policy rules only, ie those rules which determine processes and procedures in a human activity system. Once policy rules have been defined, processing and implementation rules can follow. This question examined whether policy rules were perceived as workable and appropriate.

3.2.3 Context of a Business Rule

This question explored whether the set of constructs used was perceived to be a relatively complete and comprehensive set for describing a business rule.

3.2.4 Direct Modelling of Business Rules

Traditionally in information systems development, behaviour models such as object life histories are constructed after other kinds of model have been developed, eg a data model. The approach here describes a way of bringing forward the specification of states, events, conditions and signals as early as possible in the development cycle and without the need for constructing a full data or object model.

3.2.5 Construction Process

This question explored whether the particular construction route chosen was perceived to be appropriate.

3.2.6 Notational Review

With the use of the Harel blob a number of rich modelling constructs have been able to be incorporated on the BRDv3. Provision was made for selection, parallel, complex and simultaneous situations. In addition an asterisk was used to indicate repetition.

3.2.7 Event Specification Table

The event specification table allows for more detailed specification of the information system and heralds the beginning of the next phase of the specification process. Details of events, triggers, messages, conditions, pre- and post- states together with grouping and logic information are recorded.

3.2.8 Modelling of Static Rules

It was speculated that it might be possible to add static business rules such as 'one customer may have many orders' to the BRDv3. Where a component of an object may occur many times with respect to the object itself (eg one order may have many credit notes), additional notation had been provided for that also.

3.2.9 Clarifying OLHs (Object Life Histories)

The BRD technique has provided a mechanism for clarifying OLHs. It was speculated that initially perceived states were often unnecessary or inappropriately specified. This question examined that speculation.

3.2.10 UBRD (User Business Rules Diagram)

A mechanism for reducing complexity for users has been provided through the UBRD. Yet it arguably enables a rich set of information to be available to the analyst. The flowchart 'look and feel' of the UBRD is submitted as a good intuitive way of encouraging users to work with business rule logic.

The BRD technique arguably makes it easy to reason about the human activity system and to envisage the consequences of changes to the human activity system. This is facilitated through the use of structured walkthrough sessions, abnormal life analysis and amendment analysis. Each of these are now briefly discussed.

3.2.11 Structured Walkthrough

A structured walkthrough allows for the analyst to confirm earlier statements made by participants by reviewing the deliverables of techniques. Where errors or omissions have occurred these can be corrected.

3.2.12 Abnormal Life Analysis

Abnormal life analysis specifically examines exceptional or unusual scenarios in a UBRD.

3.2.13 Amendment Analysis

Amendment analysis explores the impact of amendments in terms of a UBRD.

3.2.14 Modelling at the Instance Level

This question explored whether it was considered appropriate to model business rules at the instance level, ie in terms of modelling instances of states of objects (eg manyness requires that we think of the number of instances of states involved in a state change).

3.2.15 Significant Modifications

A significant modification is one which changes a state in the BRD and thus can be perceived as a change to the information system by an external observer. This was thus a key aspect of the BRD design.

In summary, the above features constituted aspects of the research question while working on the BRDv3 and as such were an integral part of the discussions and questioning that took place.

3.3 This Research demonstrates how to Design Diagrams using Action Research

In this section there is firstly an overview of the limitations of previous research. This leads to a brief discussion suggesting why such limitations might exist before explaining why this research is considered a unique contribution.

3.3.1 Limitations of Previous Research

There is a paucity of research in the literature regarding the designing and testing of techniques in the information systems field (Lyytinen and Hirschheim 1987, Galliers 1992 Ryan 1995). Moreover the following authors note:

- *'New design tools are created and marketed every day, but there is little research available to enlighten information systems designers and managers about the effects or implications of incorporating these tools into their design environments. This situation is relatively common in the information systems design field, where in spite of intensive research activity centering on information systems and information systems design, relatively little is known about these topics*' (Calloway and Ariav, 1995, p77),

- '*At this point we thought that we would examine the way that other systems development techniques had been validated. This was more difficult than at first thought because there are not many system development techniques available to examine and those that were available did not appear to have been validated'* (Fitzgerald 1991, p663) and
- '*Despite the significance of specifying information requirements to the successful design of systems and the large body of descriptive and conceptual literature on methodologies, surprisingly few empirical studies have been conducted into the specification process itself'* (Vessey and Conger 1993, p179).

One important reason for this paucity, I believe, is to do with the roots and history of information systems as a discipline. The following distinction in emphasis is often now made between software engineering research and information systems research. In software engineering, the central focus is on design. In many ways the design is an end in itself to a software engineer. If this research were solely concerned with software engineering perspectives, it would be sufficient to demonstrate theoretically that the BRD can model business rules in a range of information systems from a purely technical perspective. Social questions such as whether the BRD is user-friendly, whether it can be learnt or how it can be used sensibly in the context of a systems analysis are of relatively peripheral interest to the software engineer. In contrast, to the information systems researcher these questions are central.

3.3.2 The Contribution of This Research

An important contribution of this research is that it demonstrates how to go about **developing new diagrammatic techniques using action research**. In a comprehensive literature review of action research in information systems over the last 25 years, Lau (1997) classified these studies under a number of headings. No action research study in this review focused on developing new diagrams. At the time of writing I can find no published work which actually uses action research for designing new diagrams or subsequently testing them out. The closest type of study to this research are those studies which use a field study approach to test a diagram which was previously designed by the researcher. But this type of study does not directly involve analysts or users as **co-researchers in the actual design process**. Examples include Calloway and Ariav (1990), Fitzgerald (1991) and Flynn and Davarpanah Jazi (1998). With regard to Calloway and Ariav's work, they had previously designed a dialogue chart and then used field trials with students with interview follow-up to gauge the success of their diagram. They did not use action research while designing their diagram. Similarly, the work reported by Fitzgerald reflected on how a type of action diagram might be validated, but again action research was not used to design the diagram. Flynn and Davarpanah Jazi tested an event flow diagram with users in the context of user led development. Again, the diagram had been designed prior to the research commencing. These approaches reflect a different mindset concerning developing diagrams. In these approaches it is assumed that the designer knows and understands the problems of analysts and users and has addressed these in the candidate diagram. The object of these studies was therefore to test (or as Fitzgerald puts it to validate) the diagram after it had had been designed. Using action research, the analyst or user as co-researcher can be involved in the design process as well as the test process. This offers the opportunity for other aspects, for example social aspects which the researcher may have little insight into, to be addressed while the diagram is still being designed.

I am not arguing however, that this research covers all aspects of developing new diagrams or indeed that is the only way to go about it, but I am asserting that it contributes to an understanding of the issues involved and that it is an exemplar of one way of going about developing new diagrams. Moreover, along the way many lessons have been learnt which would be useful for any researcher with similar goals.

3.4 Structural Aspects of the BRD

Much of the work in the first action research study focused on the arguments for incorporating additional constructs into the business rule which led to the conclusion that a full description of a business rule demanded a more comprehensive description involving a rich set of constructs. Two aspects of this work highlight the significance of the research. These are the structured representation of a business rule and the provision of a comprehensive definition of a business rule.

3.4.1 Structured Representation of a Business Rule

In the following discussion a business rule may be taken as any functional information requirement, in contrast to a non-functional requirement which for example may refer to a performance or economic requirement (Greenspan et al. 1994, IEEE 1984).

There has been continuing attention to IRD in the literature, for example as far back as Miller (1964). Most authors split the process of IRD into two main steps, ie requirements acquisition and requirements modelling (Valusek and Fryback 1987, Greenspan et al. 1994) although there exceptions to this (Zmud, 1983, Vitalari 1992). Currently, there is much interest in the problems of the early stages of IRD concerned with acquiring or eliciting requirements. For example, Vidgen (1997) looks at stakeholder approaches, Darke and Shanks (1997) examine a related concept known as viewpoint modelling and Flynn and Davarpanah Jazi (1998) recommend user-led construction of requirements. Other problems with requirements elicitation include emergent requirements which is to say that users themselves are not fully conversant with requirements or have not fully thought through the implications of requirements (Goguen and Linde 1993, Leifer et al. 1994). Clearly, clarifying and agreeing requirements, especially when political factions are involved or competing interests have to be traded off, is an important early first step before proceeding with the implementation of an information system.

However, techniques which are designed for requirements acquisition or elicitation tend to use informal representation, typically narrative statements of requirements. Informal representation carries the significant disadvantage that it is unstructured, ie it does not require the identification of specific types of constructs to be identified which are perceived to form part or all of a requirement. For example, Byrd et al. (1992) identify eleven requirement acquisition techniques. These are prototyping (Avison and Fitzgerald 1995), open interview (Davis 1982), brainstorming (Davis 1982), goal-oriented approach (Zmud et al. 1992), cognitive mapping (Eden 1988), variance analysis (Mumford 1985), repertory grids (Gutierrez 1987), scenarios

(Mittermeir et al. 1987), structured interviews (Davis 1982), critical success factors (Rockart 1979) and future analysis (Land 1982). To this Darke and Shanks (1997) add joint application design (Avison and Fitzgerald 1995) and focus groups (Leifer et al. 1994) and conceivably others could be added such as the Soft Systems Methodology (Checkland 1981) and the Strategic Assumption Surfacing and Testing (Mason and Mitroff, 1981). With the possible exception of prototyping, there is no structure in the end-point of any of these techniques in terms of pre-defined constructs and so it is possible that aspects of an information requirement (business rule) may well be omitted, ambiguous or assumed by the reader.

As an illustration of this concern over informal representation, consider Sowa's work (1984) on business rules. In his approach, the basic idea is to capture business rules as sentences in natural language (say English). The main components of the sentence are then identified as constructs in a graphical model. Sowa's work on conceptual graphs (Sowa 1984) provides the theoretical background for this research. In his book he uses examples such as 'the cat sat on the mat'. This is converted to three constructs, namely cat, sat, and mat and these are then shown with graphical connections. There is a problem with the natural language approach when presenting a complete business rule. The identification of for example cat, sat and mat has no classification structure in terms of types of construct. They are simply objects with some relationship. Unless natural language sentences are structured (or classified) in such a fashion to ensure that all aspects of completeness of a business rule are declared, ie so that constructs such as states, events, conditions and signals are **explicit**, then this a serious omission in the structure of a business rule.

The BRD on the other hand is quite explicit about the constructs which need to identified in order to define a business rule and thus can be considered a structured representation. The term **structured** is used here in preference to the term semi-formal and also to avoid confusion with the term formal which is often reserved for mathematical specifications (Pohl 1994). While it is recognised that requirements need to be comprehensive, there is some doubt as to whether users are able to fully understand 'computer-oriented' notations (Flynn and Warhurst 1994) and so this opens up the question of what approaches to modelling requirements are deemed appropriate. Having said that, though the extent to which users can work with diagrams is an important question for this research and indeed is considered (see chapter 12), in my view a more fundamental initial question is to establish the basic constructs required to depict information requirements.

Given that it is accepted that formalising (structuring) the representation of a business rule is valuable, a related question is how this would fit into the process of IRD and with information systems methodologies generally (Chatzoglou and Macaulay 1996). Three situations can be identified. These are firstly where no formal technique for IRD exists, secondly where a technique for IRD exists but it has an informal representation and thirdly where a technique exists which has a structured representation but not necessarily as complete as the BRD. Where no formal technique for IRD exists, the BRD technique must be flexible enough to permit the establishment of requirements, although my strong personal preference here would be to incorporate tested techniques designed to address the kinds of social and political problems for example inherent in such situations. Accordingly, the BRD has steps which (simplistically) allow users to brainstorm aspects of business rules. The second situation, ie where there is an informal statement of requirements, is the stereotypical scenario for using the BRD technique assumed in this research. In other words it is assumed that some requirement acquisition technique has been used to some acceptable degree. In the third situation, consideration has to be given to how to convert from one representation to another. Though there may be difficulties with different starting points, an advantage claimed for the BRD is that it provides a comprehensive basis for later stages in the development of an information system. As long as information systems can be easily developed from the BRD, the BRD can be seen as an appropriate endpoint to IRD in that it is a complete and structured representation of preceding effort in (functional) determining information requirements.

In summary, two arguments have been made in this sub-section. The first is that structured representation of information requirements is valuable because it makes explicit that which is important in specifying requirements. Secondly, there is a need for an agreed and formally represented endpoint to the IRD process which is amenable to later information system development. Informal representations are considered unamenable since they are ambiguous, open to assumptions or contain omissions. The BRD is argued to be more complete, structured and less ambiguous and therefore not open to assumption.

3.4.2 Comprehensive Definition of a Business Rule

One important outcome of this research has been the construction of a comprehensive definition of a business rule. The definition itself gradually evolved over the course of the research and involves many facets. It contains a list of the constructs of a business rule namely states, events, conditions and signals. In addition the definition stipulates something of the context and nature of a business rule in *'a business rule is an explicit state change context in an organisation which describes the states, conditions and signals associated with events that either change the state of a human activity system so that subsequently it will respond differently to external stimuli or reinforce the constraints which govern a human activity system'*.

One significant aspect of a business rule is the idea of a state change context in a business rule. This is a unique solution in business rules models, ie one not seen in other definitions. Most other definitions of a business rule focus on conditions and do not even involve the state as a construct (eg Appleton 1988). In my definition states are those anchors upon which the business rule is modelled. Without states it is much more difficult to put a boundary on a business rule. In other words, the state change context makes business rules modelling more tractable.

3.5 Construction Aspects of the BRD

The second part of the research question attended to construction aspects of the BRD. Much of the motivation for this concerned issues or ideas already published in the literature such as achieving completeness, working with participants with different backgrounds and preferences. To that extent this research was not unique or significant in the way that construction steps were implemented, rather it reflected good practice. However, one notable exception to this is described in the next sub-section.

3.5.1 A Third Generic Approach to Systems Development

A potentially significant outcome of this research is the assertion that the BRD can be generated **directly** from informal requirements. Only a cursory identification of objects and states is necessary in order for a meaningful dialogue with users to ensue and therefore for the BRD to be constructed. This represents a third route to developing information systems, the other two routes being approaches involving process modelling first such as in for example Gane and Sarson (1979) and those involving data (or object) modelling first such as in Information Engineering (Martin and Finkelstein, 1981). However, without further research for example, by fully developing information systems in a number of contexts using the BRD as part of a complete methodology, the position here is that there is **evidence to speculate** that the BRD represents a third generic approach to systems development.

3.6 The BRD has Features considered useful in resolving Difficulties in IRD

In the first sub-section below, there is a review of features of the BRD which are argued to specifically address problems in IRD to some extent. This is then followed by a second sub-section in which there is a review of the IRD literature which identifies aspects of IRD which may be considered problematic. Also in this second sub-section, a discussion takes place which relates how the BRD features are argued to address or resolve these IRD problems. Inasmuch as IRD is agreed to be a problematic activity, any technique which assists in the specification of information systems requirements generally, makes some contribution to easing the problem. The case is made by demonstrating how specific features of the BRD address specific problem areas in IRD.

3.6.1 Features of the BRD which aid IRD

In addition to the structural and construction aspects of the BRD already highlighted, the BRD contains a number of features which can assist in ameliorating some problems in IRD. These features are:

- complexity management tactics,
- assertion testing,
- collaborative nature and
- the ability to reason with the diagram.

The following paragraphs provide some background to these features. While it may be argued any one feature on its own might be limited in the extent to which it can be said to make a real impact to the process of IRD, as a battery of tactics it is submitted that such overall impact is significant enough to contribute to improving the process of IRD generally.

Complexity management tactics

Three aspects of the BRD in particular are argued to assist in managing complexity, ie where the human participants are in danger of becoming overloaded with detail. The issue of complexity and its consequences for the design of the BRD are discussed later in the thesis. The three tactics described here which mitigate perceived complexity are the User Business Rules Diagram (UBRD), the Harel blob and policy rules.

The UBRD breaks up the specification of business rules into single use cases which correspond typically to single business rules. This means that only users directly concerned with a particular business rule need be involved in its specification and therefore other business rules unconnected with a user need not appear on the same diagram. Clearly this reduces the volume of information any one user needs to examine at one time and also only aspects directly of relevance to a particular business rule need to be shown.

The Harel blob is a visual device for reducing the number of connecting arcs on a diagram yet retaining the informational power of what is depicted. It is argued to be intuitive in the way that it depicts this information and also easy to learn. The Harel blob is used in both the UBRD and the BRD as a means of communicating information in as simple a manner as possible.

The third complexity management tactic of the BRD is its focus on 'policy' rules. It is proposed that is not necessary or even desirable for an organisation to lump all of its rules together and call them business rules. Instead, it is submitted that (at least) three types of rule in a business can be identified and that each might be better kept in separate repositories. Furthermore, the topmost level of business rule (the policy rule) can be captured as a conceptual model of the business by showing rules which are unconstrained by how people in the organisation execute these rules or indeed how a computerised information system might implement these rules. It qualifies as a complexity management tactic because, as will be demonstrated later in this thesis, it effectively reduces the number of rules that have to be depicted (since it only deals with policy rules) and also the amount of detail associated with any one rule is restricted to policy aspects.

Assertion testing

Earlier it was noted that requirements analysis (or IRD) consists of three steps namely requirements generation, assessment and specification (Vitalari 1992). Realistically the BRD is a technique which supports the specification of requirements and so only directly contributes to Vitalari's third step. Importantly however, Vitalari (1992, p163) recognises that the '*initial hypothesis is usually ill-defined or ambiguous and requires elaboration to develop the system*'. Often the problems of ill-definition or ambiguity only manifest themselves at the specification stage (if not later) and therefore one role of the BRD would be as a gatekeeper in confirming earlier assertions with users about the business. In this regard the process of IRD may be seen as iterative rather than linear. So for example, where there is not full agreement amongst users about a particular rule, this would need to be resolved by re-examining the initial hypothesis (assertion) or revisiting earlier IRD processes before continuing.

Collaborative nature of the BRD technique

I have used the term collaborative nature to describe a participative environment for developing the BRD. This is achieved by taking every opportunity for involving users and obtaining feedback from users. The outcome of this is a better understanding of the problem domain and a better rapport with users as well as a better quality specification.

Ability to reason

In addition to being able to model the business, the BRD offers the opportunity for analyst and users to be able to reason about the consequences of changes to the business. Though it depends on the users being sufficiently comfortable with the diagram to be able to reason about consequences and changes to the business, clearly, this would be a powerful asset in assisting businesses review current operations as well as plan for the future. Users working with the analyst could use the UBRDs to think through the implications and impact of changes arising out of for example a business process re-engineering study.

3.6.2 Problems in IRD

In this sub-section a number of problems in IRD from the literature are identified which are seen as relevant to this discussion. After that, BRD features considered helpful in ameliorating them are recalled and linked to these problems.

IRD is seen as highly problematic in the literature. For example, a review of IRD literature revealed that:

- IRD is propositional,
- IRD is difficult to perform,
- IRD is performed in a social context,
- there are communication difficulties with IRD,
- broader social and organisational issues are often overlooked in IRD,
- user expectations are unrealistic in IRD and
- IRD is becoming harder because of increasing complexity and rate of change.

(Bostrom and Heinen 1977, Davis 1982, Bostrom and Thomas 1983, Kaiser 1985, Hirschheim 1986, Dumdum and Klein 1986, Valusek and Fryback 1987, Dos Santos and Hawk 1988, Corbato 1991, Flynn 1992, Newman and Robey 1992, Byrd et al. 1992, Vitalari 1992, Tan 1993, Goldkuhl and Rostlinger 1993, Goguen and Linde 1993, Vessey and Conger 1993, Bickerton and Siddiqi 1993, Macaulay 1993, Leifer et al. 1994, Shand 1994, Pohl 1994, Greenspan et al. 1994, Flynn and Warhurst 1994, Ewusi-Mensah and Przasynski 1994, Shah et al. 1994, King 1995, McKay et al. 1995, Vidgen 1997, Darke and Shanks 1997, Flynn and Davarpanah Jazi 1998)

Propositional nature of IRD

Vitalari (1992, p163) in devoting a whole paper to the propositional nature of IRD notes '*Thus requirements are only propositions, hypotheses, or plausibility statements until the analyst and the users subject the hypothesis to a process of disconfirmation. .... by better understanding the dynamics of this process we can propose better approaches for the requirements analysis process....'*. The BRD can be used as a tool for **testing assertions** and so can assist the process of disconfirmation referred to above.

Difficulties in performing IRD

Davis (1982) identifies three main reasons why it is difficult to obtain information requirements generally These are:

- human information processing difficulties,
- the variety and complexity of information requirements and
- the complex patterns of interactions between users and developers in defining requirements.

Other authors refer to errors made while identifying user requirements. Problems occurred in attempting to completely, correctly and consistently identify information requirements with users (Bostrom and Thomas 1983, Leifer et al. 1994, Ewusi-Mensah and Przasynski 1994).

It is contended that **the complexity management tactics** of the UBRD, Harel blob and policy rules, can contribute to mitigating this difficulty. For instance focusing only on policy rules reduces human processing difficulties because less information needs to be shown on a diagram. Further, the complex patterns of interaction between users and analysts is to some extent determined by the nature and frequency of collaboration, ie the steps in the BRD technique and who it is decided is to be involved in each step.

Social context

IRD takes place in a social context. Typically, IRD involves at least one analyst and perhaps several users. It follows that all the problems arising in social situations will be present in the context of IRD. This therefore has implications for the kinds of issue that the IRD process has to address. For example Goguen and Linde (1993, p153) note that '*The problems of requirements elicitation cannot be solved in a purely technological way, because social context is more crucial than in programming, specification and design phases'.* Macaulay (1993) in referring to requirements capture as a 'co-operative activity' demonstrates an approach for involving all relevant stakeholders in gathering requirements. Both these papers contain an implied warning to the developer which is why they are referenced here. Their warning is that developers and researchers must accept the premise that IRD is a social process and that participation and co-operation are vital. The **collaborative nature** of the steps in the BRD acknowledges this social process by incorporating processes for involving users.

Communication difficulties

A common theme in papers on IRD is the view that communication difficulties exist generally (Davis 1982, Goldkuhl and Rostlinger 1993, Goguen and Linde 1993, Macaulay 1993, Leifer et al. 1994). Further, a number of papers specifically mention communication difficulties between analysts and users and amongst users (Bostrom and Thomas 1983, Kaiser 1985, Valusek and Fryback 1987, Dos Santos and Hawk 1988, Tan 1993, Shand 1994, Shah et al. 1994). Suggested possible reasons for this were differing conceptual models and often conflicting perspectives and worldviews which frequently resulted in breakdowns and misunderstandings (Darke and Shanks 1997, Vidgen 1997). The steps built into the technique for obtaining feedback from users (for example the structured walkthroughs) are specifically designed to reduce communication difficulties (ie **collaborative nature** of design). Further, the BRD presents a complete model for business rules to users based on states, events, conditions and signals. The UBRD for instance can be read by users and analysts alike using a consistent symbology thus reducing the propensity for misunderstandings. In other words aspects of the structure of the BRD may help here.

Broader social and organisational issues often overlooked

In some of the literature it was alleged that broader social and organisational issues are largely being overlooked. In other words in terms of the system design, purely technical rather than socio-technical solutions were being identified and relatively little emphasis was being given to human and social dimensions of development (Bostrom and Heinen 1977, Hirschheim 1985, Dumdum and Klein 1986, Flynn 1992, Newman and Robey 1992, McKay et al. 1995, Vidgen 1997, Flynn and Davarpanah Jazi 1998). While there is no mechanism in the BRD as such which ensures that socio-technical issues are taken into account, it is argued that the **collaborative nature** of the design encourages discussion and consideration of broader organisational issues as well as technical questions and so to that extent the BRD addresses this problem.

Unrealistic user expectations

There is some evidence that user expectations of the system are often unrealistic. For example, misperceptions may have arisen from failure to understand context and limited knowledge of technological capability (Shand 1994, King 1995). It is suggested that the required high levels of participation associated with using the BRD particularly the walkthroughs, amendment analysis and abnormal life analysis (ie its **collaborative nature**) provide users with an opportunity to clarify misperceptions thus achieving a clearer idea of what to expect from an information system. Further, the **ability to reason** about implications may cause users to reconsider their expectations in the light of a better understanding of a problem situation.

Increasing complexity and rate of change

Corbato (1991) alleges that increasing complexity and the rate of change in the environment makes it more difficult to specify requirements. Unfortunately, overall trends in business dictate that this situation is unlikely to reverse. Nevertheless, techniques which aim to simplify specification through **complexity management** tactics contribute to reducing the impact of this growing problem, as well as the **ability to reason** about the implications of change and the rate of change.

3.7 Summary

This chapter introduced a formal statement of the research question. Since the question evolved and matured over a number of version of the diagram, the research question is stated in terms of the last version of the diagram, ie the BRDv3. The detailed features of the BRDv3 being tested at this point were described.

Also in this chapter a number of arguments were outlined supporting the significance of this research. The first argument was that this research is an exemplar of how to design new diagrams using action research. On the right-hand side of figure 3.1, the cognitive map showed a case outlining the contribution of the BRD. With regard to the BRD itself, three arguments were submitted. Two of these were with respect to the structure of the BRD. These were arguments supporting the need for formality of representation (ie a structured representation) and the value of a comprehensive definition of a business rule. The third argument related to the construction of the BRD and concluded that there was evidence to speculate that the BRD represents a third generic approach to systems development.

The significance of the BRD was also discussed in the context of IRD generally. The IRD literature was reviewed and a number of problems identified for which the BRD was seen to ameliorate in some way. The features of the BRD which were seen to have significance in an IRD context were complexity management tactics, assertion testing, the ability to reason and the collaborative nature of the technique.

Chapter 4

Literature Review

4.1 Overview

There has been a continued and active discussion of business rules in both the practitioner and academic literature. With regard to the practitioner literature, especially in North America, there is a concern that there has been insufficient focus on business rules generally (Chikofsky 1990, Sandifer and Von Halle 1991a and 1991b, Jones 1991, Lucas 1993, Moriarty 1993a, 1993b, 1993c, and 1993d, Von Halle 1994, Baum 1995). The discussion in the practitioner literature is wide ranging. For example, it covers questions of how a DBMS environment can deal with business rule implementation options (Von Halle 1994) through to Joint Application Design (Lucas 1993). Much of the discussion however, quite naturally focuses on immediate solutions to existing problems. So for example proposals are often made illustrating how business rules could be integrated into data dictionaries (Sandifer and Von Halle 1991a) or how CASE tools could improve the documentation of business rules (Baum 1995) rather than specifically considering the development of **new** diagrammatic techniques.

As regards the academic literature, there are a number of examples presenting graphical solutions to modelling business rules. However there are some difficulties in undertaking such a review. One difficulty with a review of business rules literature is that there is a wide variety of names given to diagrams which arguably model business rules or at least some aspect of a business rule. Further, there is no standardisation of terminology in this area.

Another difficulty is the question of establishing a diagram's role within a methodology and also even whether within a methodology more than one diagram may combine together to describe business rules. Rumbaugh et al.'s OMT (1991) for instance involves constructing three models, namely an object model, dynamic model and a functional model. Here the state diagram is the chief dynamic model and is drawn after the object model. Methodologically, this makes some sense since the object model by definition identifies all object classes for the universe of discourse, but it also raises questions concerning the relationship of one diagram to others in a methodology and further how such relationships might impact a discussion which compares the BRD to other techniques embedded in a complete methodology.

Literature which surveys the area has difficulty in determining scope and in standardising on terms and definitions. For example, both Herbst et al. (1994) and Chow and Yeung (1996) present surveys of diagrammatic approaches to business rule modelling, though Chow and Yeung's survey encompasses behaviour modelling which is arguably broader in scope. Herbst et al.'s review is more straightforward in the sense that specific models of business rules are identified and reviewed, whereas Chow and Yeung's review is actually of methodologies and within that behaviour modelling and business rule modelling are discussed though there is no formal definition of a business rule provided.

A simple, practical approach has been taken to reviewing the literature. Later in the thesis, a learning framework is used to guide the research. A central idea in this framework is the notion of an ontological world. An ontological world contains a set of constructs which are considered to represent the real world. So, for example, the real world may be considered to consist only of states and so a diagrammatic technique would only model states. Thus the literature review has been segmented into a number of sections. These sections compare aspects of the BRD to diagrams or methodologies in the literature with similar ontological characteristics. Specifically the review is partitioned in terms of the main constructs of the BRD, ie state, event, condition and trigger. It was not considered necessary to consider message-based models, though a class of extended static models was added because they represent an alternative to the approaches represented by the four constructs of the BRD. Later in the thesis, the extent to which the BRD is able to capture the type of information contained in static maodels will be evaluated. The sections in this chapter are thus entitled:

- extended state-based models,
- condition-based models,
- extended static models,
- trigger-based models and
- event-based models.

It should be understood that due to the richness and diversity of notations, allocation to a single section was still sometimes problematic. Also because of the number of potential models that could be compared, not every model has been included here. Rather, models which represent each class have been selected. At the end of each section an evaluation with respect to the BRD is carried out to highlight the major differences that exist.

4.2 Extended State-based Models

There are many models to be found in the literature which are based on the modelling of states and which also contain extensions or additional information to some degree. Indeed in one form or another they are increasingly popular as one technique in a complete development methodology (Rumbaugh et al. 1991, Booch 1991, Shlaer and Mellor 1992, Embley 1992, Coleman et al. 1994, Yourdon 1994, Graham 1994, Martin and Odell 1995, Henderson-Sellers and Edwards 1995, UML 1997). These models are not always known as finite state diagrams; other names include state transition diagram, state net, object life history and state machine. The Petri-net is one kind of extension to the finite state diagram.

The finite state diagram is a uni-partite directed graph, ie it contains nodes of a single type representing states or properties of a system. Movement from one state to another (ie a transition) may take place if an arc exists between those two states.

There are essentially two types of finite state diagram. These are the Mealy and the Moore 'machines' (Martin and Odell 1995). In the Mealy machine the operation is associated with the transition. Booch (1991) originally employed the Mealy model though later he rejected it in favour of a combined Mealy/Moore approach (Booch 1994). In the Moore machine the operation is associated with the state. Shlaer and Mellor (1992) use the Moore model. Some authors use a combination of Mealy and Moore models in which both transitions and states can have operations associated with them (Rumbaugh et al. 1991, Embley 1992, Booch 1994).

Peterson (1977) provides a good survey of the use and application of the Petri-net. A Petri-net is a bi-partite directed graph. This means that two types of nodes are present. One node represents transitions and the other represents places. Places can be used to represent states though not necessarily. For example, Lausen (1988) uses places to represent milestones such as 'judgement' and 'external order' in addition to what intuitively might be seen as states. However, in the BRD a judgement is likely to be an event and an external order a trigger. This example suggests there is some confusion or at least disagreement about the semantics of terms such as states, events and triggers. Because the arcs are connected between transitions and places, greater expressive power is possible in the Petri-net than in the finite state diagram. In essence this is the advantage of a bi-partite graph over a uni-partite graph. So in a Petri-net many places may be predecessors of a transition and one transition may be the predecessor of many places. Some authors annotate the Petri-net (or accompany it) with a pseudo-mathematical notation (Herbst 1996).

An aspect which applies to both Petri-nets and finite state diagrams is their asynchronous nature. In other words, there is no intrinsic measurement of the passing of time for example in terms of micro-seconds or days. '*This reflects a philosophy of time which states that the only important property of time, from a logical point of view, is in defining a partial ordering of the occurrence of events. Events take variable amounts of time in real life; the Petri-net model reflects this variability by not depending upon a notion of time to control the sequence of events*' (Peterson 1977, p229). This asynchronous notion of time has been implemented in the BRD by triggers.

In this section three types of finite state diagrams and one type of Petri-net are reviewed. The finite state diagrams are taken from Booch (1991), Shlaer and Mellor (1992) and Rumbaugh et al. (1991). They have been chosen because their characteristics allow the highlighting of major differences between extended state-based models and the BRD **in general**.

4.2.1 Booch

Booch (1991) uses an essentially straightforward state transition diagram notation based on the Mealy machine (ie operations on the transitions or arcs). Booch's operation can be argued to be equivalent to the BRD's event. The transitions may also be annotated by the 'event type/trigger rule'. This amounts to a trigger in BRD terms. So, in BRD terminology, states are shown graphically and events and triggers may be inferred from annotations.

4.2.2 Shlaer and Mellor

Shlaer and Mellor (1992) use a state transition diagram based on the Moore machine which is to say that operations are ascribed to the 'state' itself. In practice this means that states are often processes. For instance, in an example of an account object, states of 'taking deposit', 'creating account' and 'considering withdrawal' are identified (Shlaer and Mellor 1992, p71). There is considerable annotation on these diagrams including conditions and also sometimes BRD states can be inferred from the transition labels.

4.2.3 Rumbaugh et al.

Rumbaugh et al. (1991) have a more comprehensive approach to describing their state diagram which involves a combination of Mealy and Moore machines (Martin and Odell 1995). Essentially this involves depicting states graphically and annotating triggers, conditions and events on the transitions.

4.2.4 Petri-net+

Peterson (1977) discusses the history and use of the Petri-net in some detail. Included in this discussion is something of the variety of purpose to which different versions of the Petri-net are put. For the purpose of this review, it is assumed that a basic Petri-net notation is used with additional pseudo-mathematical notation which describes triggers and conditions. In BRD terms this means that events and states are shown graphically, but triggers and conditions appear non-graphically.

4.2.5 Evaluation of Extended State-based Models

Table 4.1 is an attempt to highlight the major differences and similarities between the constructs of the BRD and the techniques just discussed. The columns in the table represent each major construct of the BRD. In addition there is a further column for static data, the reason for which will become obvious in a later section. The rows indicate whether or not a particular feature occurs in that technique. Because sometimes features are graphical and on other occasions not, 'Yes' indicates that a feature occurs graphically and 'yes' indicates that a feature occurs but not graphically eg a feature may be part of some pseudo-mathematical notation.

	Triggers	**Conditions**	**Events**	**States**	**Messages**	**Static data**
BRD	Yes	Yes	Yes	Yes	Yes	No
Booch	yes	No	yes	Yes	No	No
Shlaer & Mellor	No	yes	Yes	yes	No	No
Rumbaugh et al.	yes	yes	yes	Yes	No	No
Petri-net+	yes	yes	Yes	Yes	No	No

Table 4.1 Feature Analysis of BRD and Extended State-based Models

Three major differentiating features of the BRD are now outlined. These features are the expressive power of the BRD, modelling significant states and capturing inter-object dependency.

There is problem with the expressive power of the finite state diagram. The problem is one of showing too much information on a diagram which does not have a rich enough set of graphical constructs (see chapter 2). For example, on Shlaer and Mellor's state diagram, both the operation and what they call the method which tests the history of the state are associated against the state. Similarly, the transition displays the event type/trigger rule. On the Booch's Mealy model the event type/trigger rule and operation are both depicted on the transition. A notation with a richer set of constructs would reduce ambiguity and so reveal its information in a more explicit fashion.

Both the BRD and the Petri-net have a richer set of graphical constructs. As discussed in chapter 2, the Petri-net with its bi-partite properties is superior to that of the finite state diagram when the intention or need is such that more expressive power is required. The BRD is distinguished from the Petri-net in that both the trigger and condition are shown graphically. So the BRD has at least as much expressive power as the Petri-net and arguably its graphical depiction of trigger and condition make it superior to the Petri-net for describing business rules.

The second differentiating feature is how states are used and defined within the BRD. The term **significant state** refers to states of an information system which determine how subsequent triggers to the information system will be processed. In other words a state change implies that the information system itself has changed in some way and that thereafter it may respond differently to external stimuli. A constraint imposed on the object life history and the BRD itself is that all states must be significant states. This constraint is not evident or recognised in the literature, though I believe it is crucial in modelling business rules. It is crucial because significant states ground the model construction in a manner which determines how a business rule's associated conditions, triggers and messages are clustered and presented and thus a more stable, tractable and easily verifiable model is created. Examples of this are provided later in the thesis.

The third major differentiating feature is to do with inter-object dependency. I can find no evidence in the literature in which state-based models other than the BRD show dependency between objects. Typically state-based models are elaborations of a single object. The resultant object life history (or whatever term is used) shows the milestones and dependencies within that object's life but not whether other objects have to be in certain states for a state transition to occur. Some methodologies do recognise this problem and attempt to deal with it by constructing object communication diagrams (Shlaer and Mellor 1992) but because of the level of abstraction involved (ie that of the object and not the state) the detailed inter-object state dependencies cannot be defined. One obvious argument against depicting inter-object dependencies on a single diagram is the explosion in size that inevitably occurs. In this respect, the decision to limit the BRD to policy rules goes some way to reducing this problem. As far as the literature is concerned, there is no evidence of partitioning of business rules into policy and other kinds of rules.

4.3 Condition-based models

All models in this section contain conditions at their core, though they also contain other constructs. In fact each of the models reviewed contain the constructs of trigger, condition and event. Thus the question arises as to why the condition has been chosen as the descriptor for these models. Clearly, such a decision is a subjective one and is based on my evaluation of the model concerned. Interestingly one of these models (CPM) has the same evaluated characteristics as an event-based model (Martin and Odell 1995), yet it was classified differently due to what I considered as its essential character.

4.3.1 Herbst

Herbst (1996) presents a meta-model and repository system for business rules. The rules themselves follow the ECA or ECAA structure (based on the work of Tsaligatidou et al. 1990) in which E represents an event, C a condition and A an action. ECA reflects a simple case in which there is only one action, whereas ECAA is required in a 'then-else' situation. Herbst's events are equivalent to triggers in the BRD and Herbst's actions are either events or messages in the BRD. The word condition means the same in both techniques. Herbst does not present these components of a business rule graphically however; rather they are expressed in a pseudo-mathematical notation. However, he does model a complete business rule as a single graphical construct (ie as one symbol) in a diagram interleaved with triggers. In addition he provides a graphical means of decomposing business rules into smaller business rules. This aspect contradicts the philosophy of a significant state. Herbst also refers to the possible use of high level Petri-nets to graphically model aspects of business rules. In these Petri-nets the places are events however and not states as in the BRD.

4.3.2 Tsalgatidou and Loucopoulos

Tsalgatidou and Loucopoulos (1991a, 1991b) present business rules using a structured English form again based on the ECA structure. If a condition fails however, another business rule can pick up this failed condition and in that way the ECAA structure is simulated. They additionally have a graphical notation based on the Petri-net in which the transitions represent complete business rules and the places represent in their words the 'triggering condition'. Many, though not all of their triggering conditions could well be states such as 'order on hold' in their example, but others would not be states eg 'before create order-line'. I consider their identification of states here coincidental.

4.3.3 Sandy

Sandy (1996) presents a graphical notation again based on the ECAA structure. More graphical than the others above, he also introduces the concept of a compound rule in which many conditions are evaluated before the rule fires. He also provides a mechanism for decomposing rules into smaller units.

4.3.4 CPM

The conceptual processing model (CPM) of Merise (Quang 1991) can be argued to provide all necessary components of the ECA structure. However, there are only two graphical constructs used. The ellipse may be a trigger, message or even coincidentally a state. The rectangle is a condition.

4.3.5 Evaluation of Condition-based Models

Table 4.2 highlights the major differences between the condition-based models and the BRD. The same conventions and approach have been used to its display as in the previous table.

	Triggers	**Conditions**	**Events**	**States**	**Messages**	**Static data**
BRD	Yes	Yes	Yes	Yes	Yes	No
Herbst	yes	yes	yes	No	yes	No
Tsalgatidou & Loucopoulos	yes	yes	yes	Coinc-idental	No	No
Sandy	Yes	Yes	Yes	No	No	No
CPM	Yes	Yes	Yes	Coinc-idental	Yes	No

Table 4.2 Feature Analysis of BRD and Condition-based Models.

The major difference between the BRD and the condition-based models is that condition-based models effectively do not incorporate a state construct. From the arguments in the previous section relating to significant states, it follows that models which do not depict states suffer from a lack of tractability. This is a major weakness because with such models there is no consistent way of organising and structuring the presentation of rules. In practice two analysts working on the same problem might construct two very different presentations of business rules, although each may be quite correct in capturing the inherent logic of the problem. In other words, these rules or aspects of them are likely to be packaged in different ways. Although Herbst's and Sandy's approach of functional decomposition is quite understandable as a means of partitioning a full set of rules, this exacerbates the problem of inconsistency of presentation further. This is not the case with the BRD, since the state change constraint forces analysts to package business rules round the state change and therefore consistently.

Sandy's more graphical approach is superior to that of Herbst and Tsalgatidou and Loucopoulos. Sandy has isolated the graphical constructs of trigger, condition and event, I believe, in an attempt to underline the importance of depicting them visually. The same argument applies in the BRD where four main constructs are depicted graphically to force attention on these aspects of a business rule.

Upon inspection of the table it may be felt that CPM is quite similar to the BRD since it seems to support triggers, conditions events and messages. However, CPM uses only two constructs to describe what are four constructs in the BRD. This is ambiguous and inferior to the BRD which explicitly distinguishes these constructs.

In summary, the main distinction between condition-based models and the BRD is in the absence of the state construct. This is a major differentiating factor. Also the limited use of graphical constructs in some models leads to ambiguity and further serves to separate the BRD from the others.

4.4 Extended Static Models

In this section, models which are created first by constructing a data or object model with inter-object relationships and then extending this by adding dynamic features are discussed. As discussed anecdotally in chapter 2, static models can capture some types of business rule. By extending static models it seems reasonable that more types of business rule could be captured.

4.4.1 Kappel and Schrefl

Kappel and Schrefl (1989) and earlier Eder et al. (1987) introduce the BIER or Behavior Integrated Entity-Relationship approach. Essentially, this is a conceptual object model to which dynamic constructs based on Petri-nets have been added. The model produced is therefore a very rich one, since both static relationships between objects as well as dynamic relationships can be inferred from the diagram. A decomposition mechanism is also provided.

4.4.2 Kung and Solvberg

Kung and Solvberg (1986) present a conceptual model which has some similarity with the BIER model in the sense that it incorporates both static data and dynamic information. Kung and Solvberg's model depicts static data such as entities and data attributes and has the ability to show relationships to subsets. Their examples only cover simple objects so it is not clear how or if the model deals with complex objects. Petri-net constructs are also used, with places connected to subsets. Each transition is augmented with a pseudo-mathematical notation detailing any pre- and post-conditions of the transition. Conceptually subsets perform a similar purpose to states. For instance, in their example of an order they identify the subsets of new orders, rejected orders, back orders and shipped orders. Post-conditions define the transfer of membership from one subset to another and in this way subsets with post-conditions effectively perform the same function as states with connecting arcs.

4.4.3 Evaluation of Extended Static Models

Table 4.3 summarises the main differences between the BRD and the extended static models. Again the same conventions and approach have been used to its display as in the previous tables.

	Triggers	**Conditions**	**Events**	**States**	**Messages**	**Static data**
BRD	Yes	Yes	Yes	Yes	Yes	No
Kappel & Schrefl	No	No	Yes	Yes	No	Yes
Kung & Solvberg	No	yes	Yes	Sub-sets	No	Yes

Table 4.3 Feature Analysis of BRD and Extended Static Models

The main differentiating factor with these models compared to the BRD is the amount of information that is being displayed on one diagram. The BRD does not contain static information (eg attributes and relationships between objects) which thus reduces the informational content significantly. An important issue in this thesis is concerned with the perceived complexity of a diagram. The major concern is the problem of information overloading. With so much information on one model, questions arise in relation to the ability of analysts as well as users to absorb the information contained when dealing with medium or large information systems (Winograd and Flores 1987).

With regard to the BIER model (Kappel and Schrefl), it does not show signals, ie both triggers and messages, so there is some distinction between the BIER model and the BRD. However, the BIER model does attempt to model states through the places of the Petri-net and also activities which have some similarity with the events of the BRD.

Kung and Solvberg's model is powerful and rich. Although it covers many aspects that the BRD does not, it does not depict signals, ie triggers and messages. By its use of post-conditions the detail inside a process is effectively declared, although some of this is dealt with in the BRD with the event specification table. On the other hand, by information hiding the BRD deliberately seeks to avoid that type of specification by indicating only the whole event without any detail. Like the BIER model, there is a concern of information overloading with non-trivial systems.

To summarise, although deficient in some constructs of the BRD, extended static models in the literature provide additional information not shown on the BRD. There is a concern over information overloading in extended static models.

4.5 Trigger-based Models

In this section, diagrams which in essence model triggers are discussed.

4.5.1 Joosten

Joosten (1994) uses a trigger model for workflow analysis. The principle focus of the diagram is on the triggers which initiate events. Hence these two constructs (ie the trigger and the event) are depicted graphically. There is some discussion in his paper as to how triggers can be mapped onto Petri-nets, but Petri-nets are not part of the model itself. The trigger model could be used as a pre-cursor to modelling a human activity system with Petri-nets.

4.5.2 Flynn and Davarpanah Jazi

Flynn and Davarpanah Jazi (1998) present an Event Flow Diagramming technique which is somewhat richer than Joosten's. In fact, three related diagrams are involved, but for the purposes of this comparison only the Event Flow Diagram (EFD) will be considered. The EFD contains the constructs of event, process, object, department, external body and flow. However, in the terminology of the BRD, the event in the EFD is essentially the same as a trigger, hence its categorisation here. The process in the EFD may be an event in the BRD; however, events in the EFD may also constitute only part of an event in BRD terms. There is no concept of state in the EFD. Messages are shown in the EFD through reference to a table. Static data is declared through the object construct; however, this is only defined at a high level of abstraction, ie no detail of an object is defined. Conditions are generally not shown although flow of information is shown through and AND/OR type logic, so some degree of control flow can be detected.

4.5.3 Evaluation of Trigger-based Models

Table 4.4 compares the differences between the BRD and the Joosten model.

	Triggers	**Conditions**	**Events**	**States**	**Messages**	**Static data**
BRD	Yes	Yes	Yes	Yes	Yes	No
Joosten	Yes	No	Yes	No	No	No
Flynn & Davarpanah Jazi	Yes	Limited	Yes	No	Yes	Limited

Table 4.4 Feature Analysis of BRD and Trigger-based Models

It can be seen from the table that the main area of commonality between these trigger-based models and the BRD is in depicting triggers and events although events in these trigger-based models are not anchored around a significant state, ie because states are not present in these trigger-based models, the earlier arguments regarding this omission apply to these trigger-based models also. Further, the lack of a full description of conditions is also a weakness in terms of modelling of business rules.

4.6 Event-based Models

In this section, diagrams in which the concept of the event is at the heart of the model are discussed.

4.6.1 Dataflow diagram

The dataflow diagram has been popular with practitioners since the late seventies (De Marco 1978). Though termed 'dataflow' diagram the concept of input-process-output is used as a means of modelling human activity systems. External triggers are captured as incoming dataflows but time triggers are deliberately not shown. Messages are indicated through outgoing dataflows. Conditions are deliberately hidden on these diagrams. The datastore construct depicts static data but at a high level of abstraction eg relationships between data are not shown.

4.6.2 Martin and Odell

In Martin and Odell's event diagram (Martin and Odell 1995), as the name implies, the event is considered the main focus of the diagram although triggers and conditions are also captured. Like some other models mentioned previously, states are sometimes made explicit though this is coincidental, ie there is no explicit attempt to discern significant states. Note also that since an event in the BRD is defined as something which changes states of objects, the events in the event diagram are different to those in the BRD.

4.6.3 Evaluation of Event-based Models

Table 4.5 summarises the main differences between the BRD and the event-based models.

	Triggers	**Conditions**	**Events**	**States**	**Messages**	**Static data**
BRD	Yes	Yes	Yes	Yes	Yes	No
Dataflow diagram	External	No	Yes	No	Yes	Limited
Martin & Odell	Yes	Yes	Yes	Coinci-dental	No	No

Table 4.5 Feature Analysis of BRD and Event-based models

Once again, perhaps the most revealing difference between the BRD and the event-based models is the lack of the state construct and the implications that has for the modelling of a business rule. Because the event in both these event-based models is not necessarily anchored around a state change, these diagrams in my view suffer from the same problems that condition-based models do.

4.7 Summary

The task of comparing the BRD with other models is not a trivial one. This is partly due to a lack of standardisation of terminology in the area and partly because of difficulties in clarifying the role of a particular technique within its methodology and with other diagrams. At times it was seen that that the BRD provided a richer set of constructs than other models; on other occasions other models seemed to offer more constructs raising the concern of information overload. A pragmatic approach was taken to discussing and classifying the literature on business rules. This involved essentially a feature analysis of each main construct of the BRD and searching the literature for competitors. The focus of discussion and evaluation in this chapter was on aspects which differentiated the BRD from its competitors.

Table 4.6 is a summary table which combines into a single table the analyses of the earlier tables.

	Triggers	**Conditions**	**Events**	**States**	**Messages**	**Static data**
BRD	Yes	Yes	Yes	Yes	Yes	No
Booch	yes	No	yes	Yes	No	No
Shlaer & Mellor	No	yes	Yes	yes	No	No
Rumbaugh et al.	yes	yes	yes	Yes	No	No
Petri-net+	yes	yes	Yes	Yes	No	No
Herbst	yes	yes	yes	No	yes	No
Tsalgatidou & Loucopoulos	yes	yes	yes	Coinc-idental	No	No
Sandy	Yes	Yes	Yes	No	No	No
CPM	Yes	Yes	Yes	Coinc-idental	Yes	No
Kappel & Schrefl	No	No	Yes	Yes	No	Yes
Kung & Solvberg	No	yes	Yes	Sub-sets	No	Yes
Joosten	Yes	No	Yes	No	No	No
Flynn & Davarpanah Jazi	Yes	Limited	Yes	No	Yes	Lim-ited
Dataflow diagram	External	No	Yes	No	Yes	Lim-ited
Martin & Odell	Yes	Yes	Yes	Coinci-dental	No	No

Table 4.6 Summary Table containing a Feature Analysis of the BRD against other Models in the Literature.

In summary, the main arguments used for differentiating the BRD from other models in the literature were:

- problems with the lack of a state construct in other models,
- other models not using states to anchor the level of abstraction,
- problems with the lack of inter-object dependency in other models,
- problems with the expressive power of finite state diagrams,

- that graphical constructs are superior to non-graphical constructs,
- problems with ambiguous graphical constructs in other models,
- problems with information overloading in other models and
- problems with the semantics of constructs in other models.

Chapter 5

Research Methodology

5.1 Overview

The purpose of this chapter is to explain the rationale for choosing action research as the prime research method. The decision involved determining which approach was the 'best fit' for the research question. Figure 5.1 is a cognitive map which summarises the steps in firstly selecting action research as the most appropriate approach for this research (research methodology) and then designing the detailed structure of the research (research design).

The decision to select action research took some time and frankly was not as organised or as linear as the cognitive map implies. Necessarily there was a need to have a reasonably deep understanding of issues such as interpretivism and positivism and with how such issues affect this research. Blind alleys were explored; alternative approaches were considered. However, over time a gradual firming up took place of the conclusion that action research was the most appropriate approach for this type of research question.

This chapter will describe the steps involved in selecting the research approach, ie those steps shaded in grey in figure 5.1. Note that at some of the steps it is possible that the outcome of deliberation may cause returning to an earlier step in the selection process in light of the (un)acceptability of an outcome.

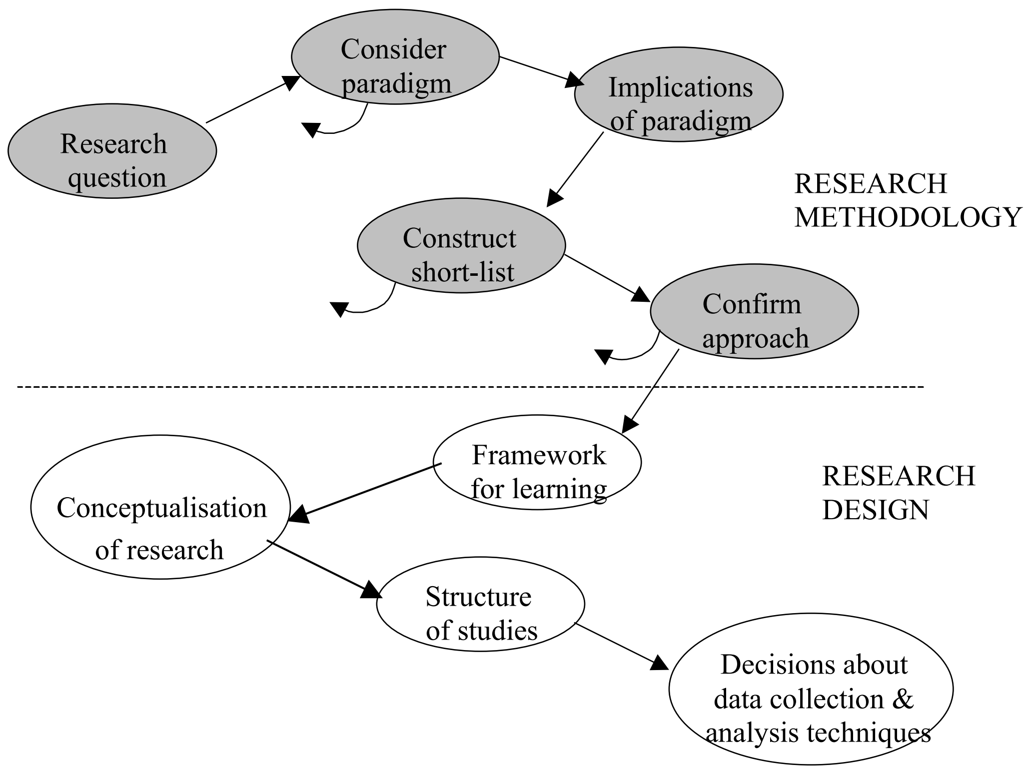


Figure 5.1 Cognitive Map indicating Steps in Research Methodology and Design

5.2 Consider Research Paradigm in Relation to Research Question

Deciding on a research paradigm is not a simple question. Even a cursory view of the literature reveals a major dichotomy between **positivism** on the one hand and **interpretivism** on the other. This dichotomy has significant implications for anyone researching information systems. The problem is that these two views or philosophies reflect very different views about the nature of knowledge and how we perceive the world. Consequently, the tactics and techniques one chooses to investigate research questions are determined by the perspective one holds. It also follows that the 'validity' of the research as viewed by others hinges on the perspective they hold.

I consider it self-evident from the research question that the introduction of the Business Rules Diagram (BRD) while conducting an information requirements determination (IRD) of a human activity system intervenes into and therefore influences the social environment of an organisation. Checkland in quoting Vickers describes this principle succinctly when he referred to the work of Copernicus and Marx (Checkland 1991). In acknowledging the work of Copernicus, Vickers noted that no matter what Copernicus deduced about the laws that govern the heavens, such conclusions would not affect the how the heavens actually operate. By contrast, the social theories developed by Marx shaped social history and continue to influence events today. The difference is that Marx and his theories were actually part of the system of social history and so influenced it. As regards introducing the BRD into an organisation, the technique would be used by analysts and users. Necessarily, it would change their work practices and therefore the way that these people make sense of their world. Indeed, the analysts and users will determine whether and exactly how the BRD will be used within their own organisations. A related observation is made by Klein and Hirschheim with respect to data modelling when the noted that '*no data modelling can avoid philosophical assumptions, because data modelling is a process of inquiry that has intrinsic similarities with classic scientific construction'* and later they remark '*Integrity of data means consistency with (possibly varying) social uses and not with an objectively given state of affairs. This provides the philosophical base of the subjectivist paradigm of data modelling in that different approaches to data modelling will lead to different pictures of reality*'. (Klein and Hirschheim 1987, p10). So it is important that this research should focus on the social uses of the technique as well as its technical efficacy.

This hermeneutic principle of being inside or outside the system is also central when considering the research paradigm. Orlikowski and Baroudi provide a good definition of the positivist perspective in '*Understanding phenomena is thus a problem of modeling and measurement, of constructing an appropriate set of constructs and an accurate set of instruments to capture the essence of the phenomenon. It is assumed, explicitly or implicitly, that there is a one-to-one correspondence between the constructs of a researcher's model and the events, objects, or features of interest in the world. The researcher herself is seen to play a passive neutral role in this investigation, and does not intervene in the phenomena of interest*.' Orlikowski and Baroudi (1991, p9). The positivist perspective has been criticised because it limits the kind of research that is considered acceptable. For example, the weaknesses of a purely positivistic outlook are recapitulated in Klein and Lyytinen (1985, p139) where they state '*The danger of scientism when applied* *to the resolution of practical problems lies in a narrowing of problem perception to those aspects which are "researchable" by scientistic methods....This in fact biases the whole definition of information systems towards construing them as deterministic artefacts which must be "scientifically" engineered*'. And Boland (1985, p194) comments '*Positive science explains reality as the law-governed outcome of causally related, antecedent variables. It starts with theory and proceeds through empirical tests to disconfirm a theory. A theory identifies variables and their causal relationships in order to make predictions.'*

This is in sharp contrast to an interpretivist view in *'While it has been argued that there exists an independence between researcher and phenomenon of study in the natural sciences, the same assertion cannot be made for the social sciences. While the results of natural science do not impinge on and change the nature of the phenomena studied, the results of social science do enter into the discourse of everyday human reality, and clearly can and do transform the nature of these phenomena'* Orlikowski and Baroudi (1991, p11).

From the above discussion and quotations, it is clear that an interpretivistic approach is highly desirable, otherwise valuable insight and understanding may well be overlooked.

5.3 Implications of Research Question and Interpretivistic Paradigm

There are a number of implications which follow on from the research question and the decision to adopt an interpretivist perspective in undertaking this research. These implications have a bearing on the specific type of research approach that may be selected and so are summarised here before examining available options.

5.3.1 Real-world Context

As stated in the introductory chapter, techniques which are merely asserted as useful under Galliers' 'subjective/argumentative' heading from a technical or software engineering perspective cannot be regarded as useful until they have been tested more rigorously. It is considered highly valuable to ground the research as far as possible in the 'real-world' of the analysts' and users' organisation. Of course there are difficulties in using an organisational situation, and these must be overcome or at least managed, but to avoid working in a ‘real-world’ context because of such difficulties is ‘throwing the baby out with the bath water’.

5.3.2 Social and Technical Domains of Interest

There is a need to focus on the social uses of the BRD in the 'real-world' of the organisation. The BRD technique, like data modelling, is a process of inquiry and as Klein and Hirschheim (1987) suggest, such processes open up myriad possible ways in which a technique might be incorporated into a social environment. Both systems analysts and users will work with the BRD and therefore how they relate to it in their working environment will be a significant factor in determining whether it will be considered successful. From the perspective of selecting a research approach, it is therefore imperative that an approach enables the researcher to identify and examine social as well as technical aspects.

5.3.3 Iterative Learning

The 'real-world' of organisations can be viewed as a highly complex system involving many different types of components. The question arises as to how a researcher might seek to learn or explain behaviour amidst that complexity. At a practical level, the research process could involve taking working versions of the diagram to an experienced analyst, using that version on real problems, obtaining feedback, analysing that feedback and then creating a revised version in a spiral fashion. I consider this iterative learning process ideal for designing new diagrammatic techniques because it provides an opportunity to test ideas, reflect on them, modify them and then retest them. In other words it reflects an evolutionary view of systems and of how systems change consistent with that of Checkland (1981). An ideal research approach here would accommodate iterative learning.

5.3.4 Explicit Learning

The purpose of this research is to develop a diagram for organisational use. As part of that task it will have to **evaluate the impact** of the introduction of a diagrammatic technique into an organisation. Because the research has pre-defined goals, it carries with it some assumptions about the value of the technique to an organisation. It is therefore important to be explicit about any assumptions that are being made and about what the researcher's mindset, ideas and philosophies are with regard to the research. Only then can a complete explanation of why events and decisions taken during the research process be considered to have been provided. As Boland (1985, p194) puts it when referring to phenomenology: *'Phenomenology, in contrast, questions the value of a science that does not first make it clear what it is speaking about. We only find out what things are through a methodical process of description*'. This implication will affect the way that the research programme is constructed, ie steps must be included to inform the reader of such matters and later deliberate them in the context of what has been learnt.

5.3.5 The Degree of Involvement of Participants

An interpretivistic style has implications for the role that the researcher might play. In particular, the focus on social issues and social uses of the technique implies that the researcher is no longer the expert in terms of how the technique should be embedded into the participant's environment. The degree of involvement of the participants (ie analysts and users) should reflect this shift in expertise. So, an important aspect in choosing an appropriate research approach would be to find an approach which would have a high degree of participation. Participants may be regarded as co-researchers here.

5.4 Create Short-list of Research Approaches

Galliers (1990) identifies a range of research approaches considered suitable for research in information systems. His work builds on earlier attempts at classifying information system research approaches (Van Horn 1973, Hamilton and Ives 1982, Vogel and Wetherbe 1984, Galliers and Land 1987 and Farhoomand 1987). For consistency's sake the Galliers taxonomy is used here. The major categories are laboratory experiment, field experiment, survey, case study, theorem proof, forecasting and futures research, simulation or game/role playing, subjective/argumentative research, action research and descriptive/interpretative research. In terms of the discussion in the previous section, a number of research approaches can be removed as inappropriate without further discussion. These are survey, theorem proof, forecasting and futures research, simulation or game/role playing and subjective/argumentative reasoning. For example, simulation and game/role playing do not enter into the real-world (section 5.3.1) and subjective/argumentative reasoning does not involve any kind of rigorous testing involving participants (5.3.5).

The remainder will now be reviewed in terms of their appropriateness to the research question, ie as a research vehicle to assist in developing the BRD.

5.4.1 Laboratory Experiment/Study

Laboratory experiments, as the name implies, involve setting up and conducting experiments in a laboratory environment. One example in the context of this research this could be an experiment in a classroom using students as surrogates for analysts or users. The main weakness of laboratory experiments is their artificiality, ie the lack of 'real-world' context which is central to the research question. Laboratory experimentation ideally implies the use of control groups which for pragmatic reasons may be difficult to set up. For example, if groups of students are used, it is often difficult to engineer proper control groups into a university timetable while at the same time ensuring there is no transfer of ideas between groups. Where there is less emphasis on quantitative aspects such as measurement of dependent and independent variables I prefer the term laboratory study or controlled study.

5.4.2 Field Experiment/Study

Galliers (1990, p161) states that '*field experiments are an extension of laboratory experiments into the 'real' world of organisations/society. The idea here is, of course, to attempt to construct an experiment in a more realistic environment.'* Field experiments also can be difficult to engineer since the researcher has to rely on the participating organisation to provide access to staff and other resources. Further, questions also arise in relation to what, if anything, the organisation hopes to gain from the activity. At one extreme, the organisation may have little or no interest in the research but has acquiesced to the request of the researcher. On the other hand, the organisation may be experiencing acute problems in an area and feels that the field experiment may be able to shed some light on the problem. Of course, at either extreme political and motivational issues arise and therefore ought to be borne in mind while undertaking the research. However, this kind of research does meet the need of a real world-setting and is therefore a candidate approach.

However, perhaps the terms field study or field trial are more acceptable where the analysis of the activity is more qualitative than quantitative. The field study therefore also extends into the real-world of the organisation, but does not require quantitative analysis or the pre-definition of hypotheses. Use of a field study would be consistent with an interpretivistic line of inquiry and thus a field study would be preferable to a field experiment given the research question and its social implications previously discussed (sections 5.3.2, 5.3.5).

5.4.3 Case Study

A case study involves observation of an existing phenomenon in a relatively unobtrusive way. A case study can be of great help in explaining why something has happened. In the context of this research it could possibly be used to show that existing IRD techniques are not particularly good at defining business rules, but in itself it would not be appropriate to assist in designing and testing a new technique. In other words, a more active research method, for example, involving iterative learning (section 5.3.3) to test out ideas or modifications, is seen as desirable here to meet the implications of the research question.

5.4.4 Action Research

Action research involves the researcher intervening into the problem situation and in particular working with participants in a co-learning environment to achieve common goals. It is set in a real-world context and problem situation and therefore is a candidate approach. Its weaknesses stem from its subjective nature and from the difficulty of generalising from a specific situation. Much more is said about action research shortly.

5.4.5 Descriptive/Interpretive Research

This form of research involves the researcher immersing himself or herself in the phenomenon being investigated. It can be done in organisational settings and often involves longitudinal study. It is considered a highly subjective form of research though its proponents consider it often highly effective in establishing a rich understanding notwithstanding its potential for bias. However, it is mostly used to understand existing phenomena as opposed to testing new phenomena such as the research question here where a new notation is being devised.

5.4.6 Short-list

In previous sub-sections a number of research approaches were considered as candidates for this particular research question. Laboratory experiment/study has been deleted from the list of potential approaches because it lacks real-world context and to a large degree it would be difficult to focus on social issues in a laboratory. However, the use of laboratory experiment/study should not be excluded as a component of a hybrid study. Case study and descriptive/interpretive research are now removed as candidates because their strength lies in analysing existing phenomena rather than predicting the impact of a new phenomenon. This leaves field study and action research as two strong candidates.

5.5 Confirmation of Research Approach

Arguably, there is some overlap between the terms field study and action research in the sense that an action research study could be described as a field study, because it involves testing a question or working on a goal in a real-world setting. However, a strong differentiating feature of action research, based on generally accepted definitions of action research and the characteristics usually associated with it, is the notion that participants are collaborative co-researchers in the research process. In other words, if it is intended in the research to use participants **during** the research process to solicit feedback and use that feedback subsequently, action research would be regarded as a much better descriptor. Table 5.1 is a review of some attempts at classifying the characteristics of action research. Note how often terms such as collaboration, co-learning, participation and so on occur in the table.

With the field study, the researcher still remains the expert and the participants are components of the experiment, which reduces the opportunities for collaboration, participation and co-researching considerably. In the section 5.3.5 it was argued that the participant was the expert in terms of determining the social uses to which a technique may be put and so it is concluded that action research is the chosen research approach. Hult and Lennung's widely accepted definition (1980) captures the essence of this research in '*Action research simultaneously assists in practical problem solving and expands scientific knowledge, as well as enhances the competencies of the respective actors, being performed collaboratively in an immediate situation using data feedback in a cyclical process aimed at an increased understanding of a given social situation, primarily applicable for the understanding of change processes in social systems and undertaken within a mutually acceptable framework'*. Note that among a number of important features of action research in this definition (such as data feedback, cyclical process, increased understanding and change processes), there is emphasis on social phenomena in a collaborative context. This is a good fit with respect to the style and direction of the intended research.

Author	Characteristics
Dick (1992-3)	cyclic, qualitative, **participative**
Burns (1990)	situational, **collaborative, participatory**, self evaluative
Ketterer, Price and Polister (1980)	problem focus, **collaboration**, development of scientific and practical knowledge, research utilisation
Elden and Chisholm (1993)	purposes and value choice, contextual focus, change based data and sense making, **participation in the research process**, knowledge diffusion
Greenwood, Whyte and Harkavy (1993)	**collaboration**, incorporation of local knowledge, eclecticism and diversity, case orientation, emergent process
Israel, Schurman and Hugentobler (1992)	**participatory, cooperative, co-learning process**, involves systems development, empowering process, balance between research and action
McKernan (1991)	increases human understanding, concern to improve quality of human action and practice, focus on problems of immediate concern to practitioners, **collaborative**, conducted in-situ, **participatory**, focus on the case or single unit, no attempt to control setting variables, the problem, aims and methodology may shift as inquiry proceeds, evaluative-reflective, methodologically eclectic-innovative, scientific, sharable, dialogue/discourse-based, critical, emancipatory

Table 5.1 A Review of Action Research Characteristics from the Literature

Baskerville and Wood-Harper maintain that many of the problems of action research are problems generally in social science research. They state that '*These problems are actually general problems of social science research. In reality action research shares these problems with other methods. Perhaps the distinguishing difficulties with action research are those of degree, rather than taxonomy*.' (Baskerville and Wood-Harper 1992, p10). Such problems or weaknesses include:

- the inherent subjectivity of the approach,
- the inability of the researcher to remain unbiased,
- the perceived lack of discipline in action research and
- the inability of the researcher to exclude alternative explanations, particularly cause-effect relationships.

(Baskerville and Wood-Harper 1992, Shanks et al. 1993)

So, what in practice can be done to address these weaknesses? The answer lies in the construction of a sound programme of action research which makes explicit what it is attempting and clearly demonstrates how it has achieved its outcomes. Particularly important here is the notion of explicit learning mentioned earlier. By setting out the predispositions and Weltanschuungen of the researcher and indeed other participants, that creates a scorecard of 'prejudices' against which subsequent decisions and behaviour can be evaluated. In that way the reader can assess the validity of the aforementioned weaknesses of subjectivity, bias and lack of discipline. Further, by explaining clearly the rationale for decisions, the researcher presents a case for why a particular explanation is valid. Again, the reader is left with the information to evaluate the research against the fourth weakness above concerning inability to exclude alternative explanations.

Of course it is also important to construct a sound research programme. Swepson and Dick (1993) make a number of proposals. These include:

- the use, wherever possible, of multiple sources of data (different informants, different samples, different research settings, different researchers),
- testing in the later cycles the tentative interpretations emerging from the earlier cycles and
- seeking out exceptions to whatever generalisations have so far emerged and when exceptions are found seek out explanations.

As far as possible the above should be built into the research design strategy as the following discussion tries to illustrate.

Multiple sources of data

This is demonstrated more fully in later chapters, but for example, where possible different application problems should be used to explore the BRD in combination with different organisations and different people within these organisations. Any conclusions drawn from the research are strengthened by the fact that different sources of data (presumably) confirm the same phenomena.

Testing in later cycles

This is another aspect of design that should be 'built in' to the programme and relies on the iterative learning concept discussed earlier. As an example of how this concept might be implemented, suppose a conclusion is reached at one stage with one group of participants in one organisational setting on one application problem. If that conclusion is re-confirmed with other participants, in another setting or on another application then the strength of that conclusion is clearly enhanced to some degree.

Seeking out exceptions and seeking explanations

This strategy uses the ideas of iterative learning and explicit learning mentioned earlier. It may be carried out by the researcher while the research is taking place and also when the research is being analysed. For example if a semi-structured interview is being used within an action research framework then such an interview enables exceptions to be explored and possible explanations identified. Further, these explanations can then be verified by exploring them in later phases of the research.

The way that action research is reported also has a role to play in mitigating its perceived weaknesses. For example, Walsham (1993, p15) argues for the '*plausibility and cogency of the logical reasoning used in describing the results from the cases, and in drawing conclusions from them'*. In other words, just as in quantitative research an evaluation is made based on the quality of for example the statistical methods deployed. Thus, with action research and other qualitative approaches the extent to which the researcher puts forward a logical case, argues for and justifies conclusions, explores and disconfirms alternatives and so on, has a bearing on the evaluated quality of the research.

It has to be accepted that the problems of bias and subjectivity identified by Shanks et al. (1993) and Baskerville and Wood-Harper (1992) will always exist with approaches such as action research, but if the research and its reporting are carried out in a disciplined manner which seeks to explain rationale at each step, then this makes for a sound basis on which to evaluate the quality of conclusions reached.

Ultimately, the question of selecting a research approach rests on which is the 'best fit' for the research question identified. Overall, the advantages of action research outweigh the disadvantages for this particular research and this has been argued from a number of perspectives. Particularly attractive is the congruence of some of the features of action research with the nature and basic mission of this research programme.

5.6 Summary

In this chapter the rationale for choosing action research as the research approach for this research question has been set out. This has been done by working through a number of steps in a decision process. Firstly, the background to whether a positivistic or interpretivistic approach to research was more appropriate was discussed. Secondly, the implications of the decision to adopt an interpretivistic style for this particular research question were analysed and a number of observations were made which assisted later in selecting an approach. Thirdly, a short-list of possible approaches was obtained by eliminating inappropriate ones. Finally, a decision to use action research was confirmed. This involved identifying action research as the 'best-fit' and also reviewing its recorded weaknesses with a view to mitigating these.

The next chapter describes the detailed components of each phase of the research and therefore continues to demonstrate the appropriateness of action research for this research question.

Chapter 6

Research Design

6.1 Overview

This chapter contains a detailed account of the research design for this thesis. The last chapter attempted to justify the choice of action research. In this chapter the decision to use action research is assumed. However, it is still necessary to describe how the research programme has been synthesised. Figure 6.1 shows the main steps in this process.

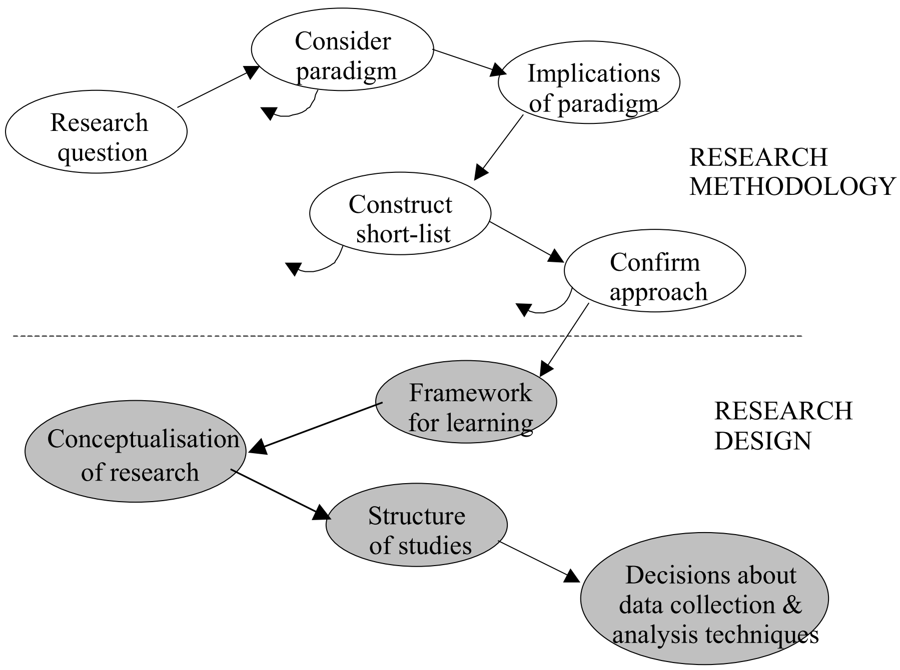


Figure 6.1 Cognitive Map indicating Steps in Research Methodology and Design

The chapter proceeds as follows. In the first section, a framework for learning is discussed. In line with discussion in the previous chapter regarding how to mitigate the weaknesses of action research, the section describes in some detail an explicit approach for framing and guiding the research in terms of learning. The second section deals with the conceptualisation of the research. Conceptualising the research involves identifying how the learning framework can be embedded into the action research process. Further it is helpful to consider how this research might be classified hence there is a discussion on types of action research. The final two sections of this chapter detail the structure of the studies themselves and the data collection and analysis techniques used.

In this chapter, examples are provided in terms of what was actually done rather than what was planned at the outset. This is because specific events and decisions arising during the course of the research caused me to change the focus of this research and clearly this had an impact on the detail of the studies themselves. This retrospective device has been used because it allows me to 'inform and explain' the nature of the research design to the reader in a more efficient and straightforward manner.

6.2 Framework for Learning

Over the years, Peter Checkland has continually argued for the use of an intellectual framework to guide action research (Checkland 1985, Checkland 1991, Hindle et al. 1995). An intellectual framework is a set of ideas or a theory which *'a priori are thought to be relevant to an understanding of the problems faced*' (Hindle et al. 1995, p455). As such an intellectual framework is both a means of expressing the underlying philosophical perspective that the researcher adopts in carrying out the research and also a means of interpreting and evaluating that research. Figure 6.2 shows how the intellectual framework interacts with the learning process.

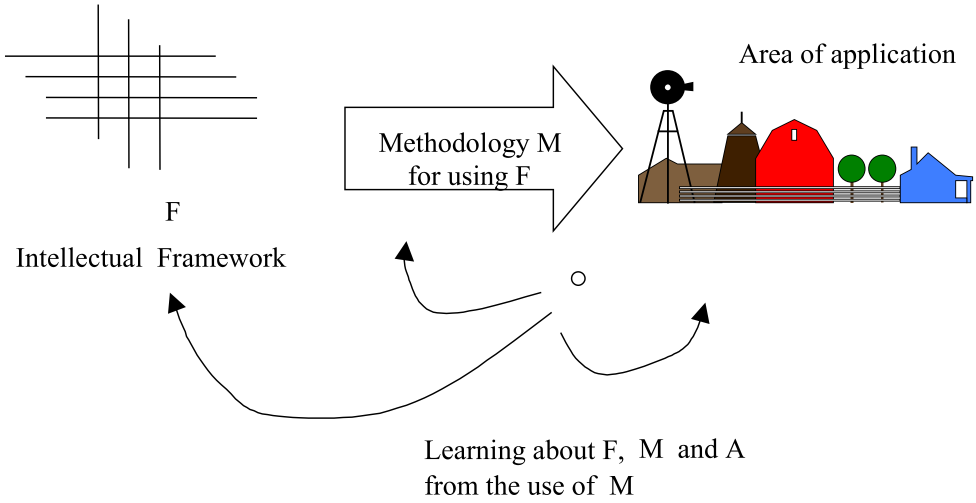


Figure 6.2 The Organised Use of Rational Thought, after Checkland, 1985

In figure 6.2, F represents a framework of ideas which underpin the application of a particular methodology M in an area of application A. The purpose of the Checkland (1985) paper was to discuss soft systems thinking in terms of a learning paradigm. Specifically, the framework of ideas (F) discussed was soft systems ideas whose genesis Checkland attributes to Vickers, Ackoff, Mason and Mitroff and others (Checkland 1985) and the methodology (M) was SSM (Soft Systems Methodology). Checkland's point was that there is strong intellectual consistency between a framework of ideas based on soft systems ideas and a methodology which facilitates the application of soft systems ideas. However, Checkland also makes the point that '*This is a very general model of the organised use of rational thought and applies not only to O.R. but to applied natural sciences as well'*. (Checkland 1985, p758). In other words, Checkland was presenting this as a general **learning** framework that could be applied in many types of research circumstances. An advantage of this approach is that it forces the researcher to make **explicit** what is considered relevant to the problem on hand.

6.2.1 Chosen Intellectual Framework

The work of Wand and Weber (1993) hereafter referred to as W+W, is based upon a particular ontological model developed by Bunge (1977, 1979). Ontology is a branch of philosophy concerned with articulating the nature and structure of the world. Many ontologies are possible and indeed the different versions of the diagram that I developed represent different ontological views of the world. Of particular relevance here is the notion of mappings of ontological constructs onto design constructs. Ontological constructs are asserted to represent relevant constructs in the real-world. For example, when working with the BED (Business Event Diagram), the assertion was that events were a fundamental construct of the real-world and so the ontology reflected that. Design constructs are the set of constructs which exist in the design world and these should be able to represent the constructs in the ontological world. In other words, design constructs are syntactic representations of semantic concepts in the ontological world. So, for example, in the BED there was a notational symbol to represent the event. For ease of understanding by the analyst, each version of the diagram was written as a paper and included descriptions of both the ontological and design worlds.

When the research began, it was viewed largely as a confirmatory exercise in the field. The assumption was that the research would corroborate the technical or software engineering ideas embodied in the technique. The BED was to be drawn by elaborating entity life histories which themselves were elaborations from a data model. W+W provide a framework which is consistent with and arguably formalises this view of elaboration. They speak of the mathematics of mappings in which the real-world is mapped via scripts into the machine world (figure 6.3). In the context of the BED, script 1 could be a data model, script 2 entity life histories and script 3 the BED.

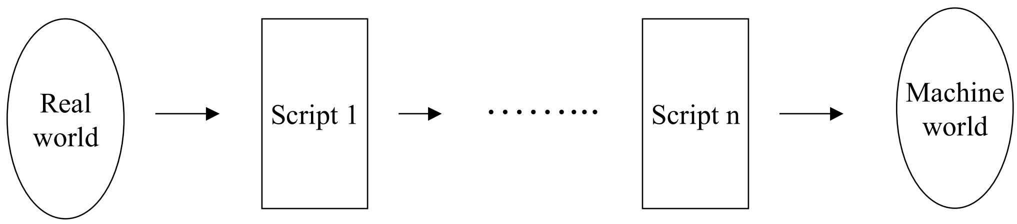


Figure 6.3 Mapping of Scripts from Real World to Machine World

Figure 6.4 shows five scenarios. In figure 6.4a, ontological completeness is shown. This means that there is a good mapping between the ontological world and the design world, ie all ontological constructs have a corresponding design construct and all design constructs have an ontological construct. In figure 6.4b however, there is ontological incompleteness. Here one particular ontological construct does not have a corresponding design construct. W+W (1993) argue that ontological incompleteness is undesirable because the design world would not be able to represent all real-world phenomena. In figure 6.4c a situation called construct overload exists. Construct overload occurs when one construct in the design world represents two (or more) constructs in the ontological world. Again, this is argued to be undesirable because it lacks clarity and may lead to confusion or ambiguity when reading design artefacts. Figure 6.4d shows construct redundancy which in one sense is the opposite of construct overload. In construct redundancy there are two (or more) constructs in the design world which can represent a single construct in the ontological world. This is considered undesirable because designers have to work with more constructs than necessary as well as leading to possible confusion. Lastly figure 6.4e describes construct excess. Construct excess arises when a construct exists in the design world which does not have a counterpart in the ontological world. The presence of construct excess may indicate three things. Firstly, it may indicate deficiency in the ontological world. Secondly, it may suggest that the design world contains an unnecessary construct, ie one which is outside the scope of interest of the ontological world. Thirdly it may be symptomatic of a general lack of understanding about the nature of the problem.

6.2.2 Relationship of Intellectual Framework to Research Question

Wand and Weber have applied their ideas to a number of existing techniques and methodologies although of course as stated previously these ideas have not been used in the design of new diagrams. For example, their ideas on mappings and scripts have been used in evaluating object-oriented concepts (Parsons and Wand 1991), the dataflow diagram (Wand and Weber 1993), the entity relationship model (Wand and Weber 1995) and the NIAM methodology (Wand and Weber 1996).

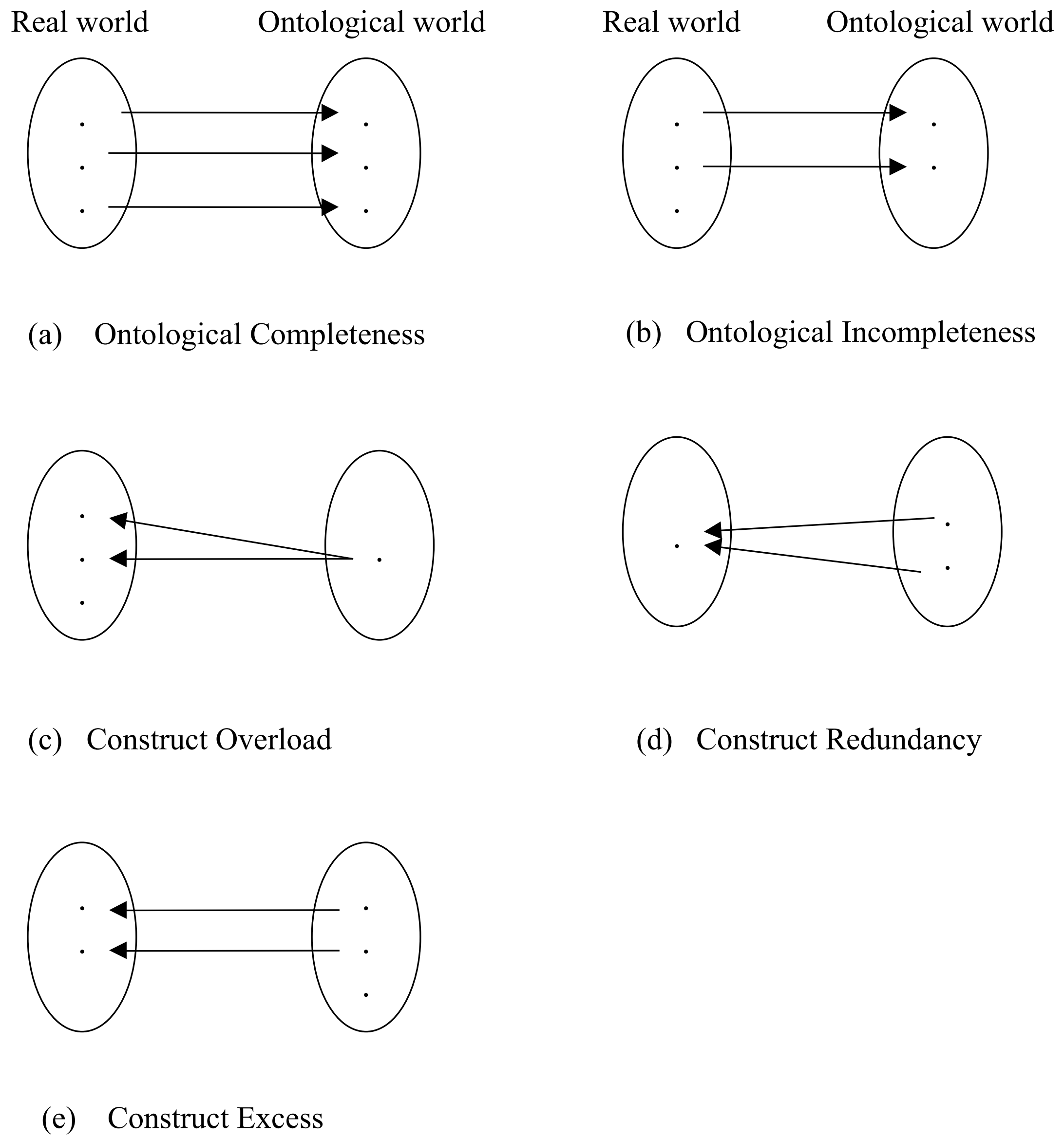


Figure 6.4 Five Scenarios in Ontological Framework, after Wand and Weber (1993)

The above ideas are the basis for the intellectual framework and they relate **directly** to the research question in the following way.

Structure

In the research question, the phrase ‘what is the structure of’ is to be addressed by the ideas implicit in figure 6.4 above. In other words, it is suggested that by considering types of mappings from the ontological world to the design world, a researcher can obtain insight to issues directly connected to the design of a new diagram.

Construction steps

In the research question the phrase 'what are the construction steps' is to be addressed by using the script model concept in figure 6.3. Here it is argued that construction steps or stages in the construction of a technique can be modelled and therefore be better understood by identifying the design worlds explicitly as indicated by figure 6.3.

6.2.3 Learning Cycle

Figure 6.5 shows the learning cycle for this research. By learning cycle, I refer to the different stages at which learning was considered to have taken place and to the different levels of learning in terms of the scope of review that took place.

In all, four versions of the diagram were developed using this process in the first action research study. As each version of the diagram was applied to the area of application A, a review was conducted against the framework of ideas. Where situations were encountered which were the equivalent of one of these scenarios, this served to re-inforce the legitimacy of the initial framework as an appropriate one for guiding and interpreting this research. Where situations were encountered which fell outside these scenarios, this indicated that the framework was possibly deficient in some way. This type of learning I have termed level 1 learning.

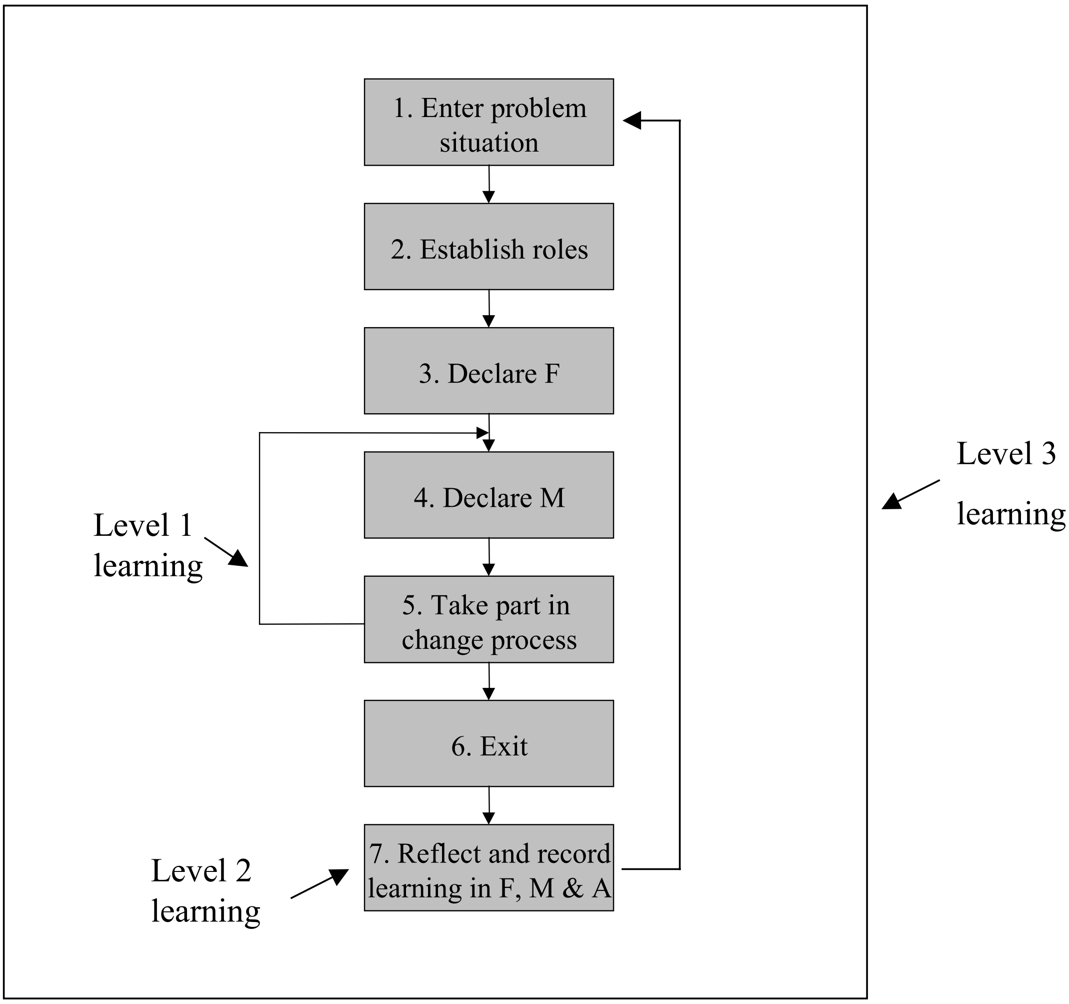


Figure 6.5 Flowchart of Steps in Research Process, adapted from Checkland (1991)

After four iterations of this cycle, it was felt that it was time to exit from the process and take stock of the whole approach. In chapter 11, a critical review of the W+W framework was undertaken. The outcome of this was to update the initial framework with additional ideas in readiness for a further action research study. In other words, the value of the existing framework of ideas was reviewed and where gaps were perceived to exist these were replaced by a suitable approach. This type of learning is termed level 2 learning.

In chapter 13, after the second action research study, a critique of the whole approach was undertaken. Here the intellectual framework was reviewed holistically in an attempt to discuss what had been learnt about the principle of using an intellectual framework for this kind of research and thus was termed level 3 learning.

So, in summary, the learning cycle occurred at three levels. These were:

Level 1: at each step of the methodology, a review in terms of applying the framework of ideas was undertaken and documented,

Level 2: at the end of the first action research study a deeper review of the whole approach was undertaken and documented which involved updating the framework with additional ideas and

Level 3: at the end of the research a critique of the whole idea of a framework was performed in order to understand, reflect and record the learning and experience of this research.

6.3 Conceptualisation of the Research

The main task of this section is to show how the learning framework was embedded into the design of the research. My position here is that a learning mindset is vital to successful action research. I use the term conceptualisation as a label to describe the process of internalising action research as a research process and to consider how the likely components of this research can fit together in a manner which supports the learning goals. As an introduction to this discussion, some background is provided on the types of action research to be found in the literature before examining process models which support learning.

6.3.1 Types of Action Research

Different authors cite the use of different types of action research methods. For example, table 6.1 shows some of the variant forms of action research to be found in the literature.

Author	Variants
McKernan (1991)	scientific action research practical deliberative action research critical-emancipative action research.
Mansell (1991)	operational research soft systems methodology socio-technical systems design viable systems diagnosis
Dick (1992-3)	participatory action research action science soft systems methodology evaluation (actually a family of methods)
Baskerville and Wood-Harper (1996a)	canonical action research action science action learning action case organisation development process consultation information systems prototyping soft systems methodology Multiview

Table 6.1 Types of Action Research Method

Analysis of the table reveals that there are two ways in which these action research methods are typically labelled. The first way is determined by the specific activity or problem on hand. For example, process consultation is a term used where an outside consultant is involved in organisation development in a particular company (Schein 1969). An example of this sort of labelling in relation to systems development would be information systems prototyping.

The second way in which action research types are labelled is by emphasising a particular characteristic of action research by adding an adjective to highlight the dominant characteristic eg participatory action research emphasises the participatory nature of the activity (Whyte 1991).

I consider socio-technical action research to be a good descriptor of the type of action research method employed in this research. The ‘socio-technical’ label is an example of the dominant characteristic approach just discussed and is so labelled to emphasise the social aspects of this research. The socio-technical approach to action research emphasises that a problem situation can be viewed from both a social perspective and a technical perspective at the same time and further that the social and technical perspectives need to be integrated with each other in order to achieve an overall acceptable resolution to a problem situation (Susman 1983). One well known example utilising socio-technical ideas is a method called ETHICS (Mumford 1985). The ETHICS method involves a step-by-step approach in which a 'best fit' is sought between social objectives (such as job satisfaction) and technical objectives (ie those which the organisation lays down as part of an employee's job description). In integrating the social and the technical it is accepted that for example the technical objectives may have to be reviewed (ie compromised to some degree) in order to achieve a 'best fit'. Behind this view is the recognition that a 'best fit' is in fact a more satisfactory overall solution to a problem situation.

In the context of this research, the technical objectives are considered to be those concerned with the specific constructs identified in the diagram and to some degree the construction of the diagram. It will be recalled that originally the technique was developed from a technical or software engineering perspective. Underpinning this research is the question of to what extent do such technical objectives fit with the social environment of those using the technique. For example, how do analysts find the technique? Are users comfortable with the vocabulary used, with the diagrams they are asked to work with and so on? In the course of this research I have added a new type of diagram called the User Business Rules Diagram in order to make users more comfortable by focussing on only the rules that concerned them. Further, socio-technical action research involves a continuous interplay between the social and technical as assertions are continuously tested in subsequent iterations of the approach. The term socio-technical action research is therefore an appropriate label for this research.

6.3.2 Process Models of Action Research

Kurt Lewin is often considered the father of action research, although there is some dispute (Elden and Chisholm 1993). Much of the early work in action research addressed social problems and later it was used in socio-technical contexts (Trist and Bamforth 1951). There is significant use of action research in many areas of the social sciences. For instance, there is a strong tradition in education where it is has revitalised thinking on the curriculum and in educational change generally (Elliott 1991).

Process models refer to the generic activities undertaken within action research (Baskerville and Wood-Harper 1996a). Of particular relevance here is the idea of a learning cycle which underpins much of action research. Lewin developed a process model which can be considered a cyclic model. Later adapted by Burns, this is shown in figure 6.6.

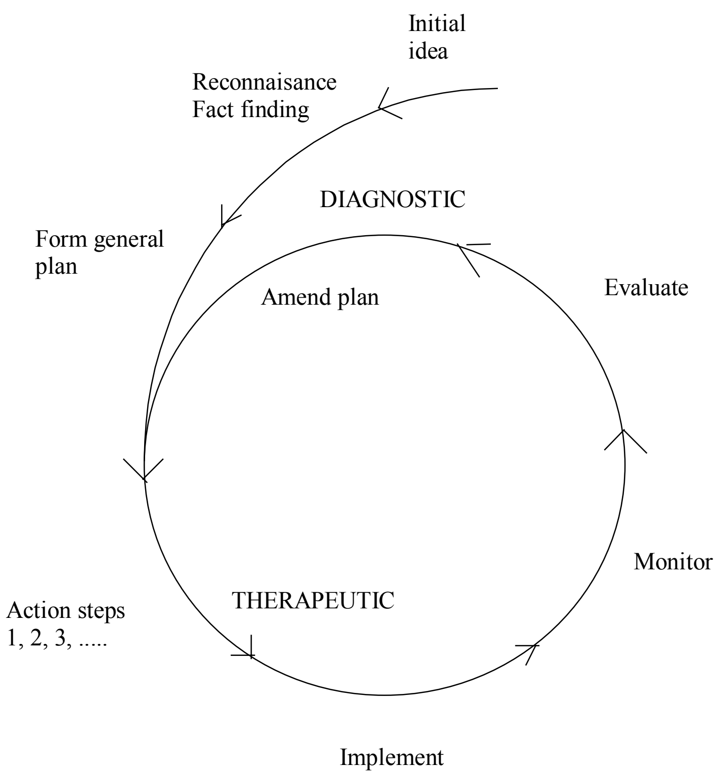


Figure 6.6 Lewin's Cyclic Model, adapted by Burns (1990)

A similar but later model of the cyclic nature of action research is provided by Susman (1983, after Susman and Evered 1978) in the context of a discussion on socio-technical action research and is shown in figure 6.7.

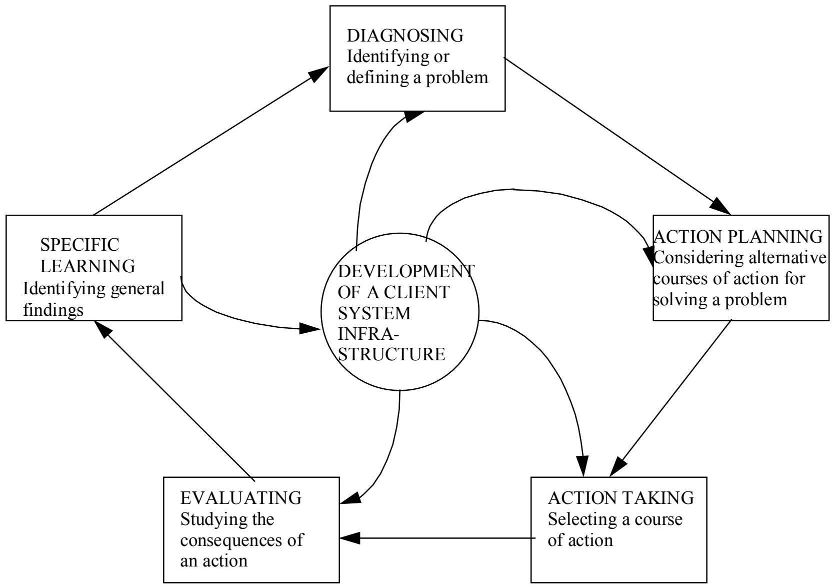


Figure 6.7 Sociotechnical Action Research Cycle, after Susman and Evered, 1978

Though these models are similar they are not identical. However, one important feature they have in common, is the theme of a learning paradigm described by Kolb et al. (1979). See figure 6.8. Here a reflective-action cycle is achieved through a spiral of concrete experience, observation and reflection, abstraction and generalisation and then testing of concepts. This is a very powerful aspect of action research since it facilitates continuous and on-going learning (Baskerville and Wood-Harper 1996a).

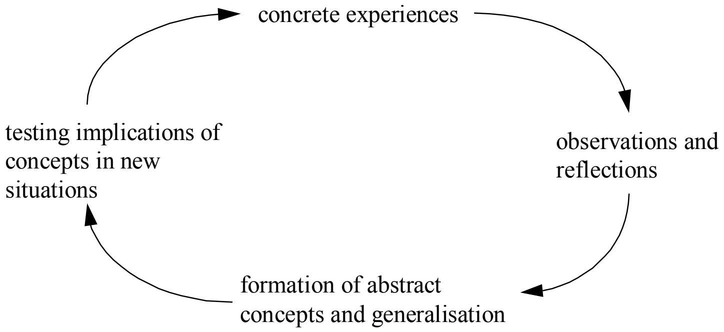


Figure 6.8 Kolb's Experiential Learning Cycle, after Kolb et al. (1979)

I have adopted a similar approach to the Kolb et al. (1979) experiential learning cycle in the reporting of my research although slightly different terms to describe each of the phases have been used. I use the terms preparation, application, review and decision to record progress in the reporting of the research and the reason for using these terms arose out of the specific nature of this research project. The term preparation involves the preparation (ie the writing) of a new version of the Business Rules Diagram (BRD). The need for new versions arose out of previous reviews. The preparation step was a significant step in terms of the effort expended since it involved in essence the construction (ie writing) of a paper which reflected new ideas and concepts. Further, as the paper itself was being written I found that some of the general ideas I was writing about were impractical. So, these were modified as the paper was being written. In other words, the writing of each paper was in itself one kind of 'testing of the implications' of the 'abstract concepts and generalisations' in the previous step to use Kolb et al.'s terms. Accordingly, for the purposes of reporting, the preparation step is the equivalent of the 'testing of implications' step in Kolb et al.'s model. The other steps in the research map much more easily onto the Kolb et al. cycle, ie the application step roughly equates to the 'concrete experiences' step, the review step to the 'observations and reflections' step and the decision step to the 'formation of abstract concepts and generalisation' step. Table 6.2 shows an approximate mapping between the steps I have used in the reporting of my research and the models by Kolb et al. (1979), Burns (1990) and Susman and Evered (1978).

McDermid	Kolb et al.	Burns	Susman and Evered
preparation	testing implications of concepts in new situations	amend plan, action steps 1,2,3...	action planning
application	concrete experiences	implement	action taking
review	observations and reflections	monitor	evaluating
decision	abstract concepts and generalisations	evaluate	specific learning, diagnosing

Table 6.2 Mapping of Process Steps in Different Action Research/Learning Models

6.4 Structure of the Research Design

In terms of executing the research, there were three main components. These were an action research study, a critical review of the framework used and then a second action research study. The two action research studies had a different focus or emphasis. In the first action research study the views of the analyst only were sought. In the second study the focus was widened to include not only the analyst's perspective but also the views of users. In other words, overall a three stage strategy has been adopted, ie from software engineering idea to analyst view to analyst and user view. The two studies are also differentiated in that different individuals, different applications and different organisations were involved thus increasing the number of sources of data used thereby improving the confidence of any conclusions obtained.

The critical review was seen as a milestone in the progress of the research. It had always been understood that the initial study would end at some point. By the end of the fourth iteration of the diagram, there were a number of issues such as complexity which it was felt that the initial framework was just not addressing. The critical review offered an opportunity to stand back from the detail of the early work and reflect on progress.

6.4.1 First Action Research Study

In this study I worked with an experienced analyst on an application in the analyst's organisation. The analyst was familiar with the application though initially I was not. The original intent of this study was to test the diagram, ie the initial purpose of the research was seen as confirmatory rather than exploratory. As it turned out, experience with working with the analyst on a real application resulted in four major revisions of the diagram and so the methodology was applied to the application area on four occasions (M1 through to M4). In other words, the methodology was considered to be different on each occasion because each version of the diagram contained a different ontological view. The same framework of ideas F and area of application A were used in testing all four methodologies. Figure 6.9 summarises this diagrammatically.

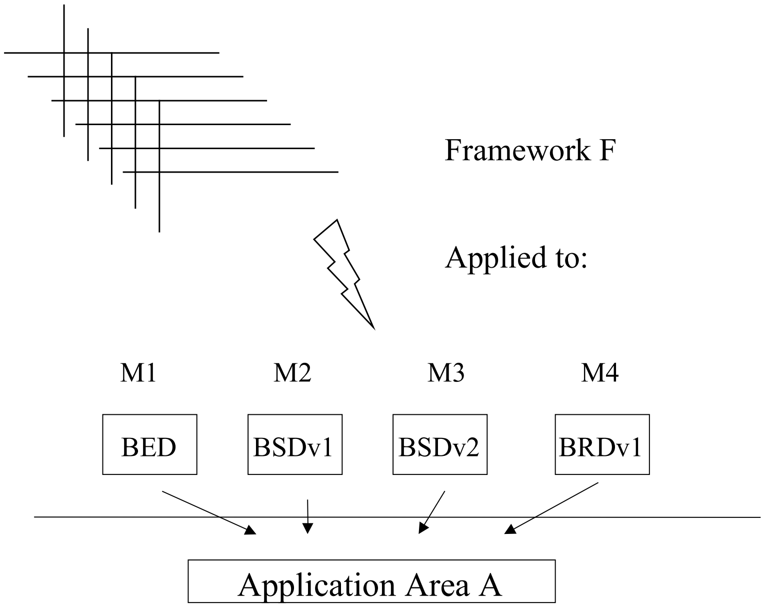


Figure 6.9 Framework F, using one of M1, M2, M3 or M4 applied to Application Area A in the First Research Study

Boland (1990) defines hermeneutics as the study of interpretation. Hermeneutically, the ideas and assumptions that I was seeding into each version of the diagram were interpreted by the analyst who in turn responded back to me with feedback. So in the first iteration my ideas and assumptions were reflected in the Business Event Diagram (BED) paper. After working with the analyst on an application using the BED, I obtained feedback from the analyst (figure 6.10). That feedback caused me to modify my ideas and assumptions to the extent that a new diagram was developed, ie the Business State Diagram version 1 (BSDv1). In the second iteration my ideas and assumptions reflected in the BSDv1 were used on the same application with the same analyst and again feedback was obtained. This cycle continued through a further two versions of the diagram.

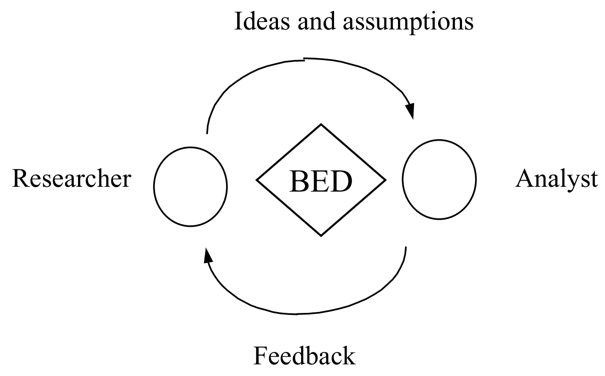


Figure 6.10 Learning Cycle in Action Research Study 1

This study not only confronted technical questions (such as the technicalities of modelling events as opposed to states) but also it entered the social domain by requiring the analyst to consider the appropriateness of the diagram holistically (ie not just technically) and in the context of his working environment. So as each new version of the diagram was developed elements of social as well as technical questions, problems and solutions were being aired and resolved. For example, as the transcript of the interview confirms, the analyst stated a preference for modelling in states rather than events.

6.4.2 Critical Review of Research Approach

Probably the most valuable benefit of the critical review was that it provided an opportunity to re-assure myself that the research was productive and that real progress was being made. The work in the first action research study had not gone as planned. It had been expected that the ideas in the original State Dependency Diagram would be vindicated. Instead, a completely new focus of a business rule had emerged from the research process. The critical review afforded the opportunity to re-affirm the validity of this shift and to consolidate it by planning how to continue the development of ideas in the next study. In terms of the framework of ideas being used, though the Wand and Weber (1993) concepts had been useful, it was clear that there were many areas which were not being addressed by their framework and that the framework of ideas required to be updated to deal with these issues.

Figure 6.11 describes the process of the critical review. Since many changes had resulted from decisions taken at the end of the first action research study, it was decided to write these into a revised version of the diagram (BRDv2) as it was felt

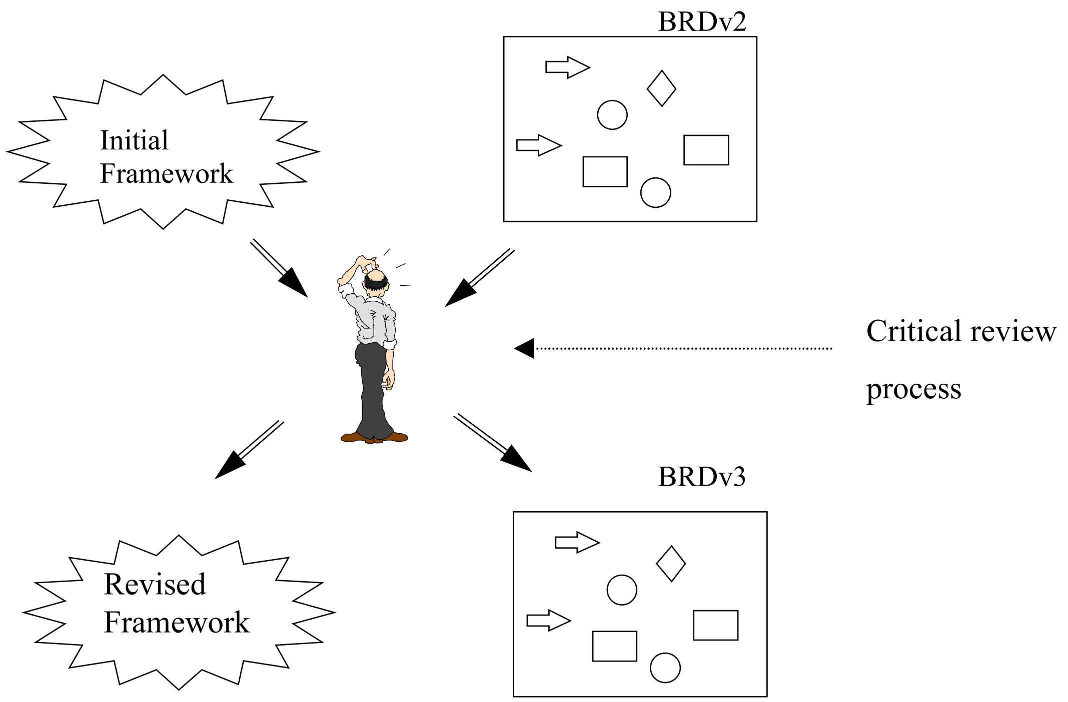


Figure 6.11 Changes resulting from Critical Review

desirable to have a fresh version of the diagram before moving into the critical review. The critical review involved examining and categorising the types of problem that had emerged over the course of the first action research study. Once categorised, Wand and Weber's model was then extended to depict these additional problem categories. The framework of ideas was then revised by adding a number of ideas from the literature which were selected to address some of the problem areas. So for example, McCabe's complexity metric (1976) was used as a basis for assessing the complexity of a particular instance of a diagram and was thus added to the framework of ideas.

Because the focus in the first action research study had been the analyst's view, relatively little attention had been paid directly to user issues and concerns. This was particularly so with respect to complexity. Since its inception, it had always been intended that the diagram would be used with users. However, it was argued that if complexity was perceived as high for analysts then it may be virtually impossible for typical users to work with the diagram as it stood. So a major change in the diagram was introduced at that time (along with other changes) to deploy a simpler diagram for users. This was written into a new version of the diagram (BRDv3).

The outcomes of the critical review were therefore a vindication of the research process so far, a revised intellectual framework, a new version of the diagram and a readiness to move to the next phase of the research which would involve users as well as analysts.

6.4.3 Second Action Research Study

The rationale for this study stemmed from the desire to involve users. Until now no real feedback had been obtained directly from the perspective of the user (as opposed to the analyst). This last study was therefore constructed to explore the latest version of the diagram from both the perspective of the analyst and the user.

In this study a different analyst was used in a different organisation on a totally different problem. Unlike the first action research study I decided that I would not be directly involved in the intervention. Once the analyst was trained in the revised BRD (see appendix 1, version 3), he then trained two users in the User Business Rules Diagram (UBRD). The training and subsequent drawing of the UBRD was conducted in a cyclic fashion, because this was seen as the best way of introducing the concepts to the users. In other words versions of the UBRD produced essentially by the users were corrected by the analyst and in so doing served to train the users in the diagram. In addition since this version of the BRD incorporated many ideas from earlier versions, the opportunity to test earlier ideas existed.

The analyst and both users were interviewed by the researcher at the end of this study. The interview schedule was such that the users were interviewed prior to the analyst. This allowed the analyst to review the transcripts of the user interviews, before he was interviewed himself. Figure 6.12 shows this interesting hermeneutic situation.

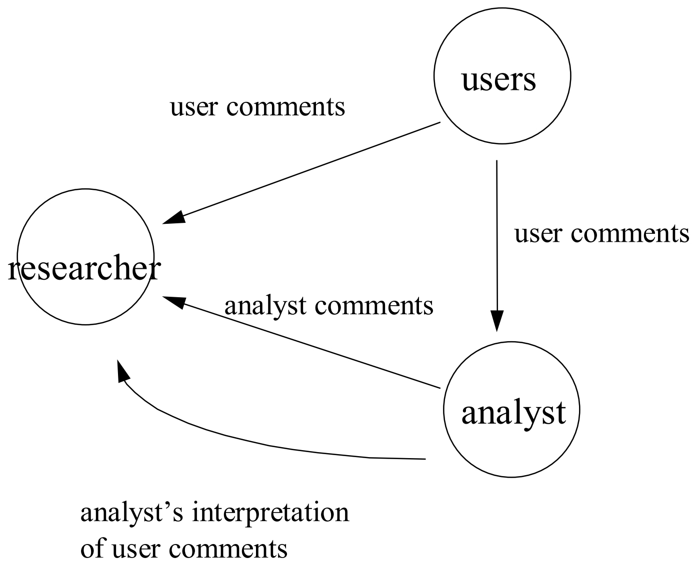


Figure 6.12 Feedback Paths in Second Action Research Study

This study clearly involves significant attention to social issues though the technical issues themselves were not forgotten. The users' views were solicited as were the analyst's views about the user views, so there is clear emphasis on a user perspective. Further, the analyst's views themselves are also part of a social perspective since the analyst would be expected to work with the diagram too. So it can be said that this study is a 'socio-technical' one in that it clearly took into account and assessed both social and technical aspects of the diagram.

6.5 Data Collection and Analysis Techniques

The purpose of this section is to provide an overview of the research instruments which were used within the research design. Research instruments are the data collection and data analysis techniques which assist in evaluating aspects of the research question. Research instruments should be consistent with and appropriate for the type of research undertaken. Having said that, since a deeper appreciation of these instruments and their role in the research can only be seen in the context of the actual studies, a more detailed review is left for later chapters. Here, the purpose is simply to introduce them and to inform the reader of their broad role in the research design.

6.5.1 Semi-structured Interview

McKernan (1991, p129) maintains that '*one of the most effective modes of gathering data in any inquiry is through the interview method'*. He identifies three types of interview. These are the structured, semi-structured and unstructured interview. In the structured interview the questions are predefined and the interviewer is constrained to the script of the interview. As a consequence, an underlying assumption of a structured interview is that questions and therefore also issues are known in advance of the interview. By contrast in the unstructured interview both the issues and topics are left to the interviewee and so the danger here for the researcher is that topics of interest to the research will never be aired. The semi-structured interview offers a good compromise between these two extremes especially if the interview is channelled into progressively more detailed or focused questions. Initially, the questions are as general as possible offering the interviewee the opportunity to talk in his or her own terms. Later in the interview more directed questions occur thus enabling the researcher to get answers to specific research related issues. Skill is required in the way this is done. A good interviewer will use earlier quotes or themes raised by the interviewee in order to drill deeper into specific questions.

The semi-structured interview was used in both studies. In the first study the analyst involved was interviewed about each of the four versions of the diagram. Prior to the interview a template of questions was prepared. Each template reflected the ontology that was being tested in that version of the diagram and thus it served as a framework for conducting the semi-structured interview. The templates are reproduced in appendix 1. In the second study, analyst and users were interviewed. As indicated earlier the users were interviewed first so that the analyst could look at the interview transcripts of the users and therefore be in a position to comment on them during his own interview. Again, templates were used as a means of initially structuring these interviews.

6.5.2 Content Analysis

In content analysis, themes are identified from text. The text may be for instance the transcription of an interview or a document such as the minutes of a meeting. These themes are coded into a system of categories from which it is hoped some theory will emerge. '*Codes are tags or labels for assigning units of meaning to the descriptive or inferential information compiled during a study. Codes are usually attached to "chunks" of varying size - words, phrases, sentences, or whole paragraphs, connected or unconnected to a specific setting. They can take the form of a straightforward category label or a more complex one (eg a metaphor).'* (Miles and Huberman 1994, p56).

There are basically two approaches to content analysis. These are the inductive and deductive (or 'a priori') approaches. In the inductive approach, the view is taken that the codes are 'grounded' in the data (Glaser and Strauss 1967). There is no prefabricated list of codes identified by the researcher before coding commences; the codes are identified by examining the data. Of course any research is bounded by the context in which it occurs. For example, even open-ended questions in a semi-structured interview direct the interviewee along certain lines by the words used and the underlying philosophies or attitudes implied by them. So it is questionable just how theoretically sound the grounded approach really is. On the other hand the 'a priori' approach uses a set of pre-defined codes or categories and therefore it makes assumptions about the nature of the underlying constructs and issues relating to the question and thus doesn't allow interviewees to respond in their own terms. A criticism of the 'a priori' approach is that it may lead to neglecting or overlooking other better ways of looking at the problem or even worse to not identifying some issues at all.

Content analysis was used in both studies. Since templates were used it may be natural to conclude that an 'a priori' approach was taken. That is only partially correct. The themes in the template certainly were used but these were based on ontological issues thought to be relevant at that point. However, there was no such presumption about social issues. For example, in the first study the analyst was asked how he found states rather than events as the main focus of the diagram. He was not given any guidance after that. In other words, he was left to talk about his feelings. I then interpreted this into themes.

6.5.3 Documentary Analysis

In documentary analysis, documents in the environment of the research are used as a source of evidence in analysis. As McKernan (1991, p150) puts it '*a rich source of evidence for the research practitioner can be found in documents, such as texts, newspapers, minutes of meetings, articles, letters, diaries, memos or scripts - indeed any written account may be considered a document'.* To McKernan's list is added any graphical document which in the context of this research would include the BRD and other diagrams used in the creation of the BRD. Of course content analysis can be performed on documents generally and not just transcripts of interviews and thus the number of potential sources for subsequent analysis increases.

Three types of documentation are to be found in this research. The first is that for each version of the diagram a formal paper was written reflecting ideas and philosophies current in that version. Depending on the version, the size of the paper ranged from about half a dozen pages up to about thirty. Each version is documented in appendix 1.

The second type of documentation is the diagrams of the application areas which were actually produced. In study 1, diagrams were produced by the analyst and researcher working jointly on a business application. Sometimes these diagrams appeared as documentation for that version. In other words they became part of the formal paper representing that version. In addition the analyst was asked to draw versions of the diagram using a different business application with which he was familiar. In the second study, the analyst and users produced versions of the UBRD and BRD.

The third type of documentation is the minutes of meetings held between the analyst and users during the second study. Again, these provide a rich picture of the thinking and progress at that juncture.

6.5.4 Field Notes

Because action research is a form of inquiry that investigates phenomena in their natural context, it makes sense to make notes of impressions at the time so that these can be reviewed analytically later. Field notes can provide clues to issues and problems associated with the research. '*A major strength of the field note approach is that it is not rigidly structured and thus leaves the researcher open to the unanticipated and unexpected; the researcher sees it as it is, not as it is intended'* McKernan (1991, p94).

Field notes were used in the first study but not in the second study since as a researcher I was not directly involved in interacting with the analyst and users at their sessions. In the first study, many of the field notes made were questions about the diagram produced at the previous meeting and as such were more operational in terms of assisting progress from meeting to meeting. However, when an issue of significance arose, I often used the field note to express the thought. For example, I first speculated that a trigger might be a valid component of a business rule using the field note. Where field notes express some form of useful speculation, these have been seeded into the reporting of the research progress in later chapters. However, I have omitted inclusion in this thesis other field notes which relate to operational questions because these have only indirect bearing on research progress. In any case, I consider that the formal papers of each version of the diagram served to encapsulate my thinking at that point and therefore can be considered a more significant source of documentary evidence in this respect.

6.6 Summary

The purpose of this chapter was to provide a detailed account of the research design. This was done firstly be introducing a framework for learning. The idea for this was proposed by Checkland (1985). The concepts for the initial framework of ideas were developed by Wand and Weber (1993). The second section addressed the conceptualisation of the research. Here the research was classified with respect to other action research types and also how the learning cycle implicit in action research would be incorporated into the research design was planned. In the third section, a detailed outline of the main components of the research was provided. These were an action research study, followed by a critical review which was then followed by a second action research study. In the first action research study, the main emphasis was on the analyst's perspective both technically and socially. In the second study the focus was broadened to include a user perspective as well as an analyst's perspective. The final section of this chapter was devoted to describing the research instruments used in the research. As each instrument was described, its role and purpose in this research was also outlined.

Chapter 7

Action Research Study 1

The Business Event Diagram

7.1 Overview

In this chapter the background to the first action research study is described as well as the detail of the study concerning the Business Event Diagram. The next section provides an overview of the structure of the whole study indicating the main conceptual steps in the study. The following section outlines details about the roles of the participants and the general purpose of the research. In action research, this is often called the client-system infrastructure. This is followed by a section describing the business application used. In action research study 1 a number of different versions of the diagram were developed. These were the Business Event Diagram (BED), the Business State Diagram version 1 (BSDv1), the Business State Diagram version 2 (BSDv2) and the Business Rules Diagram version 1 (BRDv1). However, this chapter only describes the development of the BED. The last four sections in this chapter detail the development of this version only. Later chapters deal with the other versions of the diagram.

7.2 Structure of Action Research Study 1

Figure 7.1 outlines the main steps in the action research process. Each version of the diagram was developed by following these steps in an iterative fashion. In the first iteration, a revised version of the diagram (ie the BED) was prepared by updating the SDD (State Dependency Diagram) with changes that had occurred to me since the original publication of the SDD. This initial version was then tested by applying it to the business application. The study was undertaken in an organisation with a systems analyst as co-researcher. The successes or otherwise of implementing these changes in the diagram were then reviewed together with any other issues that emerged from this review. Clearly the review findings stage is an important reflective step in any research process. The next step was to decide on changes. Not every issue identified necessarily had a solution present itself, so some problems remained outstanding for up to several iterations. The last step in the cycle was to prepare a new version of the diagram which embodied the changes decided upon. This general process was then repeated for each new cycle (iteration) of the diagram.

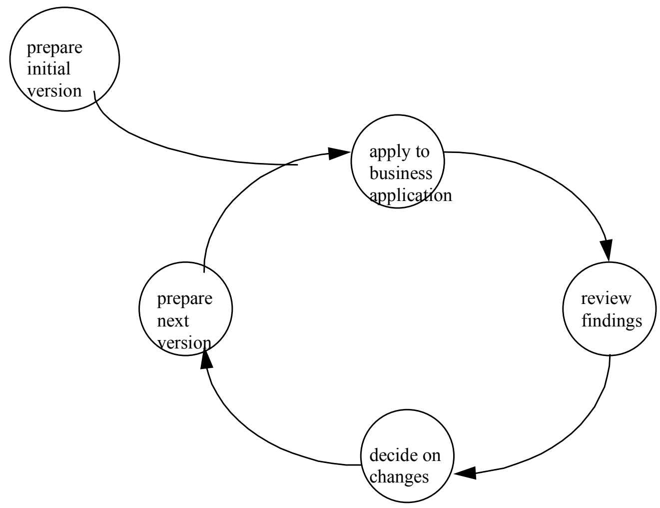
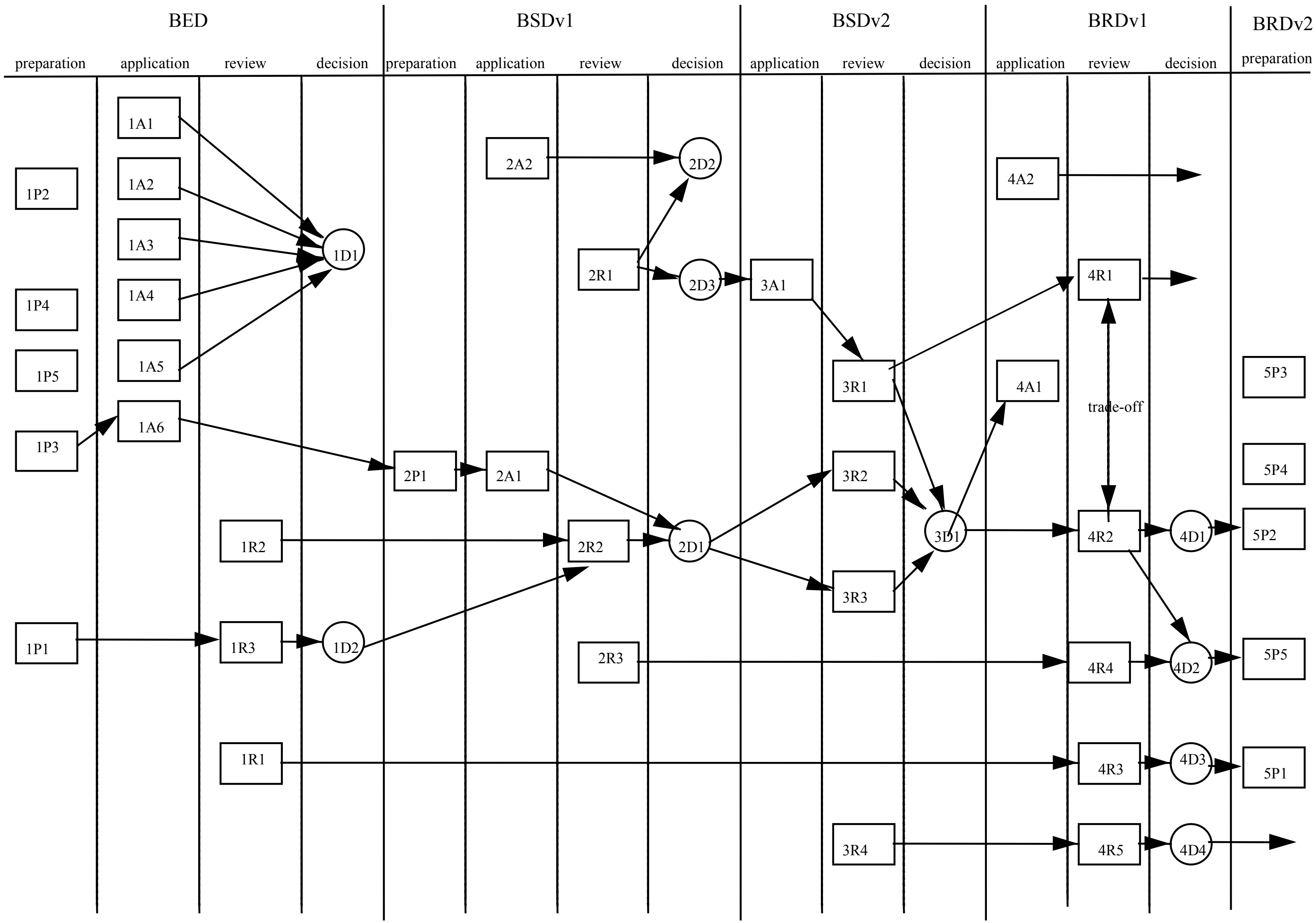


Figure 7.1 Main Steps in Action Research Process

Figure 7.2 is a type of cognitive map of the specific issues and solutions that were identified during each iteration in this study. Overall it charts research progress through the various versions of the diagram and thus it is referred to here as the research progress map. Typically each version of the diagram is represented by four columns on the research progress map. Issues (shown as rectangles) may be identified in three different steps of the action research process, ie the preparation of a new version, applying the version to the business application


BED
BSDv1
BSDv2
BRDv1
BRDv2
preparation
application
review
decision
preparation
application
review
decision
application
review
decision
application
review
decision
preparation
1A1
1A2
1A3
1A4
1A5
1A6
1P2
1P4
1P5
1P3
1P1
1D1
1D2
1R2
1R3
1R1
2A2
2D2
2R1
2D3
2P1
2A1
2R2
2D1
2R3
3A1
3R1
3R2
3D1
3R3
3R4
4A2
4R1
4A1
trade-off
4R2
4D1
4R4
4D2
4R3
4D3
4R5
4D4
5P3
5P4
5P2
5P5
5P1

Legend:

1P1	event focus
1P2	object-oriented basis
1P3	group event notation
1P4	delay analysis step
1P5	event notation refinement
1A1	prerequisite flaw
1A2	union and intersection
1A3	alternative point of entry
1A4	fan out dependency
1A5	contingent dependency
1A6	simultaneity
1R1	IS events vs business events
1R2	rule completeness
1R3	state vs event
1D1	change notation
1D2	change to state
2P1	simultaneity
2A1	simultaneity
2A2	overkill
2R1	readability
2R2	rule completeness
2R3	manyness
2D1	extend prerequisite table
2D2	complex object
2D3	Harel blob
3A1	readability
3R1	complexity
3R2	simultaneity
3R3	rule completeness
3R4	construction from complex objects
3D1	states and events
4A1	simultaneity
4A2	readability and the simple event
4R1	complexity
4R2	rule completeness
4R3	IS rules vs business rules
4R4	manyness
4R5	direct vs indirect construction
4D1	show conditions and triggers on diagram
4D2	prerequisite table
4D3	refine definition of rule
4D4	direct approach
5P1	business rules versus information systems rules
5P2	trigger selection
5P3	Harel blob for simple conditions
5P4	sub-states and parallelism
5P5	event specification guide

Figure 7.2 Research Progress Map for Action Research Study 1

and in the review of findings. The first column represents issues identified during the preparation step. The second column indicates issues identified during the step which applies that version of the diagram to a business application. In similar fashion column three represents issues aired in the review of findings step. If at any step the issue identified is able to be resolved within that step by making a modification to the solution, then there is no continuing line for that issue on the diagram. On the other hand where issues remain, a continuing line is shown on the diagram. The fourth column is a summary column and shows where a decision (shown as a circle) has been taken to make a change in the next version of the diagram.

In order to link the research progress map to discussion in this chapter a simple coding system is used of the form ‘mXn’. The ‘m’ represents which version of the diagram is being referred to eg since the BED is the first version, the number here will be 1. The ‘X’ represents the step in the action research processing. Table 7.1 summarises possible values for ‘X’. Lastly, the ‘n’ is a serial number within the research step. So for example ‘1R2’ indicates the second review issue identified in the BED. These codes will be used throughout to facilitate cross-reference with research progress maps. Further the third column in table 7.1 contains the title of the column used in the research progress map. In each iteration of the diagram the sequence of discussion will follow the cycle indicated in figure 7.1, ie it will start with issues to do with preparing the new version followed by applying the new version to the business application. This will be followed by reviewing the findings and then deciding on changes although on some occasions discussion on these last two items is merged.

Step in Research Process	**‘X’ value**	**Column in Research Progress Map**
prepare next version	P	Preparation
apply to business application	A	Application
review findings	R	Review
decide on changes	D	Decision

Table 7.1 Codes in Research Process

The term deliverable is used here to mean any written or graphical output arising out of the action research study which can be used as a source of analysis. There were four main types of deliverable from this study. The first was a set of descriptions of each version of the diagram developed. Consequently, there is a description written as a paper for the BED, the BSDv1, the BSDv2 and the BRDv1. Appendix 1 contains these descriptions. These descriptions were the working versions used (or developed) at the beginning of each iteration of the action research cycle. As such they reflect the ideas and philosophies perceived to be significant at that point and thus were an important deliverable, if not the most important deliverable in this research.

The second deliverable was the diagram (and table) produced at each iteration of the research cycle. As will be explained later, in the latter iterations the preparation of the new version step was combined with the step which applied the new version to the business application. This explains why some versions of the diagram have only three columns in figure 7.2.

The third deliverable was field notes. The majority of these field notes contained questions of clarification that took place back and forth during each iteration of the diagram. However, also recorded were observations and even perceived revelations. Only these latter items are used and discussed later where it was considered that they contributed to the flow of argument.

The fourth deliverable was a transcript of a semi-structured interview which took place at the end of the study with the analyst. The interview was in two main parts. The first part reviewed each version of the diagram in an open-ended manner, so that the analyst's views at that point could be captured. Note, however, that by that time, the analyst had been exposed to and had worked with the later versions of the diagram and so therefore may have been influenced by ideas in later versions. In the first part of the interview a checklist was used. This checklist contains the ontological issues that I considered were directly pertinent to the research question. The checklists for each iteration of the diagram are also reproduced in appendix 1. In the second part of the interview more directed questions were put in an attempt to get the analyst's view on specific issues that had not arisen earlier. Prior to the interview the analyst had been asked to draw the diagrams again but this time for an invoicing application. He did this for each version of the diagram, ie all four versions. These diagrams were examined prior to the interview and used as a source for generating questions. Appendix 2 contains a full transcript of the interview.

7.3 Client-System Infrastructure

7.3.1 The Analyst

The systems analyst had worked in this organisation, a port authority, for about 8 years. His title is Systems Development Co-ordinator. He reports directly to the Information Systems Branch Manager and is responsible for a team of 3 analyst/programmers. His duties include development and maintenance. SSADM (Downs et al. 1988) is the methodology used in the organisation although not all steps in the methodology are actually performed in practice.

Recently the organisation had moved to an Ingress platform from Informix. The availability of Windows/4GL tools has stimulated interest in the use of object-orientation and thus the vocabulary of object-orientation had begun to permeate the information systems organisation. At the outset, the analyst was qualified as being a supporter of what he refers to as ‘state diagrams’. In the SSADM methodology, these are known as entity life histories (Downs et al. 1988). He uses these models as part of the documentation of systems and on some occasions had involved users with these diagrams. However, his colleagues including members of his own development team preferred not to use entity life histories in specifying information systems. This was an issue that the analyst wanted to correct and so from his perspective working with me in producing a more elaborate form of ‘state diagram’ was seen as an opportunity which could result in evidence being identified which might change the opinion of his colleagues on the value of using ‘state diagrams’.

7.3.2 The Role of Researcher and Systems Analyst

As the architect of each version of the diagram, my role was to explore the acceptability of the diagram with the systems analyst as co-researcher. The role of the systems analyst was that of domain expert. The intention was that we would work together on drawing diagrams on a business application with which the analyst was familiar. Where problems and concerns were encountered these would be freely discussed and further action taken if any jointly agreed. However, the level of discussion and the vocabulary used was based on the particular ontology being tested at that time, eg in terms of events, states, prerequisite tables and so on. I did not involve the analyst in a discussion of Checkland's intellectual framework or other philosophical matters relating to action research.

7.4 The Business Application

Since the business application selected was an existing live system, the systems analyst had considerable knowledge of the detail of the system. Using an existing system had the benefit of allowing us to focus on how well the diagram was able to perform its function, rather than concern ourselves with gathering the requirements from users (ie requirements acquisition).

The same business application was used in each version developed during this study. This was a portbilling application. It was felt that there was advantage in ‘testing’ each version of the diagram against the same business application. Further, it reduced the learning curve for the researcher. Prior to the semi-structured interview, the analyst was asked to draw each diagram again (this time by himself) but using a different business application. He selected an invoicing application for this purpose. The next two sections provide a brief overview of these applications.

7.4.1 Portbilling Application

A portbill is a bill of work done by a port authority for services performed on a ship which has used the port's facilities. Typical services include tugging, moorage, cranage and cargo handling. Since the ship may be in dock for several days, the services required accrue over that period. It is not always possible to predict in advance the exact nature of services required nor indeed the time it will take to carry out these services.

Each ship has a client who is the owner or charterer of the ship. In addition there may be a number of agents associated with that visit. Typically agents are local companies who have requisitioned the delivery of goods from the client. It is usual for agents to be responsible for the payment of specific port services required for those goods they have requisitioned. Those services not paid by agents are the responsibility of the client.

The analyst provided initial documentation of the portbilling application including the data model and data dictionary definitions for each entity in the data model. The data model, entity life histories and entity definitions for the business application are shown in appendix 3. Note the inclusion of portbill status and portbill item status as entities. Portbill status for example, contains only three occurrences. One occurrence represents 'portbill in processing', one 'portbill cancelled' and one 'portbill released', ie all work completed on portbill. In some methodologies such as SSADM, such entities are known as operational masters since they facilitate the fast retrieval of related entity occurrences by using the operational master as a kind of secondary index (Downs et al. 1988). However, they do add considerably to the number of entities on a data model.

7.4.2 Invoicing Application

The invoicing application is related to the portbilling application. In fact, one source of inputs to the invoicing system comes from the portbilling system. However, within the Port Authority, there are also a number of other areas which invoice customers and these areas also utilise the same invoicing system. Appendix 4 shows the data model for the invoicing system and also accompanying documentation including a version of each diagram drawn by the analyst for the invoicing system.

7.5 Preparation Step of Developing the BED

Some reflection had taken place since the original publication of the SDD in 1990. This had resulted in the identification a number of possible changes to the diagram. What follows is a discussion of the specific changes that were incorporated into the revised paper, the BED (appendix 1).

7.5.1 Event Focus (1P1)

The first change identified was to do with the realisation that the terms state and event had been used virtually interchangeably in the original paper (McDermid 1990a) and wrongly so. Though there is arguably a strong connection between state and event, they clearly represent two different ontological aspects of a system. An event occurs when the state of an entity changes. A state is instantiated when an event takes place. Having considered this for some time, I had concluded that the boxes in entity life histories (ELHs) ought to represent events. However, states are also shown on ELHs by the use of state indicators. Indeed, both pre- and post- states can be shown (Downs et al. 1988). It was decided that modelling events more closely reflected what the original diagram had attempted to model and that the paper should be rewritten to remove any confusion between state and event in this regard. Also to reflect this focus, the diagram was renamed the Business Event Diagram (BED). The rewriting of the paper in this respect presented no problems.

7.5.2 Object-oriented basis (1P2)

At this time another change was introduced. This was to update the paper in terms of recent developments in object-orientation. I saw no reason why the starting point of constructing the BED could not be an object model as opposed to a data model especially if it modelled only simple objects instead of entities. In fact, in terms of their information bearing capability, they are virtually identical. Coad and Yourdon's approach (1991) is an example of an approach which uses simple objects. So the paper was rewritten to include the option of starting either from a data model or from an object model using simple objects.

7.5.3 Group Event Notation (1P3)

A third change occurred to me while rewriting the paper. In the original paper, some discussion had taken place on delay analysis. The purpose of delay analysis was to identify whether there need be no time delay between certain events (eg between deleting an order header and its order lines). It was considered that this would be a useful aspect to have identified when it came to creating transactions in the information system. However, in the original paper, this grouping of events was not depicted graphically. It occurred to me that this could be shown graphically quite easily by bundling 'simultaneous' events into the same bubble. Of course strictly such events would not occur simultaneously. In practice any two events in a computerised information system would have to be implemented in a serial manner, but they could be considered at this level of abstraction to occur as part of the same event. Henceforth the reader should assume this interpretation of simultaneity in later discussion. The label used to describe this phenomenon is a simultaneous event and rewriting the paper was a straightforward task.

7.5.4 Delay Analysis Step (1P4)

Further, it was decided to extend the construction process to include a first-cut BED (ie before delay analysis) and then to proceed to a final BED (ie after delay analysis). This caused a modification to the description of the paper where it described the steps in the construction process and was therefore a minor straightforward amendment.

7.5.5 Event Notation Refinement (1P5)

Lastly, a change to the notation of the event symbol was introduced. In the original paper, a graphical distinction had been made between creation, modification and deletion by using different symbols to represent each of these. The source of the distinction had been the LSDM methodology (LSDM 1986). It was decided to remove this distinction in an attempt to reduce the notational complexity by having a single symbol for all events. Arguably, since the description of the event would indicate the type of event, no information would necessarily be lost. Again this caused no difficulty while rewriting the paper.

In summary, therefore the BED contained the following differences from the SDD. These were:

- a rewriting to better define the use of states and events (1P1),
- the inclusion of the alternative to construct the diagram from an object model rather than a data model (1P2),
- a modification to group 'simultaneous' events together (1P3),
- an extension to the construction process to include a first-cut model then final model (1P4) and
- a modification to the notation for the event symbol (1P5).

7.5.6 Evaluation of the Preparation Step against Intellectual Framework

In terms of reviewing the above changes against Wand and Weber's (W+W) framework, the following emerged. 1P1 and 1P5 were merely refinements or clarifications to the specification of the methodology M1 and therefore were not considered relevant for analysis here. 1P3, the group event notation issue represented ontological incompleteness. Here, I am suggesting that in the ontological world it is desirable to group together events which occur effectively simultaneously. However, there was no design construct for it in M1.

1P4 concerned adding another step to the methodology M1. This is dealt with straightforwardly using the script model. It was a simple matter to add an additional design world into the script model to show the first-cut BED step.

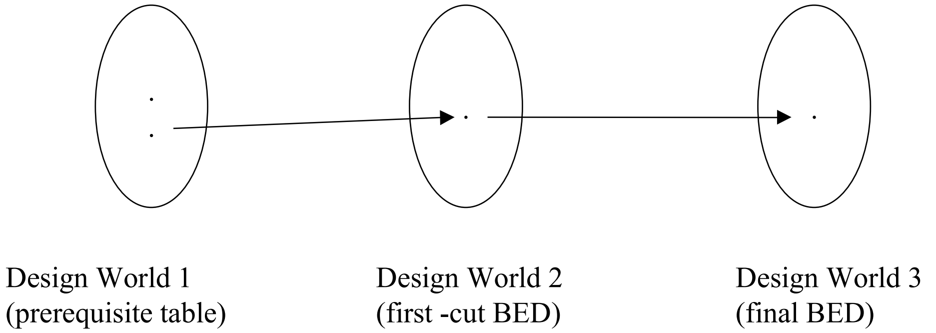


Figure 7.3 Revised Mapping showing Transitive Mappings

1P2 concerned developing the BED from an object-oriented model rather than an entity model. This was more problematic for the W+W model to handle. Figure 7.4 shows an attempt to model the problem here. The problem is that the W+W framework does not cater for the situation when two constructs are mapped onto a single construct (ie the life history in design world 2). In other words, there is nothing in figure 7.4 which indicates that one route is the alternative of the other. (Here, an ontological world is depicted instead of the real-world since it refers to a specific instance of the real-world in which entities and object are perceived constructs.)Of course it may be possible to extend the notation but as it stands, in terms of showing that alternative routes are possible, the W+W framework does not specifically accommodate this.

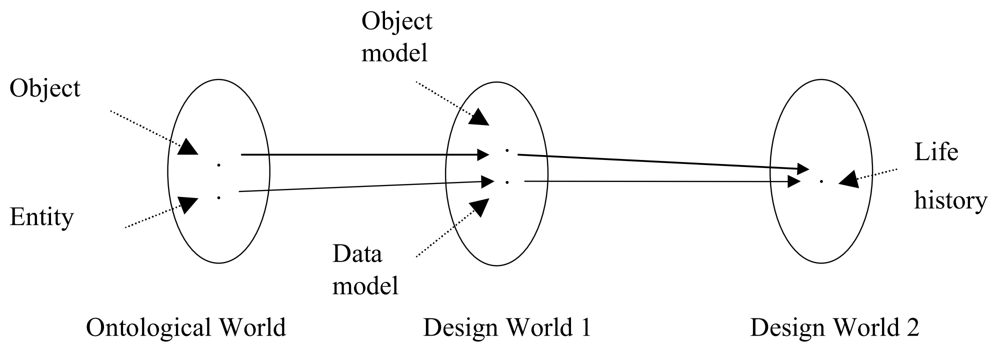


Figure 7.4 Attempt at Showing Alternative Construction Routes

7.6 Application Step of Developing the BED

Once the BED had been rewritten with the above changes, the paper was used by the analyst and myself as the basis for specifying the portbilling application. Table 7.2 shows the prerequisite table and figure 7.5 shows the BED for the portbilling application.

Event	Prerequisite
1 Process invoice	-
2 Release invoice for printing	1
3 Print invoice	2
4 Cancel invoice	[1,2]
5 Process portbill	18,16,13
6 Release portbill	[9,11]
7 Cancel portbill	5
8 Process portbill item	25,23,5(,26),22
9 Invoice portbill item	8,1
10 Transfer invoice to GL	3
11 Cancel portbill item	8
12 Request visit	-
13 Arrive in port	[-,12]
14 Depart from port	13
15 Cancel visit	12
16 Create client	-
17 Cancel client	[6,7]
18 Create portbill status	-
19 Create portbill audit	[5,6,7]
20 Create portbill item audit	[8,9,11]
21 Create portbill rate type	-
22 Create portbill rate	21
23 Create berth	-
24 Cancel berth	[9,11]
25 Create portbill item status	-
26 Create agent	-
27 Cancel agent	[9,11]

(Note that the notation for this table is explained shortly)

Table 7.2 Prerequisite Table for BED

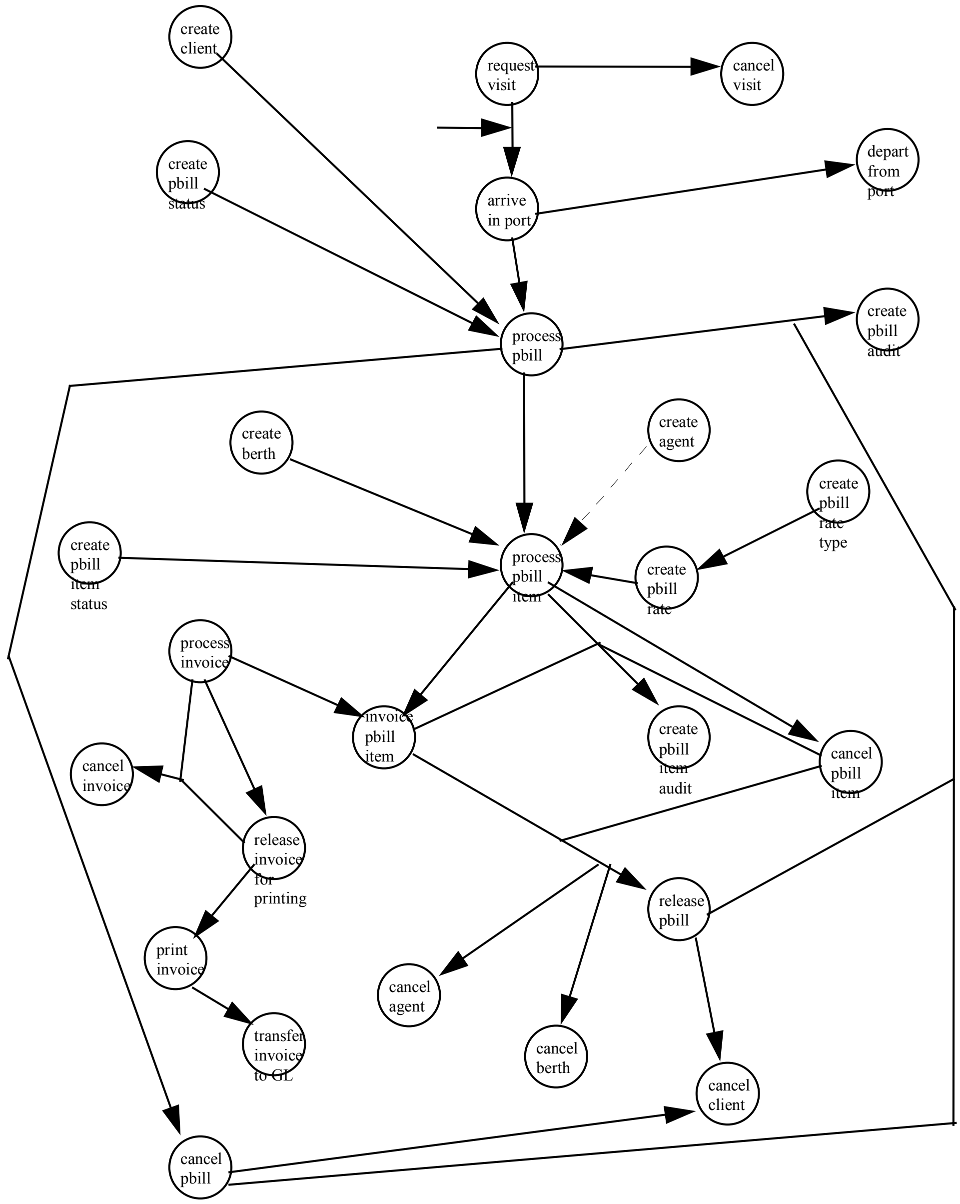


Figure 7.5 First-cut Business Event Diagram – Portbilling System

7.6.1 Flaw in Prerequisite Table Notation (1A1)

This was a problem which I did not detect during the preparation of the BED paper in appendix 1. In the original SDD paper as well as the BED, I had used a modification of De Marco's notation for selection (De Marco 1978). With a pure De Marco approach, a vertical line is inserted to indicate each option. For example [1 | 3] indicates either event 1 or event 3 is a prerequisite event but not both. I had modified the notation however to use a comma rather than a vertical line for example to [1, 3] because of the limitations of my word-processor at the time (ie when writing the SDD paper) and thought no more about it. However, as I worked on the prerequisite table here, I realised that the comma was being used in two ways. The first way was as a delimiter for selection as discussed above. The second way was as a delimiter to indicate ‘AND’. For example, the statement ‘2, 5’ means that both event 2 and event 5 are prerequisite events. The most straightforward solution here was to revert back to a pure De Marco notation by re-introducing the vertical line as the delimiter for selection.

7.6.2 The Graphical Implications of Union and Intersection (1A2)

Having realised the weakness inherent in the prerequisite table, a question then arose as to whether graphically there existed a similar weakness in the diagram. It turned out that graphical rules for showing union and intersection were also inadequate. Multiple direct dependency and selective dependency had been defined to describe intersection and union of events respectively. But what if a mixture of these situations were to occur? For example, suppose we have the following two situations A and B (figure 7.6). In situation A we have ‘[1 | 2], 3 leads to 4’ and in situation B we have ‘[(1,2) | 3] leads to 4’.

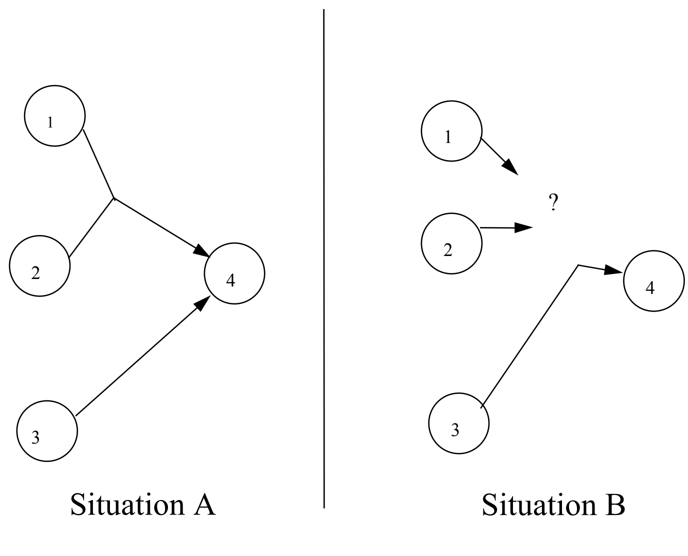


Figure 7.6 Two Heterogeneous Event Situations

In situation A, there is no problem and the existing notation is sufficient. In situation B however, the notation is inadequate since there is no construct for showing that the intersection of events 1 and 2 must take place before union with event 3. In a simple intersection situation, this was shown by physical separation of the arcs touching the event symbol, but in situation B the intersection must take place before the union and so the event symbol itself cannot be used. So a bar line was introduced to address this problem. See figure 7.7.

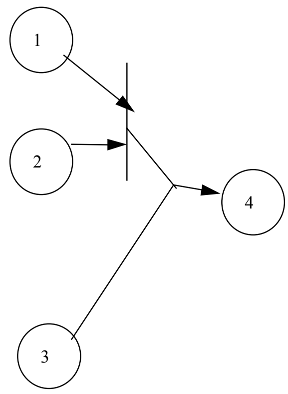


Figure 7.7 Situation B Resolved

7.6.3 Alternative Point of Entry Problem (1A3)

The 'arrive in port' event (13) in table 7.2 can occur either after a visit is requested or indeed without such a request. In the latter case, this may be because of some unexpected problem eg ship engine failure. In the order processing case study in appendix 7, there was no such example of that kind of situation and thus the problem of modelling it had not arisen. In the majority of cases at the edge of the BED, events start a train of subsequent events. Because these events have no predecessor, this is taken to mean that such events are initial events. In this case, since we wish to say that the event does not have a predecessor, there is a need to show this explicitly. The simple device of adding an arrow as shown in figure 7.5 solves this problem. The new arrow meets the arrow from the 'request visit' event indicating selective dependency. (For convenience sake, the solution to this is shown on figure 7.5

although it wasn't until the analysis step, ie working on the portbilling application that the problem emerged.)

Now, it may be argued that if an event may have a certain predecessor or not then there can be no constraint at all on the event. This is not correct. The notation is stipulating that for the event 'arrive in port' to occur then **either** that particular instance of visit object must not exist **or** previously a request to visit had been received. In other words, if for that instance, the event 'arrive in port' or 'depart from port' had already occurred, then it would not be acceptable to initiate the event 'arrive in port', ie non-existence of a state can be treated like any other state here.

7.6.4 Fan Out Dependency (1A4)

While drawing first drafts of the BED for the portbilling application, another improvement to the notation emerged, though in this case a minor one. In the portbilling system an agent may only be cancelled if there are no outstanding portbill items for that agent, ie all associated portbill items for the agent must be either cancelled or invoiced. Exactly the same rule existed for the somewhat unlikely event of cancelling a berth. This can be drawn as in figure 7.8(a). Alternatively to

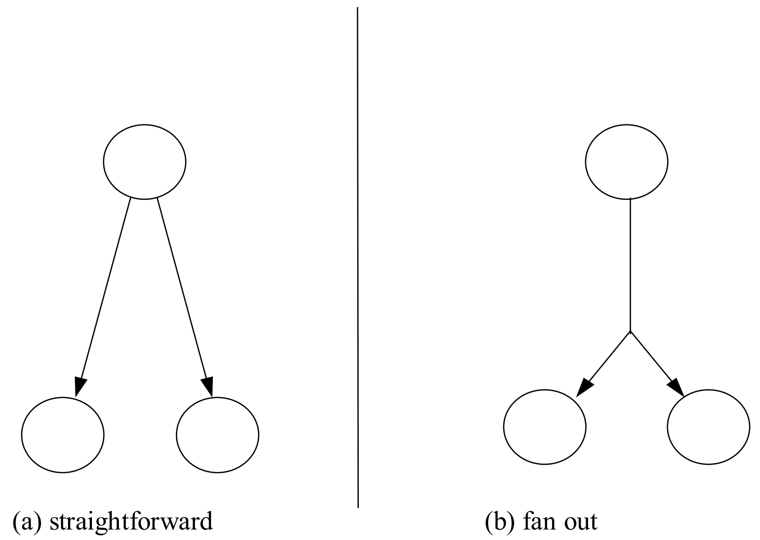


Figure 7.8 Straightforward Dependency and Fan Out Dependency

improve readability it can be drawn as in figure 7.8(b). This latter situation was termed fan out dependency. To make the diagram more presentable it was decided to incorporate fan out dependency as part of the notation.

7.6.5 Amendment to the Notation for Contingent Dependency (1A5)

The previous sections outlined a number of changes to the notation of the diagram. There was thus a feeling that the notation of the diagram was being tightened. To this end, instead of using a dashed line to indicate contingency, an arguably more intuitive notion borrowed from data modelling was introduced. See figure 7.9.

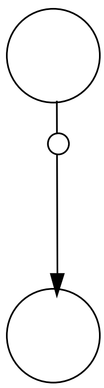


Figure 7.9 Contingent Dependency

In data modelling, the 'O' symbol on the line means optional, ie that an instance of the entity at that end of the arc may not exist. Here, the 'O' symbol would also indicate optionality, ie that the event at that end of the arc need not have taken place, but may have done. For example, if a credit note invoice needs to be deleted before an invoice object, contingent dependency would permit both the situation where no credit note instances exist as well as where one or more exist.

7.6.6 Simultaneity Problem (1A6)

Figure 7.10 shows the final BED. Note the existence of several large bubbles. The delay analysis revealed two so-called simultaneous situations. The first was that event ‘create portbill audit’ can be executed immediately after events ‘process portbill’, ‘release portbill’ and ‘cancel portbill’. This is because portbill audit keeps a record of all events on the portbill. The second situation concerned ‘create portbill item audit’ which had a similar purpose. In drawing the final BED, it was noticed that graphically there was a degree of repetition involved. For instance, ‘create portbill audit’ had to be repeated three times resulting in three group events. Since no immediate solution presented itself, this was noted for further consideration.

7.6.7 Evaluation of Application Step against Intellectual Framework

As regards reviewing the above changes against W+W’s framework, the following emerged. Issues 1A4 (fan out dependency) and 1A5 (amendment to the notation for contingent dependency) were minor notational changes and so do not directly concern us here. 1A1 concerned a flaw in the notation of the prerequisite table. In terms of W+W’s design constructs, the comma symbol had been used for two purposes. This is construct overload. Reverting back to De Marco’s original notation avoided the problem. With 1A2 (the graphical implications of union and intersection) the problem was one of ontological incompleteness. Here there were situations in the ontological world which could not be modelled in the design world.

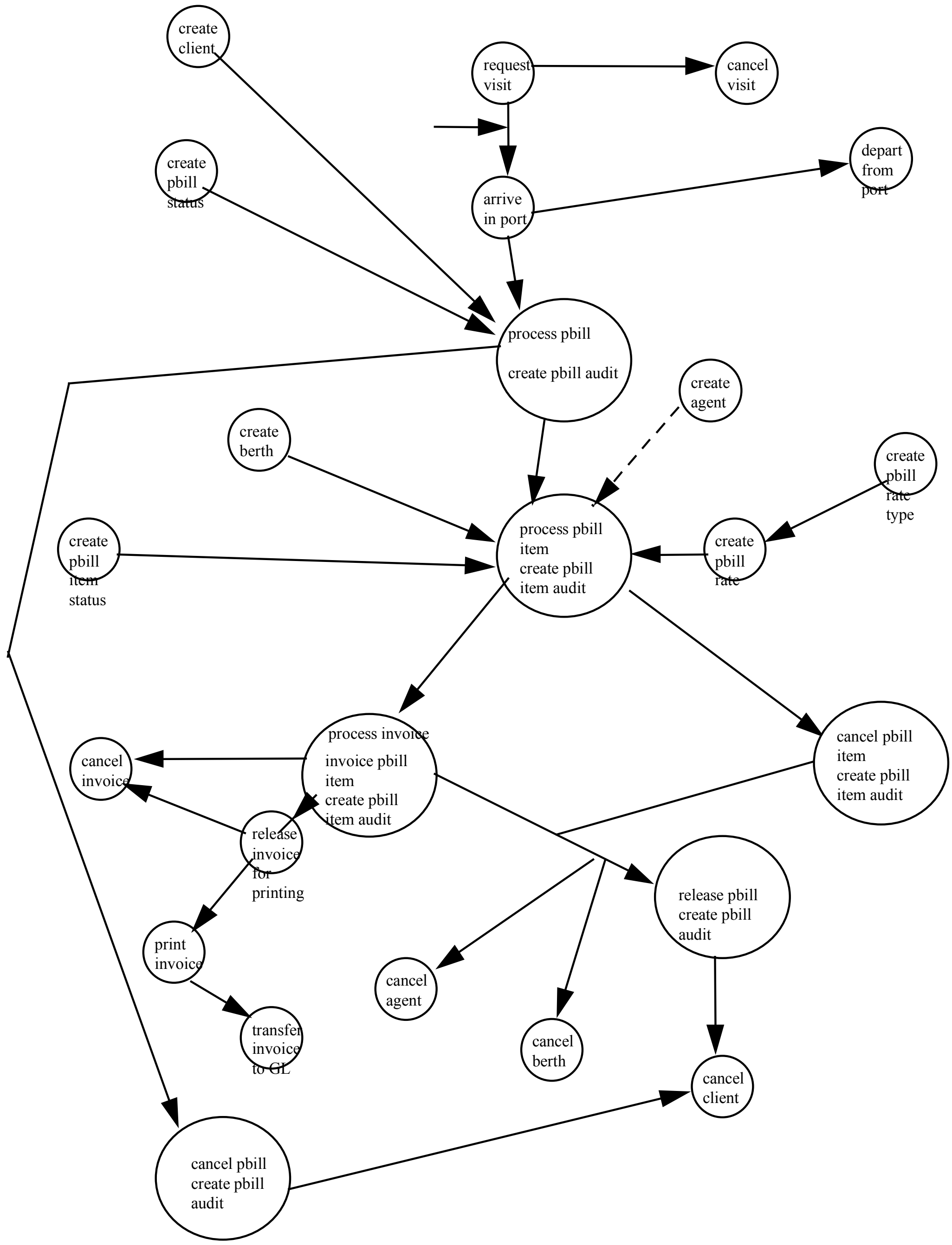


Figure 7.10 Final Business Event Diagram – Portbilling System

The introduction of a new symbol (bar) into the design world resolved the matter. Issue 1A3 (alternative point of entry problem) was also a matter of ontological incompleteness because the design world was not able to model a situation that had arisen in the ontological world (or more precisely had arisen in the real-world but not been anticipated in the ontological world). In the design world it was previously not possible to draw an arc unless it had come from some state. The last issue 1A6 (simultaneity) was also a matter of ontological incompleteness although perhaps more subtlely so. The design world was in fact able to model the situation in the ontological world. However, it was felt that this was not being done in an efficient manner with the result of having to repeat the events on the diagram. This can be argued to be ontologically incomplete in the sense that it may be considered that an appropriate design construct is missing, ie one which would facilitate such a mapping in a more direct and efficient way. Such a design construct would allow a direct mapping from the ontological world to the design world without the negative side-effect of repeated naming. However, at this juncture, I could see no obvious candidate for such a design construct.

7.7 Review Step of Developing the BED

7.7.1 Information Systems Events versus Business Events (1R1)

It is policy in the Port Authority information systems department that operational masters were to be used. This leads to a significant increase in the number of entities in the data model, an increase in the number of events in OLHs and increased complexity in the BED. With such overload, the question arises as to whether the benefits outweigh the costs. A field note refers to this.

> '*Portbill status and portbill item status contain status information for portbill and portbill item respectively. These I shall term status entities. Technically, this is sound as it can reduce redundancy in the data, but it does add significantly to the complexity of the diagram.*'

I observed that there was a dichotomy concerning the difference between what might be termed a business event and an information system event. A business event should reflect the essence of the business only, whereas I consider an information system event as an implementation of the business system. While it is perfectly reasonable for an organisation to implement its business systems by incorporating status entities as has the Port Authority, a fundamental question here concerned whether it was desirable to capture such 'implementation bound' information on a diagram such as the BED. (The information is 'implementation bound' because using a status entity is only one way of representing the phenomenon.)

This was a concern at this juncture because the aim of the BED was to assist a fuller specification of the system and so where organisations chose to include status entities in their data models, this would lead to a large, arguably overly-complex BED very quickly.

7.7.2 Need for Rule Completeness (1R2)

Although at this stage my thinking was fuzzy, there was a general concern that there was something missing in the specification. In later versions of the diagram it became clear to me that what was missing was a complete business rule. Two items from my field notes described my speculation at that time. The first item shows the first inkling of the idea that triggers might be part of this puzzle.

> *'Do we need to extend the prerequisite table to show triggering to the events? This might be especially helpful where the only prerequisite is a single event which is already on the ELH. There must be something that triggers such events.'*

And later this idea was refined.

> *'It is becoming clearer that the prerequisite table could be extended to include further detail. A possible set includes:*
> - *pre-conditions,*
> - *post-conditions,*
> - *time triggers and*
> - *who is responsible'.*

Further, an item from the analyst's interview also confirmed the need for rule (condition) based information.

> *'Could I draw your attention to this line here? I've called this transitive dependency, because we could argue that if we create the invoice header and then create the invoice line and then release the invoice header, in order for us to release the invoice header we've got already the invoice line as a prerequisite state and a prerequisite*

to that is that the invoice header is already created, so I wondered if we really need that extra dependency. Is it not already implied because invoice line exists and the only way invoice line exists is because invoice header exists?

(researcher)

Yes, when I drew this I actually had the **business rules** in mind. Suppose we don't have this line. It would mean quite differently that you release the invoice just by having the invoice line.

(analyst X)

But the only way we could have the invoice line was that we created the header anyway.

(researcher)

What I was also trying to capture was that the invoice line amount must be equal to the GL allocation amount and in this case the invoice header status must be processing. So it's **a combination of conditions** before we release the invoice header'.

(analyst X)

Although I was trying to clarify an issue concerning transitive dependency, the response that came back suggested that more information was needed than the diagram provided. In fact, the phrase 'combination of conditions' was actually used. However, note at the time of the interview the analyst had worked with later versions of the diagram and may have been influenced by the ideas implicit in those diagrams.

Nonetheless, the view that something was missing in the diagram was evident here and part of the general discussions taking place between myself and the analyst.

7.7.3 States versus Events (1R3)

Whilst a decision had been taken that the diagram should model events as the main focus of interest, the possibility of using states as the main focus had never been totally rejected. The popularity of the finite state diagram in many methodologies, was testimony to the potential of the idea. Though, for the most part, the duality between state and event had been taken for granted by myself, there were the beginnings of doubt over just how interchangeable these two constructs were. For example, in the portbilling system 'invoice status is unused' is a state that could be modelled, yet would there be an event to set invoice status to unused? Also, a field note illustrates a more pressing doubt which went back to the time of the publication of the original SDD paper.

> *'The SDD paper catered for the possibility of different business events leading to the same state (eg setup first order, setup standard order). These were shown as separate event boxes on the SDD. A careful review is required to assess whether separate event boxes are necessary or desirable and if so, what does this mean for the relationship between state and event? For instance, can separate arrows be used, each arrow representing discrete business events? Nodes would therefore denote states.'*

Having now created a doubt in my mind about the whole relationship between state and event, one obvious way forward was to consider recasting the diagram in terms of states rather than events and then to evaluate the differences, ie to use the cyclic character of action research to test this assertion.

One final observation on this issue was obtained from the analyst interview.

> *'One last question about the OLHs is that although these are events and we've written them down in the table as events, on the diagram, we tend to be using terms in the past tense such as 'invoice status unused', 'invoice line updated'. That's more a state in my interpretation.*
>
> (researcher)
>
> Those should be states rather than events.
>
> (analyst X)
>
> *Do you find it's easier to think in terms of states rather than events?*
>
> (researcher)
>
> I was trying to get the states first then to the events.'
>
> (analyst X)

Again, I was trying to establish whether the analyst preferred to work in states rather than events. The response that perhaps finding the states first and then through states the events, led me to speculate that states might be a more preferable starting point for analysts.

7.7.4 Evaluation of Review Step against Intellectual Framework

In terms of reviewing W+W's framework against the above issues, the following emerged. 1R2 (need for rule completeness) was suggesting ontological incompleteness. This was a good example of how the framework might begin to support and inform the research activity. However, it has to be said that although the framework explained the problem in terms of mappings between sets, it didn't in itself suggest how to resolve such problems. Issue 1R3 (states versus events) is not classified in terms of W+W's framework as such, although I do not consider this problematic. Issue 1R3 is in fact implying the need to change the nature of the ontological world from one which focuses on events to one which focuses on states. One way of describing issue 1R1 (information systems events versus business events) would be construct overload because the same design construct was being used to model two ontologically distinct entities, ie information systems events and business events. However, perhaps a better way of looking at this was to view information system events as being modelled in a later design world to business events. In other words, it was speculated that capturing information systems events could be delayed to a later stage. See figure 7.11. In the end, I did not follow this up until a later stage in the research. However, it illustrates well how using an intellectual framework might begin to structure one's thinking and assist in solving specific problems.

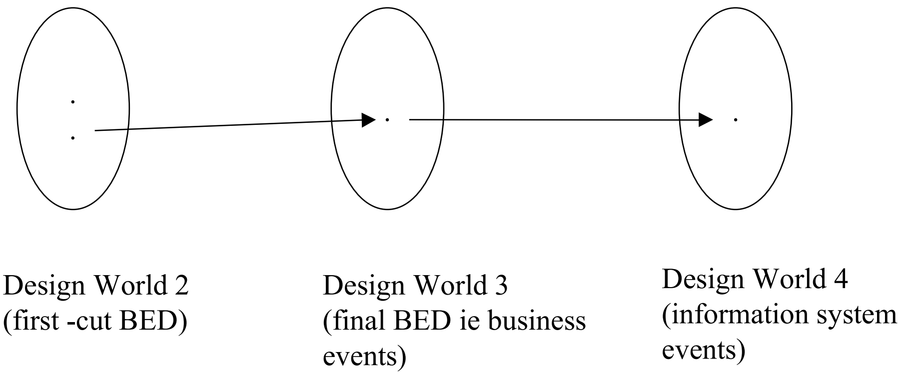


Figure 7.11 Separation of Business and Information System Events

7.7.5 Discussion in terms of Wider Literature

The purpose of this section is to relate the developments in this chapter back to the literature at large. This is an important part of the reflective process underpinning action research as it seeks to confirm or otherwise the developments or improvements being suggested at the time when they are being asserted within the research process. Though the literature review has identified many viewpoints about other approaches to business rules modelling, relatively little literature is available which can be used to guide the designer of notations. However, one model is available which defines a list of constructs considered necessary for describing the real world. This is again by Wand and Weber, though a later paper describes this more fully (Wand and Weber 1995). Table 7.3 describes the constructs in their representation model.

A major problem in conducting this review however, is to recognise that the diagram is not offered as a complete representation of all aspects of an information system. Rather, the question addressed is to design a diagram suitable for describing and discussing business rules in the context of IRD and so arguably a subset of a complete representation model is more appropriate, though clearly the key is in defining what constructs are the most helpful at this level.

In terms of the representation model in table 7.3, the main construct in the BED is the event. Therefore event, event space and lawful event space all have some meaning in the context of the BED. Other related constructs are external event, internal event, well-defined event and poorly defined event. These relate essentially to distinguishing between what happens inside and outside of the system. In connection with the Portbilling system some events may be classified as internal, eg create portbill rate type and others as external such as request visit. Clearly the earlier discussion distinguishing information systems events and business events may have some overlap here.

Perhaps a more notable observation is to observe how grounded their representation model is in terms of states. Particularly, many other constructs are defined in terms of states. This would therefore support a case for considering to model in states.

Ontological Construct	**Explanation**
Thing	The elementary unit in their composition model. The real world is made up of things. A composite thing may be made up of other things (composite or primitive).
Property	Things possess properties. A property is modelled via a function that maps the thing into some value. A property of a composite thing that belongs to a component thing is called a hereditary property. Otherwise it is called an emergent property. A property that is inherently a property of an individual thing is called an intrinsic property. A property that is meaningful only in the context of two or more things is called a mutual or relational property.
State	The vector of values for all property functions of a thing.
Conceivable state space	The set of all states that the thing might ever assume.
State law	Restricts the values of the property functions of a thing to a subset that is deemed lawful because of natural or human laws.
Lawful state space	The set of states of a thing that comply with the state laws of the thing. It is usually a proper subset of the conceivable state space.
Event	A change of state of a thing. It is effected via a transformation.
Event space	The set of all possible events that can occur in the thing.
Transformation	A mapping from a domain comprising states to a codomain comprising states.
Lawful transformation	Defines which events in a thing are lawful.
Lawful event space	The set of all events in a thing that are lawful.
History	The chronologically ordered states that a thing traverses.
Coupling	A thing acts on another thing if the existence affects the history of the other thing. The two things are said to be coupled or interact.

System	A set of things is a system if, for any bi-partitioning of the set, couplings exist among things in the two subsets.
System composition	The things in the subsystem.
System environment	Things that are not in the system but interact with things in the system.
System structure	The set of couplings that exist among things in the system and things in the environment of the system.
Subsystem	A system whose composition and structure are subsets of the composition and structure of another system.
System decomposition	A set of subsystems such that every component in the system is either one of the subsystems in the decomposition or is included in the composition of one of the subsystems in the decomposition.
Level structure	Defines a partial order over the subsystems in a decomposition to show which subsystems are components of other subsystems or the system itself.
Stable state	A state in which a thing, subsystem or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event).
Unstable state	A state that will be changed into another state by virtue of the action of transformation in the system.
External event	An event that arises in a thing, subsystem or system by virtue of the action of some thing in the environment on the thing, subsystem or system. The before-state of an external event is always stable. The after-state may be stable or unstable.
Internal event	An event that arises in a thing, subsystem or system by virtue of lawful transformation in the thing, subsystem or system. The before-state of an internal event is always unstable. The after-state may be stable or unstable.
Well-defined event	An event in which the subsequent state can always be predicted given the prior state is known.
Poorly defined event	An event in which the subsequent state cannot be predicted given the prior state is known.

Class	A set of things that possess a common property.
Kind	A set of things that possess two or more common properties.

Table 7.3 Bunge-Wand-Weber (BWW) Representation Model, after Wand and Weber (1995)

7.8 Decision Step of Developing the BED

7.8.1 Change Notation (1D1)

The specific problems encountered in applying the BED to the business application led to the decision that a number of improvements to the notation for the next version of the diagram should be considered. These were discussed above under:

- a flaw in the prerequisite table (1A1),
- indicating alternative point of entry (1A2),
- graphical implications of union and intersection (1A3),
- fan out dependency (1A4),
- the notation for contingent dependency (1A5) and
- simultaneity anomalies (1A6).

In all cases except 1A6, a solution was implemented in the next version of the diagram (1D1).

7.8.2 Change to State (1D2)

At this time it was decided to implement the next version of the diagram using states rather than events as the main modelling focus. The discussion in the review section had raised sufficient concern in my mind over this issue. Testing this concern by this means seemed a good way of exploring this issue further (and illustrates the adaptability of the action research approach).

7.9 Summary of Progress with the BED

A number of issues were identified during the preparation stage. Four of these were resolved satisfactorily during the preparation stage itself. These were:

- event focus (1P1),
- object-oriented focus (1P2),
- delay analysis step (1P4) and
- refinement of the event notation (1P5).

During the application stage a number of additional issues were identified. Most of these were notational issues and resulted in a decision to change the notation in the next version of the diagram. These application issues were:

- prerequisite flaw (1A1),
- union and intersection (1A2),
- alternative point of entry (1A3),
- fan out dependency (1A4) and
- contingent dependency (1A5).

The addition of the group event notation (1P3) caused difficulty in relation to simultaneity (1A6) and remained an unresolved issue at this point.

In the review stage, three issues were identified. Two were flagged for further consideration. These were:

- information systems events versus business events (1R1) and
- rule completeness (1R2).

The other issue was concerned with whether states should be modelled instead of events (1R3) and arose as a consequence of the event focus (1P1). Overall, therefore two decisions were taken as a result of working on the BED. These were:

- to change the notation of the diagram in a number of respects (1D1) and
- to model the next version of the diagram in states instead of events (1D2).

Chapter 8

Action Research Study 1

The Business State Diagram version 1

8.1 Overview

The work on the BED (Business Event Diagram) had resulted in identifying a number of changes. The most significant change was the shift from event to state as the main unit of modelling. Because of this shift in modelling, the question also arose as to whether the other changes identified would still be appropriate or indeed feasible. For example, with the BED simultaneous events were grouped together in a single event. With the change to modelling states, there would now be no event construct in the ontological world within which to group simultaneous events!

The same cyclic approach to the research was retained. This initially involved two steps, ie preparing the BSDv1 (Business State Diagram version 1) paper and then applying it the portbilling application. In rewriting the description of the diagram, the original case study was used (appendix 7). This gave me the opportunity to test such a fundamental shift in modelling emphasis on a different case before working on the portbilling application. The cyclic approach continued with a review of the success of changes instigated in the previous iteration together with any issues which emerged.

8.2 Preparation Step of the Development of the BSDv1

8.2.1 The Simultaneity Problem (2P1)

With the move to modelling in states I had a concern that there may now be no good means of showing simultaneity. Grouping simultaneous states from different objects together in the same box (as had been done with events) was not intuitively attractive in the sense that it was the simultaneity of event and not state that had been the focus previously. The states which were brought about by the simultaneous events were a consequence of the events and semantically distinct from the events. However it was possible to show simultaneity between states with a double-headed arrow as illustrated in figure 6 of the BSDv1 in appendix 1. This had the advantage of being able to show more accurately the logic of dependency between states. This was something not possible with a list of events or states. Indeed a list can be misleading because it implies a linear sequence. In figure 6 a cluster of states surrounding the 'order header deleted' state is connected by double-headed arrows. Here something of the sequence of execution to preserve integrity can be inferred, ie that all credit notes, all payments and all order lines must be deleted before the order header can be deleted.

The interview with the analyst also revealed support for the need to show simultaneity.

> *'One of the differences was that there was a double headed arrow which shows simultaneity. Any comments on the construction process generally?*
>
> (researcher)
>
> For the double-headed arrow, I can recall that we have a database transaction process, we do have to deal with two or more objects at the same time to complete one transaction. If one fails we roll back the whole transaction. So things have to happen simultaneously.'
>
> (analyst X)

In summary it was decided to introduce the double-headed arrow as a means of showing simultaneity and this was written into the case study description for the BSDv1.

8.2.2 Evaluation of the Preparation Step against Intellectual Framework

With regard to reviewing Wand and Weber's (W+W) framework in terms of the above preparation issue, 2P1 (the simultaneity problem), we have ontological incompleteness. The switch to modelling states rather than events meant that no suitable design construct was available to capture simultaneity. The introduction of a new design construct, the double-headed arrow was seen as overcoming this inadequacy.

8.3 Application Step of the Development of the BSDv1

8.3.1 The Simultaneity Problem (2A1)

The construction of the BSDv1 followed similar lines to that of the BED. Indeed, the first-cut version of the diagram was largely a mirror image of the BED with states replacing events. The delay analysis proved problematic however in that the notation was ineffective. For example the 'portbill audit created' state is to be shown only once yet it is to be linked with three separate states, ie 'portbill in processing', 'portbill released' and portbill cancelled'. See table 8.1 and figure 8.1. Topologically, the double-headed arrows between these affected states are actually stipulating that all four states are related simultaneously which is not the case here! To avoid this, the 'portbill audit created' state would have to be drawn three times, once for each individual simultaneous connection. This was in principle the same solution that had been applied in the BED. This problem was identified as one which needed further examination and resolution.

State	Prerequisite
1 Invoice in processing	-
2 Invoice released	1
3 Invoice printed	2
4 Invoice cancelled	[1|2]
5 Portbill in processing	18,16,13
6 Portbill released	[9|11|9,11]
7 Portbill cancelled	5
8 Portbill item in processing	25,23,5(,26),22
9 Portbill item invoiced	8,1
10 Invoice transferred to GL	3
11 Portbill item cancelled	8
12 Visit requested	-
13 Visit in port	[-|12]
14 Visit departed	13
15 Visit cancelled	12
16 Client created	-
17 Client cancelled	[6|7|6,7]
18 Portbill status created	-
19 Portbill audit created	[5|6|7]
20 Portbill item audit created	[8|9|11]
21 Portbill rate type created	-
22 Portbill rate created	21
23 Berth created	-
24 Berth cancelled	[9|11|9,11]
25 Portbill item status created	-
26 Agent created	-
27 Agent cancelled	[9|11|9,11]

Table 8.1 Prerequisite Table for BSDv1

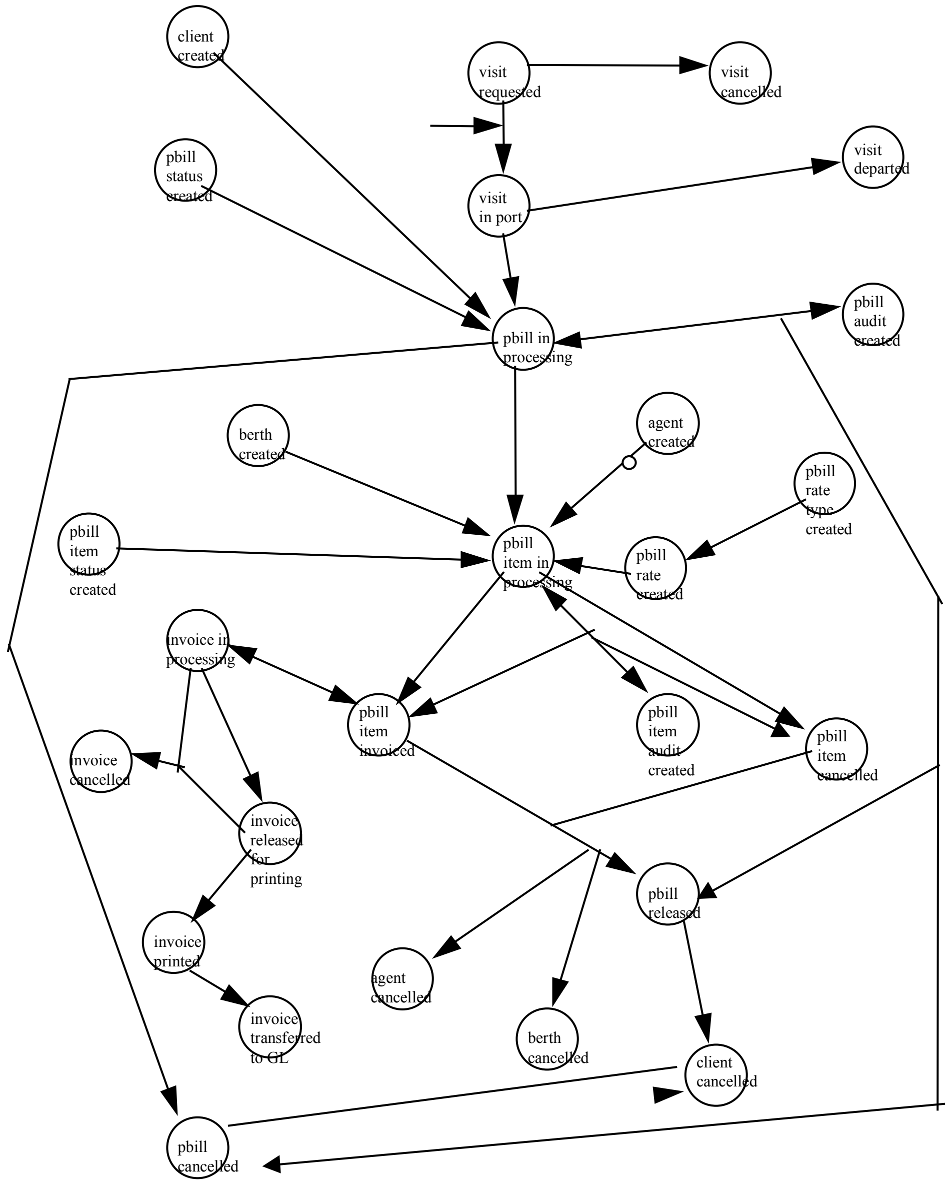


Figure 8.1 Business State Diagram version 1 – Portbilling System

8.3.2 The Overkill Problem (2A2)

I coined the term 'overkill problem' to refer to a potential pitfall in conceptual modelling. The overkill problem arises if the OLHs are modelled incorrectly in the first instance. In its simplest form overkill is the identification of duplicate redundant states in different objects. For example, an invoice header might be identified as overdue and thus acquire overdue status. If each individual line of the invoice (ie the invoice-line object as opposed to the invoice-header object) is also given an overdue state, then this would be deemed overkill.

The overkill issue, ie the need to avoid overkill, was something the analyst had encountered in the Port Authority and had also been discussed when applying the diagram to the business application.

> 'In our situation at the Port Authority, we generally don't remove a record, like order or invoice when it's no longer valid or active. It's normally set into a certain state like cancel. So when we cancel an invoice the system will so view the invoice lines as being cancelled as well, because it's the master detail of the relationship. So we deal only with the header and not with the lines because the lines carry the state of the header. It inherits the state. So you don't have two events. You have only one event when you delete or cancel an invoice. You set the status of the header to that particular state - cancel.'
>
> (analyst X)

The above discussion illustrates the benefit of looking across object boundaries. This was a main argument of the original SDD paper. The nature of OLH modelling is that each object is considered in its own right without considering what has been modelled elsewhere. This is one major reason why overkill is likely to occur. Since states are the main modelling unit here, it is not surprising that the issue of overkill (ie unnecessary duplication of states) emerges with the BSDv1. However, identifying overkill is only the first step in resolving the problem. Later a proposal is made which seeks to avoid overkill altogether.

8.3.3 Evaluation of Application Step against Intellectual Framework

With regard to reviewing the above issues against the W+W framework, the following emerged. In issue 2A1 (the simultaneity problem) ontological incompleteness was detected. In fact this was the second time this problem had occurred since when modelling events a similar phenomenon occurred under 1A6. Again, it was postulated that by introducing a new design construct the problem would be eliminated, though no candidate solution presented itself at this point. The second issue 2A2 (overkill) was strictly not a problem in terms of the framework, ie a mapping from the ontological world to the design world existed. However, the concern was that in practice analysts might not follow this correctly. While this issue was dealt with satisfactorily within the W+W framework, there was no mechanism within the framework which assisted in determining how easy or difficult it might be for analysts to successfully implement mappings between the ontological and design worlds.

8.4 Review Step of the Development of the BSDv2

With the exception of one problem, the notational changes (1D1) were successful and posed no problem with either the case study or the portbilling application. The problem was with the prerequisite table flaw. In reviewing the construction of the prerequisite table for the portbilling application, I noticed that I had incorrectly defined some of the selections. For example, state 17 (client cancelled) can only be achieved if there are no live portbills for that client. The test therefore is to check for portbills whose state is either released or cancelled and all such portbills associated with that client must be in one or other of those states. The possible permutations are that all portbills for that client are cancelled, that all are released or that a combination of only cancelled and released portbills exists. Since the selection construct is exclusive then this last possibility is not catered for. This was corrected in table 8.1 by adding a third clause in the selection construct. It can be seen that this was not so much a problem with the notation but rather the way the notation was being expressed. Alternatively an additional notational construct could be devised to indicate an 'inclusive or'. However, while considering the implications of this, it occurred to me that one important facet of this situation is that many instances of the portbill object are being checked in order to change one instance of client object. I considered this to be a useful piece of information to show explicitly and deal with this shortly under the heading of the 'manyness' problem.

As far as the change to modelling in states was concerned, three aspects emerged at this time and these are now discussed. Firstly, the change to modelling in states was tested in two applications. These were in the case study and then in the portbilling application. Both tests were successful to the extent that each problem domain could be modelled in states. However, the case study required some changes to be made in what was being modelled whereas the BSDv1 for the portbilling application was virtually a mirror image of the earlier BED. With regard to the case study, changes had resulted because of differences in the way that events and states were modelled in the OLHs (object life histories) for order header and order line. Please consult figures 3 in both the BED and the BSDv1 in appendix 1. The reason for this difference is to do with whether the modelling focus is event or state. With an event focus, two different events, eg of normal setup and outstanding setup may take place. This is reflected in the OLH for order header in the BED. With a state focus all that matters is the state achieved, not how it got there, hence the slightly more simplified OLH in the BSDv1. With regard to the portbilling application no such differences emerged between the event focus and the state focus. This can be seen by comparing the two diagrams (ie figure 7.5 for the BED and figure 8.1 the BSDv1). Overall, although some business application dependent differences emerged it was still felt that modelling in states was successful.

Secondly, in terms of ease of construction, there was a feeling that states were more straightforward. The following is an extract from the interview with the analyst.

> *'Just connecting up with that last observation, as I read your diagram here I felt that the states here were far more focused. You know you've said GL account is active. That's a state for - it's very much a state. I don't know whether you can give me a comment as to whether you find modelling in states better or worse than modelling in events which is what was being done on the previous diagram.*
>
> (researcher)
>
> I feel more comfortable with the state. State is very precise. When you draw events you tend to bring in the rules. You try to put in too much into the diagram itself. The state is really clear, straightforward. More straightforward than the events.'
>
> (analyst X)

Thirdly, from earlier discussion, it would appear that modelling in states might make it easier to identify overkill and thus avoid such pitfalls. So, overall the balance of argument at this point seemed to favour state over event. However three specific additional issues were identified.

8.4.1 Readability (2R1)

While on balance, it may be said that modelling in states achieved its goal, there was still much that was not achieved through this change. Looking at both diagrams, they remained large and clumsy, especially with the inclusion of status entities. The number of arcs interconnecting states is large and the diagram overall is unwieldy. In other words, the BSDv1 was not particularly readable. For instance, if there was a way of making it more readable perhaps by reducing the number of states that have to be modelled, that would have been an improvement. So in summary although the move to states was successful, it did not in itself resolve the problem of readability of the diagram.

8.4.2 The Need for Rule Completeness (2R2)

In a similar vein the need for a more complete picture of rules became evident here too. For example phrases from the analyst's interview refer to the need for rules or conditions. Observe the use of the word 'condition' or 'rule' in the following excerpts.

> 'So when certain states or **conditions** exist in the object, it triggers off the event.... It helps in a better understanding of how processes take place'
>
> '... where it says you can only update this object when the state or the **condition** is such and such'
>
> 'when you draw events you tend to bring in the **rules**'
>
> (analyst X)

The problem would appear to be that whether you are modelling with states only or events only, there is insufficient information to fully describe a situation. The analyst summed up his frustration (with regard to the BED) in the following sentence.

> ‘what I was trying to capture was that the invoice line amount must be equal to the GL allocation amount....’.
>
> (analyst X)

Such rules were not captured with either states or events in themselves. It was becoming clearer that there were other kinds of conditions not directly connected with state precedence that were also criteria for firing the event or changing the state. Two examples from the case study illustrate this. Firstly, for an order to be accepted, the customer’s credit balance must be greater than the value of the order. So, it is not sufficient only for the customer to be in good status. The customer must also have sufficient funds. In constructing the prerequisite table in particular this omission became quite evident. The second example concerns the checking of stock. Here in addition to the stock item existing which is the state that was being modelled currently, there must be sufficient stock to cover the order. Again, this was not being captured in either of the diagrams. So there was a growing feeling that more information was required to completely specify a state change or for that matter an event.

8.4.3 Manyness (2R3)

There was another aspect of rule completeness which was not being captured. I shall refer to this as the ‘manyness problem’ and deal with it separately here. It is to do with whether a single instance or a number of instances of particular object take part in an event. For example where stock is being checked as part of accepting an order, it is likely that a number of instances of the product object will have to be checked, ie one for each line on the order. Yet, only one order and therefore one instance of state of ‘order created’ will result from this. On other occasions only one instance of a particular object will contribute to an event. The problem is that neither the prerequisite table nor the diagram has any mechanism for showing manyness, ie that many instances may contribute to one event. Arguably the ability to indicate manyness could be seen as a desirable feature in such a diagram and is thus worthy of consideration here.

8.4.4 Evaluation of Review Step against Intellectual Framework

In terms of reviewing the W+W framework against the above issues the following emerged. Issue 2R2 (the need for rule completeness) and 2R3 (manyness) were straightforward examples of ontological incompleteness. On the other hand, issue 2R1 (readability) represented an issue apparently outside of the scope of the W+W framework since mappings between sets of constructs do not provide any sense of whether a diagram (ie an artefact of the design world) would be perceived as easy or difficult to read.

8.4.5 Discussion in terms of Wider Literature

In terms of the representation model of table 7.3, since states were now being modelled, the constructs of state, conceivable state space, stable state and unstable state has some meaning. Interestingly, the idea of a stable state had some similarity to what I had termed a significant state, ie one which could be observed by some agent outside the system.

However, though states themselves appeared to be a useful basis for modelling, using states only did not appear to address other issues raised in this chapter concerning for example rule completeness (2R2).

8.5 Decision Step of the Development of the BSDv2

8.5.1 Extensions to the Prerequisite Table (2D1)

It was decided to add two additional columns to the prerequisite table. These were called the pre-condition column and the post-condition column. The pre-condition column contains a description of the condition or conditions which must pertain for the state to come into being. Such conditions are in addition to any prerequisite states. So, for example, in the case study, a pre-condition to the 'paid invoice' state is that the payment value must be equal to the order value after the value of any credit notes has been subtracted. The post-condition column contains indicators which show either simultaneity with other states or time delays such as 'there is a 24 hour delay before the invoice is printed'. The extensions to the prerequisite table are therefore designed to assist the rule completeness problem (2R2) and the simultaneity problem (2A1).

8.5.2 Complex Objects (2D2)

Complex objects allow simple objects to be grouped in more meaningful ways. For instance the complex order object could be defined to contain two simple objects, ie order-header and (many) order-line(s). Complex objects allow a simpler and more straightforward mapping from the ontological world. For example, to create the complex order object, only one event is required as opposed to many events if simple object counterparts were used. Thus, this is argued to address the overkill problem (2A2) and also to aid readability of the diagram since the number of states in a diagram are likely to reduce (2R1).

8.5.3 The Harel blob (2D3)

The Harel blob, as explained in chapter 2 serves to encapsulate groups of constructs so that they may be treated in the same manner. The number of lines or arcs in a diagram reduces since arcs touching the edge of the Harel blob indicate connection to all states within that blob. Thus readability of the diagram is arguably improved (2R1).

8.5.4 Evaluation of Decision Step against Intellectual Framework

In terms of the W+W framework, the above decisions were interpreted as follows. 2D1 (extensions to the prerequisite table) address the ontological incompleteness problem identified earlier. Pre- and post-conditions in the table represented new design constructs and therefore eliminated ontological incompleteness in terms of these specific areas. The decision to implement complex objects (2D2) was an interesting one in terms of the W+W scheme because strictly there wasn't a problem with the framework here. However, the candidate solution involved introducing a new construct into both the design world and the ontological world. The reasoning for this was as follows. In the original ontological world only simple objects were considered to exist and thus the risk of overkill arguably increased. By introducing complex objects into the ontological world and design world, the integrity of the mapping was preserved, yet at the same time the likelihood of overkill was reduced. Introducing the Harel blob (2D3) into the design world (again for reasons of readability) created construct excess and also what Wand and Weber (1993) call 'some thorny issues'. The problem is with encapsulation. Suppose a thing is made up of parts. In mapping between the ontological world and the design world there is arguably redundancy if both the thing and its parts are mapped, since by definition, the sum of the parts represents the whole. However, ontologically speaking, if it is required to record information about the relationship between the whole and its parts, both constructs are needed. In summary the W+W framework accommodated (and predicted) this kind of situation.

8.6 Summary of Progress with the BSDv1

In summary, it is concluded that:

- the notational changes proposed in the previous section were successful (1D1),
- the change to states from events as the main modelling unit was successful (1D2) though it didn't resolve other problems common to both the BED and the BSDv1,
- a way to avoid overkill should be developed (2A2),
- the simultaneity problem remains (2A1),
- readability of the diagram is a concern (2R1),
- the need for more complete information on rules is a frustration in working with the diagram (2R2) and
- the ability to show manyness is considered a desirable feature (2R3).

The following candidate solutions have been proposed. These are:

- extending the prerequisite table (2D1),
- introducing the complex object (2D2) and

- incorporating the Harel blob into the diagram (2D3).

Collectively, these solutions are intended to address:

- readability (2R1),
- simultaneity (2A1),
- the overkill problem (2A2) and
- rule completeness (2A2).

Currently, this leaves the manyness problem (2R3) and the question of modelling information system events as opposed to business events (1R1) without any immediate solution.

Chapter 9

Action Research Study 1

The Business State Diagram version 2

9.1 Overview

By this stage of the study there was a great deal of interaction between the analyst and myself. With previous versions of the diagram, I had written the description of the version of the diagram prior to sessions with the analyst, but by now I was quite familiar with the portbilling application, so it was natural for me to work interactively with the analyst in writing the description of this version of the diagram. Consequently, the description of the BSDv2 (Business State Diagram version 2) uses the portbilling application rather than the case study. This is why figure 7.2 shows no preparation columns for the BSDv2 and BRDv1 (Business Rules Diagram version 1). Otherwise the format of this chapter remains similar to that of previous chapters.

9.2 Application Step of the Development of the BSDv2

9.2.1 Readability (3A1)

Though this version of the diagram embodies three proposed solutions namely, the use of the Harel blob, the use of complex objects and extensions to the prerequisite table, we firstly redrew the BSDv1 by incorporating the Harel blob solution only (figure 9.1). This was unsuccessful in the sense that it created greater visual complexity rather than reducing it. From a topological perspective by positioning states close to other states within the same

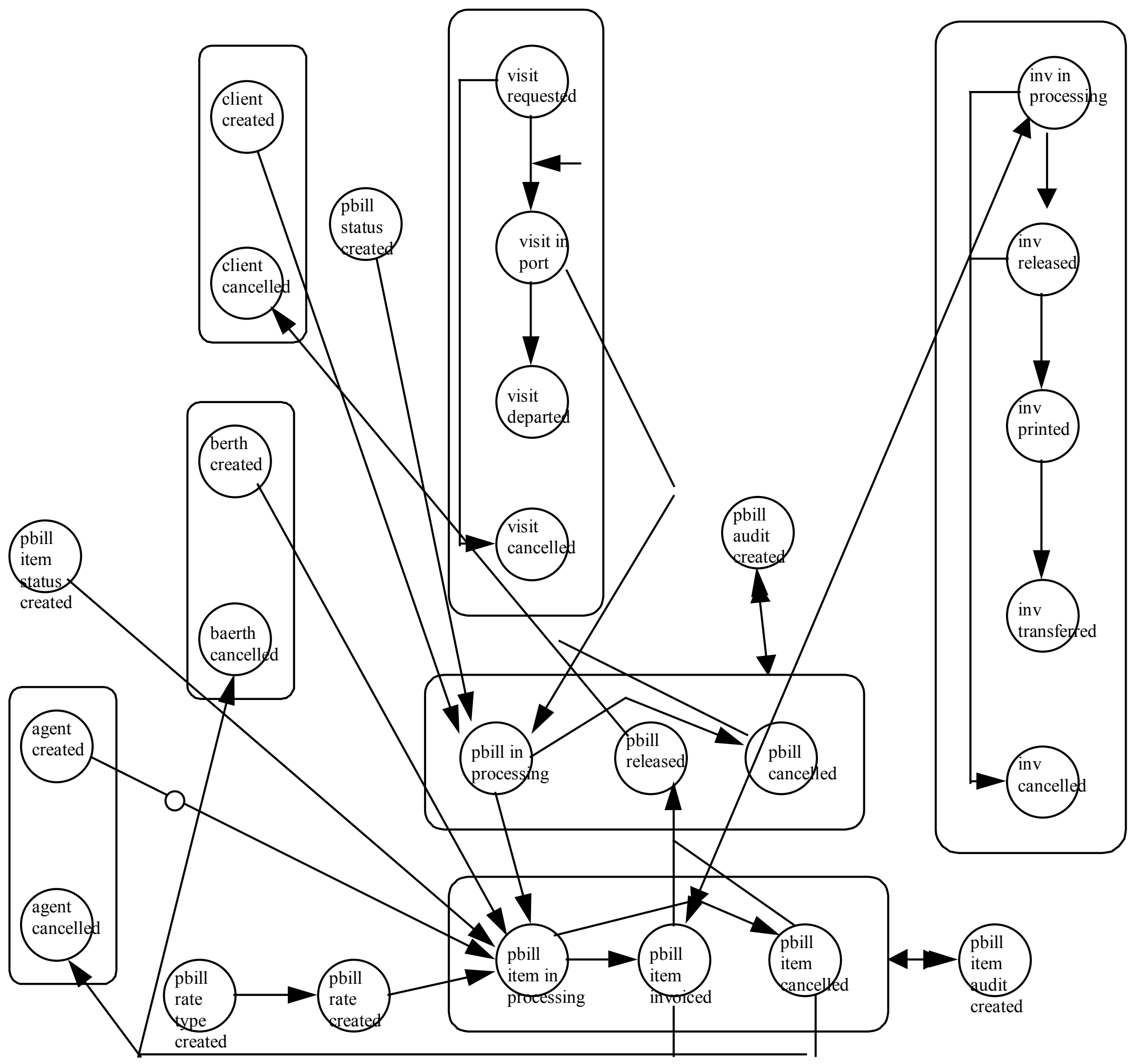


Figure 9.1

First Attempt at using the Harel Blob

simple object generated many arcs (ie dependencies) across objects and thus resulted in a visually complex diagram. The addition of the Harel blob only added complexity and only occasionally was it used for its original purpose, for example when ‘portbill audit created’ and ‘portbill item audit created’ states were generated. Of course since most simple objects did not have a need for the Harel blob in this particular application, they could be removed from the diagram. However, even in this case the overall improvement in readability was marginal at best.

A basic premise since the State Dependency Diagram had been the importance of showing inter-object dependencies. In any system it is likely that some objects will be coupled more strongly than others. Further, in identifying complex objects, highly-coupled simple objects such as order-header and order-line are strong candidates for forming a complex object. So it was decided to attempt to redraw the diagram this time combining the Harel blob with complex objects to test if that would be an improvement. The data model was re-examined and three complex objects identified. Figure 1 of the BSDv2 in appendix 1 shows the OLHs for these three objects identified and figure 2 of the BSDv2 in appendix 1 shows the entity-relationship model from which it was constructed. Figure 9.2 shows the resulting BDSv2 which it is claimed reduces visual complexity.

During the interview with the analyst, he made a comment which supports the view that the use of complex objects to solve overkill made sense.

> *‘We have the complex objects now with this Business State Diagram version 2, but I don’t think you actually needed to use it when you drew this. Is that fair? For example, the invoice and the invoice lines?*
>
> (researcher)
>
> Yes, I didn’t need to, because I encapsulate the invoice line and the GL allocation to one - represented by one object which is invoice. Because if you cancel the invoice then you cancel everything, invoice line GL allocation. So, I didn’t really see the need to show separate invoice line and GL allocation.’
>
> (analyst X)

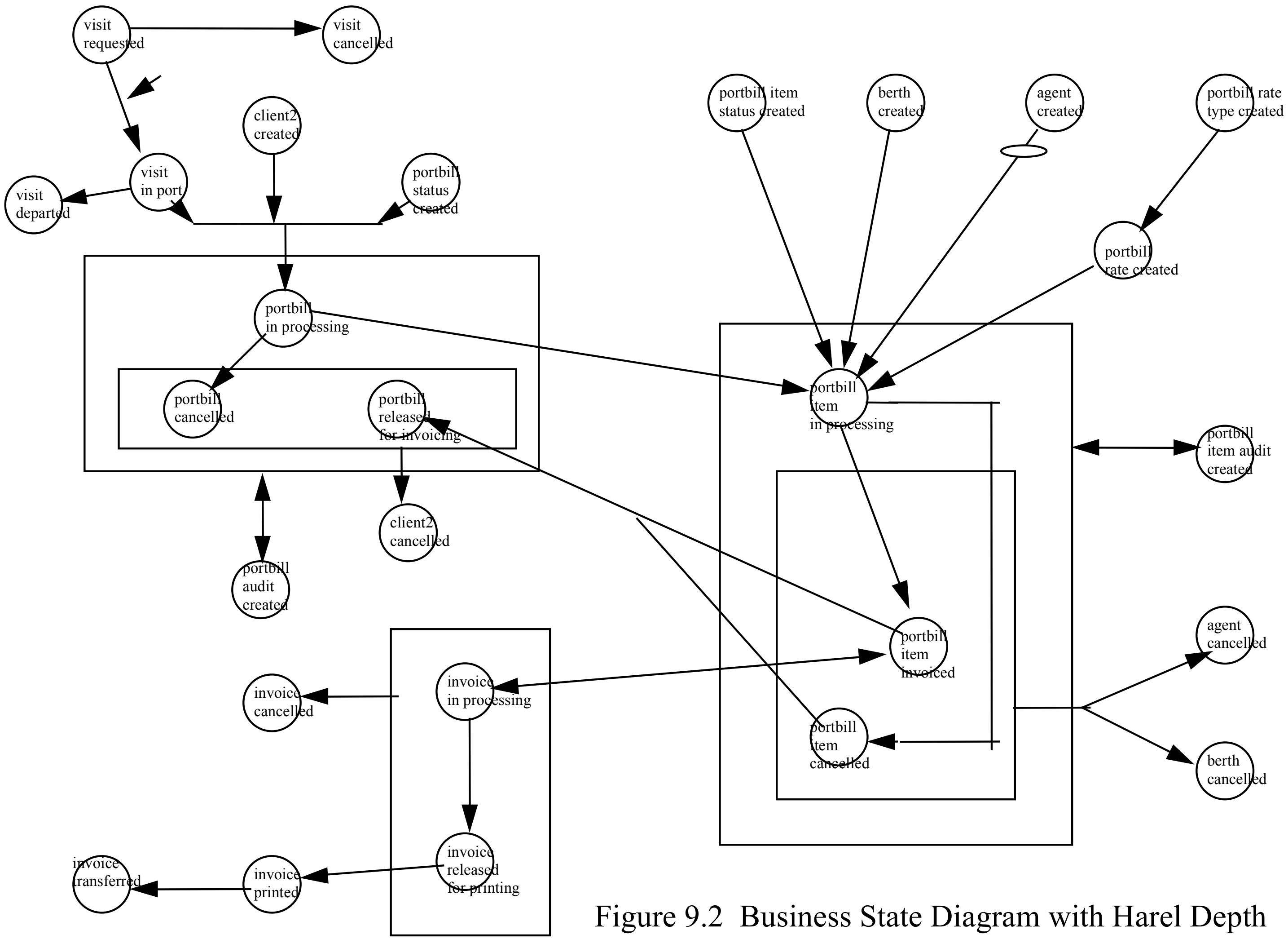


Figure 9.2 Business State Diagram with Harel Depth

Here I had been querying an oversight on the analyst's diagram of the invoicing system but nevertheless, his response indicates clear support of the view that overkill was being avoided through the use of the complex object.

The analyst was also positive about the use of the Harel blob at this stage.

> *'And the last point here is that you've used the Harel blob. Did you find it helpful?*
>
> (researcher)
>
> Yes. It certainly is helpful, otherwise you have too many branches. In this case, it's very clear.... covered the progression to another state.'
>
> (analyst X)

Whilst applying the complex object concept to the portbilling application did not provide any **direct** evidence to support the view that complex objects resolve the overkill problem, the lack of problem with overkill in the resultant diagram supports the view that the use of complex objects might be consistent with avoiding overkill, but of course this is not a strong argument in itself. However, it was felt that the demonstration of the value of complex objects with examples from outside the portbilling application (ie the case study) together with lack of contradictory evidence from within the portbilling application was sufficient to justify its inclusion as a working solution in subsequent versions of the diagram. Once again, the ability to test in later cycles the hypotheses of earlier cycles is seen as a benefit of this kind of research.

Overall therefore the success of applying the Harel blob with the complex object appeared to address the overkill problem (2A2) as well as reducing visual complexity.

9.2.2 Evaluation of Application Step against Intellectual Framework

In terms of the W+W framework, the readability issue (3A1) was viewed as follows. Identifying the concept of encapsulation and complex object as two important and related concepts in the ontological world effectively was an updating of the constructs in the ontological world. 'Good' constructs in the ontological world are likely to facilitate easy mapping onto the design world and thus make any representation of the design world easier to read. Although at one level the W+W framework adequately explained this phenomenon (ie the ontological world had to be redefined to include encapsulation and the complex object and the design world was redefined to include the Harel blob), the W+W scheme had no explicit measure of readability.

9.3 Review Step of the Development of the BSDv2

9.3.1 Complexity (3R1)

There was some feeling that progress was being made, eg the discovery that grouping by complex object led to what we considered a more organised diagram. Further, dependencies between complex objects seemed to be of greater interest to the analysis since they represented connection between semantically different parts of the portbilling system. In the earlier diagram there had been over 20 inter-object dependencies. With the incorporation of complex objects it was expected that the number of inter-object dependencies would drop simply because now many of these dependencies would exist within the object rather than across objects. For the portbilling application the number of inter-object dependencies dropped to three. This was seen as a dramatic decrease.

As for the states in the portbilling application, the number remained the same in both diagrams. This was because in the portbilling application the portbill has a somewhat separate existence from portbill item. For example, the portbill is created sometime before any portbill items. Thus no opportunities arose to reduce the number of states in each complex object. However, in the case study there would have been a reduction in the number of states. For example creating and deleting order states would not require separate states for the creation and deletion of order-lines typically.

9.3.2 Simultaneity (3R2)

The decision (2D1) to add simultaneous post-conditions posed no problem in constructing the prerequisite table (table 9.1).

State	Prerequisite	Precondition	Postcondition
1 Invoice in processing	-		T3: 9
2 Invoice released	1		D1: 3
3 Invoice printed	2		
4 Invoice cancelled	[1|2]		
5 Portbill in processing	18,16,13		T1: 19
6 Portbill released for printing	[9|11|9,11]	C1: No pbill items in proc. or inv'd	T1: 19
7 Portbill cancelled	5	C2: No pbill items in proc. or inv'd	T1: 19
8 Portbill item in processing	25,23,5(,26),22		T2: 20
9 Portbill item invoiced	8,1		T2: 20
10 Invoice transferred to GL	3		
11 Portbill item cancelled	8		T2: 20
12 Visit requested	-		
13 Visit in port	[-,12]		
14 Visit departed	13		
15 Visit cancelled	12		
16 Client created	-		
17 Client cancelled	[6|7|6,7]		
18 Portbill status created	-		
19 Portbill audit created	[5|6|7]		
20 Portbill item audit created	[8|9|11]		
21 Portbill rate type created	-		
22 Portbill rate created	21		
23 Berth created	-		
24 Berth cancelled	[9|11|9,11]	C3: No pbill items inv'd or can'd	
25 Portbill item status created	-		
26 Agent created	-		
27 Agent cancelled	[9|11|9,11]	C4: No pbill items inv'd or can'd	

Table 9.1 Prerequisite table for Portbilling Application

Similarly, the incorporation of the delay symbol into the prerequisite table also caused no problem and further supplied useful information concerning the post-condition. However, a question still remained concerning whether recording simultaneous or indeed other information in the diagram as well as recording it in the prerequisite table would increase the perceived richness or indeed completeness of the diagram. So we attempted to draw a version of the diagram which incorporated the codes for pre-conditions, simultaneity and time delays onto the diagram. The codes on the diagram would alert the reader of the diagram to the existence of the code and therefore of some special pre- or post- condition in the prerequisite table. See figure 9.3. Here for example, C1 represents a pre-condition, T1 simultaneity and D1 that a time delay exists.

As far as showing simultaneity in the diagram was concerned, it could be regarded as a success. In figure 9.2 the double-headed arrow is used to show simultaneous situations in this case simultaneous states, ie that the change to these states should happen at effectively the same time. Note that there are two types of simultaneous situation. The first is between states of different objects eg between 'invoice in processing' and 'portbill item invoiced'. When the invoice becomes active (ie is in processing) it is immediately allocated portbill items. The second type of simultaneous post-condition is between states of the same object, for example, where audit copies are being made of the main transactions in a system.

The question of whether simultaneity should be shown only in the table, only in the diagram or in both had now arisen in the analysis. So the simultaneity problem was considered partly resolved and carried over for further analysis. In summary it appeared that only a relatively minor problem with regard to simultaneity remained at this time.

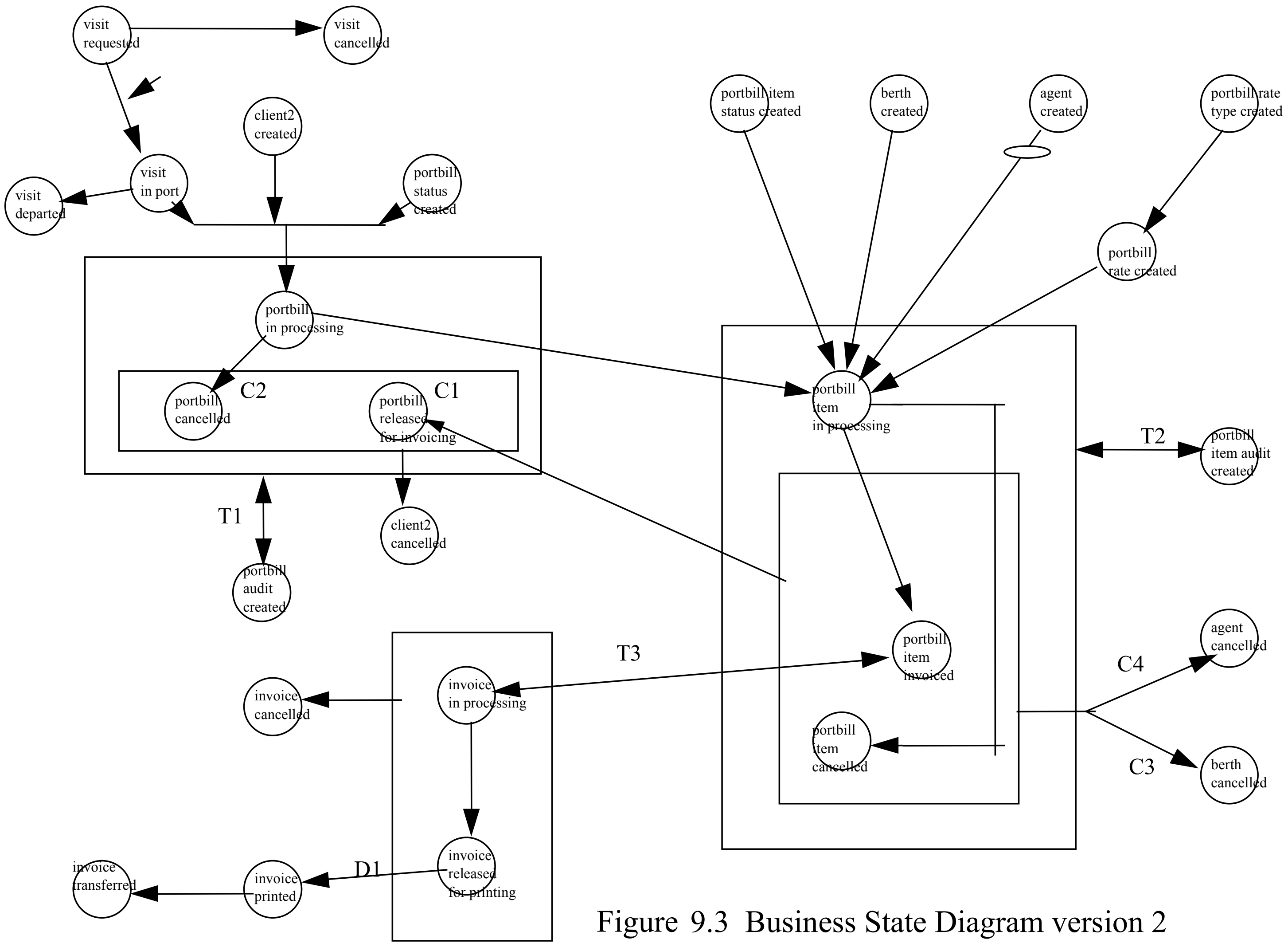


Figure 9.3 Business State Diagram version 2 with Codes

9.3.3 Rule Completeness (3R3)

This section examines the recording of the pre-conditions into the prerequisite table (table 9.1) with respect to the issue of rule completeness. The analyst's comment in relation to pre-conditions was favourable.

> *'The construction process, if we can just review that. We start with the object life histories, then the prerequisite table. This time the prerequisite table had pre- and post-conditions. And then we went on into our first-cut, then delay analysis then final. Do you have any comments on that as a process? And particularly whether the post-conditions helped at all?*
>
> (researcher)
>
> The pre- and post-conditions as I mentioned before, we deal with objects in a very modular fashion. We try to encapsulate the rules and conditions in a modular form, so that when we deal with certain objects we set the pre-conditions before we deal with it and pass the parameters telling the encapsulated module what the conditions are at this point and then when the module provides the feedback it sets the post-conditions and then the programs will deal with the post-conditions as is necessary. So I can relate to that in what I'm doing now at the Port Authority.'
>
> (analyst X)

It was considered that the inclusion of pre-conditions seemed to add an important dimension to understanding the diagram and began to address the problem of rule completeness identified in the two earlier versions of the diagram (1R1, 2R2). However, as in the case of the simultaneous codes, annotating pre-condition codes on the diagram opened up the question of where best to record pre-condition information also and in what way. Further, the assertion that trigger information might also form part of the complete specification of a rule which was posited earlier still had not been addressed. In other words, if the pre-conditions were to be shown on the diagram, should it be as a code or as a graphical construct? So the rule completeness problem was considered only partly resolved, ie some progress had been made. However, since this problem was considered fundamental to this investigation it was carried over to the next iteration of the diagram.

9.3.4 Construction from Complex Objects (3R4)

During the application step, it was speculated that the BSDv2 could be developed in two ways. The first was to build OLHs from simple objects eg a data model. This data model could then be converted to a model involving complex objects and similarly the OLHs based on simple objects combined to form OLHs based on the complex objects. This approach was originally termed the indirect approach. The direct approach in contrast would go straight to the complex objects and life histories.

However, in reviewing this a far more radical thought occurred to me in terms of directness and indirectness of development. This was to do with whether a model containing cardinal relationships was necessary at all! In other words, I speculated whether it would be possible to identify complex objects straight from the problem domain without having to construct an intermediate model. This clearly had not been tested in the construction of the BSDv2 and therefore it was identified an issue.

9.3.5 Evaluation of Review Step against Intellectual Framework

With regard to reviewing the above issues in terms of the W+W framework, the following emerged. The first issue was complexity (3R1). Issues such as complexity were not provided for within the W+W scheme. Throughout this research, this issue had repeatedly presented itself as requiring resolution and thus I began to consider this a deficiency of the intellectual framework adopted. Issue 3R2 referred to the simultaneity problem. In terms of the W+W framework, this was considered resolved (ie the solution preserved ontological completeness) although there remained a question over where simultaneity should be depicted. In 3R3 (rule completeness) progress was made in terms of adding richness to the ontological world and supporting this by an appropriate design construct (ie in the prerequisite table). However, as discussed, the introduction of further ontological and design constructs was also considered possible. Nevertheless, this issue fitted within and reinforced the legitimacy of the W+W framework. The last issue 3R4 (construction from complex objects) referred to the possibility of constructing the diagram directly from complex objects rather than an object or data model. The W+W scheme did not provide any guidance for this. This was true both in terms of depicting alternative construction routes and more significantly in terms of advising on the wisdom of such a question.

9.3.6 Discussion in terms of Wider Literature

The pre-conditions and prerequisite states in the prerequisite table have a similarity to the description of state law in the Wand and Weber representation model (table 7.3). As discussed in the last chapter, this was seen as a way of describing information requirements more fully. However, because events were a basic construct in the representation model many composite constructs could not be generated.

9.4 Decision Step of the Development of the BSDv2

9.4.1 Modelling in both States and Events (3D1)

The first inkling of the idea to model both states and events occurred to me when labelling pre-conditions on the diagram, ie while working on figure 9.3. I postulated that the reason why pre-condition labels had to appear on each arc was because pre-conditions applied to events rather than the states. I considered this a significant observation at the time though on hindsight it may seem obvious. Anyway, it was then a small step to consider modelling both states and events graphically.

Petri-nets are bi-partite directed graphs as discussed earlier. One way of using a Petri-net would be to equate transitions with events and placeholders to states. Of course the Harel blob would still be retained as a notational construct. Thus between the prerequisite table and the diagram, states, events and conditions would be defined. It was thus felt that a more complete specification of the problem domain was now being envisaged.

Further, I began to feel that many of the problems identified were more inter-related than at first suspected. The simultaneity problem (3R3), the rule completeness problem (3R2), the readability problem (3A1) and even the part resolution of the modelling in states solution (1D2) all seemed to be linked together as the notion of modelling in both states and event presented itself to me. With regard to the simultaneity problem, to be able to return to grouping events in the same circle was intuitively attractive because it is the events which are simultaneous and not the states connecting them. The rule completeness problem (3R2) and the part resolution of the modelling in states solution (1D2) would also begin to be addressed since modelling both states and events provided a richer set of constructs and therefore a fuller specification and more expressive power. It also could be argued that the readability problem too would be improved if modelling in states and events made it clearer how states and events inter-related, though there was the obvious concern over the increased complexity that would arise (3R1).

9.4.2 Evaluation of Decision Step against Intellectual Framework

In terms of the W+W framework, introducing state and event as dual modelling constructs implied that both the ontological world and the design world had to be updated. Indeed, it had become a common occurrence to recognise the pattern of having to update both worlds in such circumstances.

9.5 Summary of Progress with the BSDv2

In summary, the construction and subsequent analysis of the BSDv2 moved the research forward in the following ways. It confirmed that:

- using the complex object in conjunction with the Harel blob improved the readability of the diagram and

- the extensions to the prerequisite table provided more information which was perceived to be useful in specifying completeness in relation to the problem domain.

From the review step a proposal emerged for modelling both states and events in the next version of the diagram (3D1). It was felt that this diagram might resolve a number of problems which were seen to be connected in some way (3R1, 3R2, 3R3). The problems of manyness (2R3) and information systems events versus business events (1R1) remained outstanding as this proposal did not specifically address these. The question of directly constructing the diagram (3R4) was also not explicitly to be addressed in the next version of the diagram.

Chapter 10

Action Research Study 1

The Business Rules Diagram version 1

10.1 Overview

There was a need to rethink the name of the diagram. It seemed that the idea of a 'business rule' encapsulated state, event and condition, ie all contributed to the description of a business rule. In other words the event describes what occurs when the business rule fires and the states prior to the event specify the necessary state conditions which have to be in place for the rule to fire. Any non-state condition(s) under which the rule fired would be in the prerequisite table. So together these three components seemed to support the new name of the diagram.

The portbilling application was used for the same reason as in the previous version of the diagram, ie that the analyst and I were working interactively and the portbilling application was able to be used as a common point of reference. Thus the format of this chapter is identical to the previous one.

10.2 Application Step of the Development of the BRDv1

A decision had to be taken as to whether to use state or event as the construct for describing prerequisites in the prerequisite table. It was decided to leave the main construct as state. Previous versions of the diagram had used the state and earlier the event as the basis for specifying prerequisites. With one exception which will be explained shortly, table 10.1 is identical to that in the previous version of the diagram. Figure 10.1 shows the BRDv1 (Business Rules Diagram version 1) for the portbilling application.

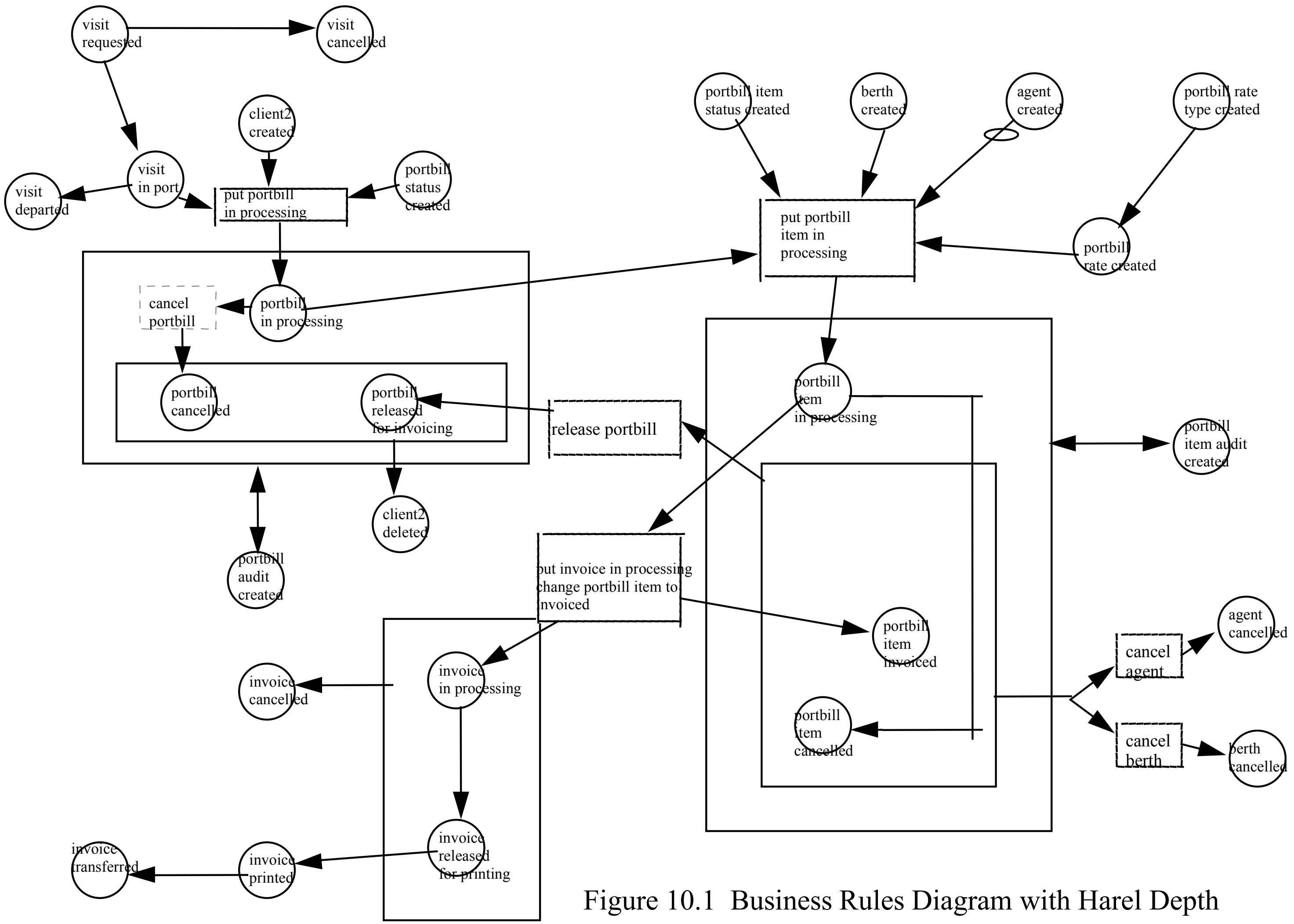


Figure 10.1 Business Rules Diagram with Harel Depth

State	Prerequisite	Precondition	Postcondition
1 Invoice in processing	8		
2 Invoice released	1		D1: 3
3 Invoice printed	2		
4 Invoice cancelled	[1|2]		
5 Portbill in processing	18,16,13		T1: 19
6 Portbill released	[9|11|9,11]	C1: No pbill items in proc. or inv'd	T1: 19
7 Portbill cancelled	5	C2: No pbill items in proc. or inv'd	T1: 19
8 Portbill item in processing	25,23,5(,26),22		T2: 20
9 Portbill item invoiced	8,1		T2: 20
10 Invoice transferred to GL	3		
11 Portbill item cancelled	8		T2: 20
12 Visit requested	-		
13 Visit in port	[-,12]		
14 Visit departed	13		
15 Visit cancelled	12		
16 Client created	-		
17 Client cancelled	[6|7|6,7]		
18 Portbill status created	-		
19 Portbill audit created	[5|6|7]		
20 Portbill item audit created	[8|9|11]		
21 Portbill rate type created	-		
22 Portbill rate created	21		
23 Berth created	-		
24 Berth cancelled	[9|11|9,11]	C3: No pbill items inv'd or can'd	
25 Portbill item status created	-		
26 Agent created	-		
27 Agent cancelled	[9|11|9,11]	C4: No pbill items inv'd or can'd	

Table 10.1 Prerequisite Table for Portbilling Application

10.2.1 The Simultaneity Problem (4A1)

There had been a problem with simultaneity in the previous version of the diagram (3R2). It was a straightforward matter to redraw the diagram with ‘invoice event’ and ‘portbill item invoiced’ event in the same rectangle with dashed lines indicating simultaneity. However, this was still problematic for the two audit events ‘portbill audit created’ and ‘portbill item audit created’ (figure 10.1). Because these two audit events applied to several states in their respective objects this would have resulted in having to repeat the audit event description along with each event with which it was simultaneous. This was seen as a poor solution here. So it was decided to retain the double-headed arrow for use in conjunction with the Harel blob to show simultaneity of this kind more succinctly. However, inter-object simultaneity through the use of the Harel blob was also retained as a clear way of showing simultaneity between objects. So although a little clumsy since two types of solution were required, nevertheless, notationally, this problem was considered resolved for the purposes of the research progress map.

It was at this stage that I realised that I had overlooked an inter-object dependency in previous versions of the diagram. Previously, I had recorded that to put ‘invoice in processing’ there was no predecessor state. The act of drawing a simultaneous event coupled to predecessor states forced me to conclude that at least one portbill item had to be in the ‘in processing’ state before it was sensible to open an invoice, ie portbill item in processing is a prerequisite to creating an invoice. I considered this observation strong support for the benefit of modelling in both states and events and this increased my confidence that a fruitful direction was being pursued.

10.2.2 Readability and the Simple Event (4A2)

There continued to be a concern over the sheer number of symbols that were required on the diagram. Clearly with two modelling units instead of one, this number would rise significantly. Yet, it was argued as necessary to have both, in order to show business rule completeness especially where the rule was more complicated. So it was decided to show events only where they were ‘complicated’, ie where there were multiple pre-states involved or simultaneous events. This was done to try to keep the number of symbols on the diagram to a minimum, yet at the same time show additional information where the complexity of the rule demanded it.

10.2.3 Evaluation of Application Step against Intellectual Framework

In terms of reviewing the Wand and Weber (W+W) scheme against the above issues, the following emerged. In 4A1 (simultaneity) it was decided that two design constructs were required to support this problem. On the face of it, this could be viewed as construct redundancy since arguably two design constructs map onto a single ontological construct. However, it was not clear that this was so. If the ontological world distinguishes the constructs of inter- and intra- object dependency then ontological completeness can be argued to be preserved. The second issue here related to the simple condition (4A2). Here a form of ontological incompleteness was deliberately introduced to ameliorate perceived readability problems. The W+W scheme would predict this as an unwise act, though in itself it takes no account of readability as an issue and further at this juncture no alternative solution was able to be identified.

10.3 Review Step of the Development of the BRDv1

10.3.1 Complexity (4R1)

Adding events as a modelling unit in the diagram even though this was limited to complex events had reduced the readability of the diagram. When interviewed the analyst agreed with the importance of showing complex events.

> *'The notation we had with this diagram, we had simple events then complex events and then complex multiple events. How did you find those? With the simple events we have just a state then an event then another state. The complex event means we involve more than one state before the event can fire, but there is only one output state for that event. With the complex multiple event the*

difference is that there can be two or more output states from that event. (researcher).

I think it very much depends on the complexity and the rules within the application. The application could be quite big and a lot of objects are involved and there is a necessity to have a complex event. As I said before in our work we have certain database transactions we could have various events take place to constitute a whole transaction. So it is a real situation.'

(analyst X)

10.3.2 Rule Completeness (4R2)

The use of both state and event as dual modelling units had raised the question of going further diagrammatically by adding a graphical construct to show conditions. The following is an excerpt from my field notes.

'Perhaps a better way of viewing the business rule problem is to put conditions onto the diagram as well as states and events rather than keeping them [conditions] in the prerequisite table. Thus we will have a more complete visual expression of the rule.'

However, there was the obvious point over reduced readability which was summed up by the analyst during his interview.

'One of the dilemmas I faced was on the prerequisite table the pre-conditions were specified. Now that's the way we used it, but I wonder whether it could have been shown on the diagram rather than on the table. Would you have a view as to whether it would have been better to have it on the table, diagram or both? So we could have shown those pre-conditions on the diagram. We didn't do that but we could have put it on the diagram.

(researcher)

I suppose you can introduce some new notation or symbols to encompass those business rules but the only danger is that if there are too many rules you clutter up the diagram. It may be better to have the business rules outside the diagram, so the diagram is clearer. Or you can have some notation to say refer to some business rules document or pre-condition document. Otherwise you may clutter up the diagram and make it not so readable.'

(analyst X)

However, at this point remember that it was unclear as to what the impact of adding another modelling construct would be. In the current version of the BRDv1 because simple events were omitted, the number of additional complex events that was required was relatively small for the portbilling application. It was postulated that similarly with conditions the impact may prove not so significant especially if some similar strategy was followed where 'simple' conditions were omitted. On the prerequisite table for the portbilling application (table 10.1) most events so far did not require an explicit pre-condition to be recorded. Only four of the 27 rows had pre-conditions. It was decided to attempt this in the next version of the diagram in order to obtain an indication of the impact on readability as well as a feeling for rule completeness.

Deciding to include conditions in the diagram also raised the question of whether triggers should also be included in the diagram. The possible use of triggers had been discussed in two earlier review sections on rule completeness (1R2 and 3R3).

It could certainly be hypothesised that triggers were a component of a business rule. Without knowing the trigger for a business rule, something is missing it can be argued. Mostly triggers would be external triggers involving the receipt of a document from outside of the application. In the case study, examples include the receipt of a customer order or of a payment. The arrival of these documents then triggers a train of events in the system. Now if the trigger is not included as part of the business rule it can be argued to be incomplete since it is the arrival of the trigger that sets the business rule in motion. Another kind of trigger is a time trigger. A time trigger initiates events in a system based on some time occurrence. For instance, the business rule 'mail orders are checked at 9.30 am' indicates a time trigger (ie that 9.30am has occurred) initiating an event called 'check mail order'. There would clearly be something missing if the phrase 'at 9.30am' was omitted from the business rule.

So it was decided to include triggers in the next version of the diagram also in order to explore full rule completeness. It was recognised that it was not necessary to have two diagrams to test the impact of conditions and of triggers. In other words the impact of adding triggers could be assessed independently of the impact of adding conditions on the same diagram (4D1).

10.3.3 Information System Events versus Business Events (4R3)

This issue had been identified in the first version of the diagram, ie the BED (Business Event Diagram) and had remained unresolved (1R1). In the Port Authority the policy is to create entities which hold status and other related information. This leads to the creation of many additional entities. For example in the portbilling application the portbill status, portbill item status, portbill rate and portbill rate type fall into this category. In the case of portbill status, the possible states are ‘in processing’, ‘released’ or ‘cancelled’. The prerequisite table (table 10.1) is specifying that there must have been previously a creation of for example the ‘in processing’ state in order for a portbill to be put into the ‘in processing’ state. Similarly for portbilling rate there must be a rate, presumably a current rate, before the portbill can be put into the ‘in processing’ state. Such ‘rules’ can be considered information systems rules since they are argued to reflect how the system is implemented. From a business perspective the rules which ‘matter’ are whether the ship has arrived in port, whether the client is a registered client and so on. At a high level of abstraction, we can assume that the organisation has pre-existing rates for work done and so the existence of ‘in processing’ states and the like seems to get in the way of describing the business. Thus we can distinguish between business and information systems events, states and conditions or put more succinctly business rules from information systems rules.

Given the concern over the complexity of the diagram, it was considered that there may be advantage in showing only ‘business rules’ on the next version of the diagram. While it was also recognised that capturing information systems rules was important too it was decided that the priority in this research would be to capture business rules for two reasons. The first reason was that it allowed us to concentrate on those questions concerned with the essence of the business. Secondly, there may be a number of ways that information systems rules can be implemented or modelled and therefore that too suggested a separate analysis.

Having conceptualised the dichotomy between business rules and information system rules, it was also observed that the audit rules, ie portbill audit and portbill item audit could be classified as information systems rules, thus reducing the complexity of the portbilling application diagram further.

The suggested solutions to the last two problems (ie 4R1 and 4R2) are likely to influence the overall complexity in the diagram in opposite ways. On the one hand, increasing the number of constructs in the diagram will increase its complexity. On the other excluding so-called information systems rules will likely reduce the overall number of symbols on any one diagram and therefore serve to reduce its complexity by containing fewer symbols. If both solutions were implemented in the same diagram, would the overall effect, result in an acceptable level of complexity? Figure 10.2 sums up this question.

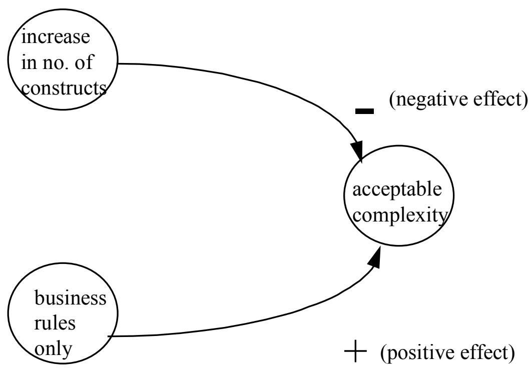


Figure 10.2 Trade-off in achieving an acceptable Level of Complexity

10.3.4 The Manyness Problem (4R4)

The manyness problem had remained unresolved since its identification in the BSDv1. Essentially, the problem was that it was felt desirable to show whether one or many instances of a state of an object contributed only once or perhaps more than once in the firing a business rule. For example, in accepting an order, good customer status is checked only once but many stock levels (ie instances of the stock object) will require to be checked typically. In extending the prerequisite table, it was observed that by adding a repetition symbol (ie a '+' sign) as a superscript to each prerequisite state where manyness occurs appears to resolve the problem (De Marco 1978). So it was decided to modify the next version of the diagram accordingly.

10.3.5 Direct versus Indirect Construction (4R5)

This was a time of convergence of ideas. The move towards emphasising the business rule also led to the re-opening of the question of whether the indirect approach to construction (ie previously constructing a full data or object model) or the direct approach would be more desirable. Clearly, if it were possible to demonstrate that the direct approach was feasible as part of the research this would add to the attractiveness of the BRD since it presented the opportunity to model at the business rules level comparatively directly with users without the intervening traditional data modelling or object modelling steps such as identifying cardinalities (Rumbaugh et al. 1991). Of course, just how feasible the direct approach would be, still remained untested at this stage. The next version of the diagram could examine this.

10.3.6 Evaluation of the Review Step against Intellectual Framework

A review of the above issues in terms of the W+W framework produced the following. 4R1 (complexity) was a problem which had appeared before and which was not provided for in the W+W scheme. Similarly, 4R2 (rule incompleteness) had appeared as a problem before. Here the issue was one of ontological incompleteness as was the case with 4R4 (manyness). Issue 4R3 (information systems event versus business event) was noted as far back as the BED (ie first iteration of the diagram). It may be recalled that it was suggested that one solution had been to postpone showing information systems events to a later ontological world. 4R5 (direct versus indirect construction) was an issue which was not dealt with by the W+W framework as previously discussed.

Most noticeably, it was observed that these issues were continuing to re-appear. For example, rule completeness had been identified in 1R2, 2R2, 3R3 and 4R1. This was an indication that the current intellectual framework was not adequately dealing with these and was a justification for review at a more fundamental level.

10.4 Decision Step of the Development of the BRDv1

Since the rationales for these decisions have been presented in the previous section, the decisions themselves will be recapitulated here only briefly.

10.4.1 Show Conditions and Triggers on the Diagram (4D1)

Having taken the decision previously to model two types of construct in the same diagram (ie state and event), it seemed logical to explore the possibility of extending this to three or four constructs.

10.4.2 Prerequisite Table (4D2)

At the same time as the role of the diagram was becoming clearer, the role of the prerequisite table was extended to include showing manyness. In retrospect, I believe this was a change in role of the table from one used to aid construction to one of storing details not depicted on the diagram. This is elaborated in later chapters.

10.4.3 Refine Definition of Rule (4D3)

A decision was taken to model rules at a high level of abstraction, so that some aspects previously modelled could be ignored. This was argued to be a test at this stage as to whether it was feasible to model aspects of business without requiring to model what were later referred to as processing and implementation rules.

10.4.4 Direct Approach (4D4)

Another significant idea in this research was being formulated at this juncture. It regarded speculation as to whether business rules could be captured directly without the need to construct a model (such as a data or object model) which many would perceive as information systems bound rather than a conceptual model of the business itself.

10.5 Summary of Progress with the BRDv1

In summary, this analysis has indicated a number of changes for the next version of the diagram. Problems or issues in applying the ideas to the business application had resulted in:

- recommending that two approaches be retained for showing simultaneity, ie using both the Harel blob and the double-headed arrow (4A1) and
- recommending that for simple events a fewer number of constructs be used (4A2).

In the review phase a major shift in emphasis of the diagram was discussed. This was a shift towards focussing on the business rule as a concept and proposing that the condition, state, event and trigger constructs together constituted a business rule. This raised the question of whether conditions should be shown on the diagram alongside states and events. However, a major concern was growing between showing completeness of a business rule (4R2) and ensuring the diagram was as readable as possible (4R1). One response to the readability question was to reduce the complexity of the diagram by modelling only the business aspects of rules (4R3). In 4R4 a tactic of using the repetition symbol in the prerequisite table to show manyness was discussed and in 4R5 it was mooted that this diagram could be constructed directly without creating an intermediate data or object model. The outcome of the review resulted in the following proposed changes (decisions). These were:

- to show conditions and triggers on the diagram (4D1),
- to include a manyness symbol on the prerequisite table (4D2),
- to remove information system states and events from the diagram, ie to focus solely on the states and events relating to business rules (4D3) and
- to demonstrate the feasibility of directly modelling business rules without recourse to data or object models (4D4).

10.6 Preparation of the Business Rules Diagram version 2

At this point I intended to conduct a review of the intellectual framework (ie at level 2 learning), so I felt it important to have a 'clean' version of the diagram which would then be available for whatever decisions were to be taken next. I decided to rewrite the BRDv1 to incorporate those decisions taken in the previous section. So it is appropriate here to describe the issues identified in preparing this new version, ie the Business Rules Diagram version 2 (BRDv2). Because it might well be decided to involve a different analyst I reverted back to the case study in order to describe the diagram. Further, it was also necessary to provide adequate background to the potentially new reader. This was done by constructing examples of typical business rules and illustrating these in the paper. The change in the construction process to a direct approach (4D4) took some time to develop but in itself posed no real problem in its writing. The description of the construction process involved identifying firstly candidate business rules and then candidate events. It was proposed that this could be done with users by brainstorming lists for each of these. Arguably this could have been done in a different sequence, ie events first. Alternatively mechanisms other than brainstorming could be used eg the analyst interviews users or scans existing documentation.

Five issues emerged in the preparation step while implementing the set of changes identified in the last section. The first was to do with business rules versus information systems rules. The second concerned trigger selection and the third a minor notational nuance with the Harel blob. The fourth issue was concerned with sub-states. The last problem area was the event specification guide. These are now discussed.

10.6.1 Business Rules versus Information Systems Rules (5P1)

The basis for the original idea of separating business rules from information system rules had been to encourage focus on those aspects that were central to the business itself. In other words, the idea was that a discussion of business rules with users could take place which could 'capture the business' yet would be unencumbered by detail which obscured or deflected attention. A good example of this distinction arose in the portbilling application. The existence of status entities forced the inclusion of rules such as 'there must be a portbill status entity in which the status is created before any portbill can be created'. Clearly such rules would get in the way of focussing on the actual business itself. On the other hand, for the system to be implemented on a computer such rules would have to be defined before a database transaction for example is executed.

Reflection while writing the paper revealed a better way of thinking about this dichotomy. In particular I felt it important to distinguish between rules in which there is some choice for the designer of the information system to decide how such a rule would be implemented and those rules for which there is effectively no choice. For instance, the above case concerning status entities is only one way of implementing such rules. Other tactics could totally avoid the use of status entities altogether. I decided to call these implementation rules to highlight that there was choice in the way they were implemented. The portbilling application also provided an example of rules for which there is effectively no choice at the implementation level yet are arguably still not what one would wish to consider as a business rule. For example, the need to make a copy of every event that takes place with respect to a portbill or a portbill item resulted in the creation of the portbill audit and portbill item audit entities as repositories for these events. While important for providing an audit trail and thus supporting the smooth running of the system such rules arguably obscure discussing the business itself. However, here the rules were not implementation dependent as in the last example. Such rules arguably exist at a higher level. In recognition of this distinction I decided to extend the dichotomy to a trichotomy and rename the levels policy, processing and implementation accordingly. I felt that the use of the term policy rule instead of business rule in the paper served to highlight the focus on the business itself. One last point with regard to policy rules is to observe that as predicted there were fewer rules in this version of the case study than previously (compare the BRDv2 in appendix 1 to earlier versions). In this version there are no rules concerning the deletion of objects such as customer or order since these are now classified as processing rules and therefore do not appear as policy rules.

10.6.2 Trigger Selection (5P2)

Deciding on a symbol for the trigger and incorporating it into the diagram posed no problem. However, while writing the section on examples of business rules for the paper, it occurred to me that there was an opportunity to specify a type of business rule using the Harel blob in conjunction with the trigger symbol. For instance suppose in an organisation there is a business rule which allows for the receipt of mail and telephone orders but not faxed orders for some reason. Using the Harel blob for trigger selection as shown in figure 10.3 illustrates how this type of rule can be easily incorporated into the diagram.

10.6.3 Harel blob for simple conditions (5P3)

There were very few situations which were simple conditions in the case study. However, apart from very complex conditions such as ‘accept order’, a frequent pattern was emerging in which one of two events typically occurred. Such patterns seem to be common at the boundaries of human activity systems. For example, when a complaint is sent by a customer either it is accepted or it is rejected. If it is accepted then a credit note is generated and a corresponding state change occurs. If it is rejected there is no corresponding state change as nothing changes inside the system in this case. Strictly this was a complex condition since it did not conform to the definition of a simple condition and so a Harel blob was drawn round those conditions in the diagram.

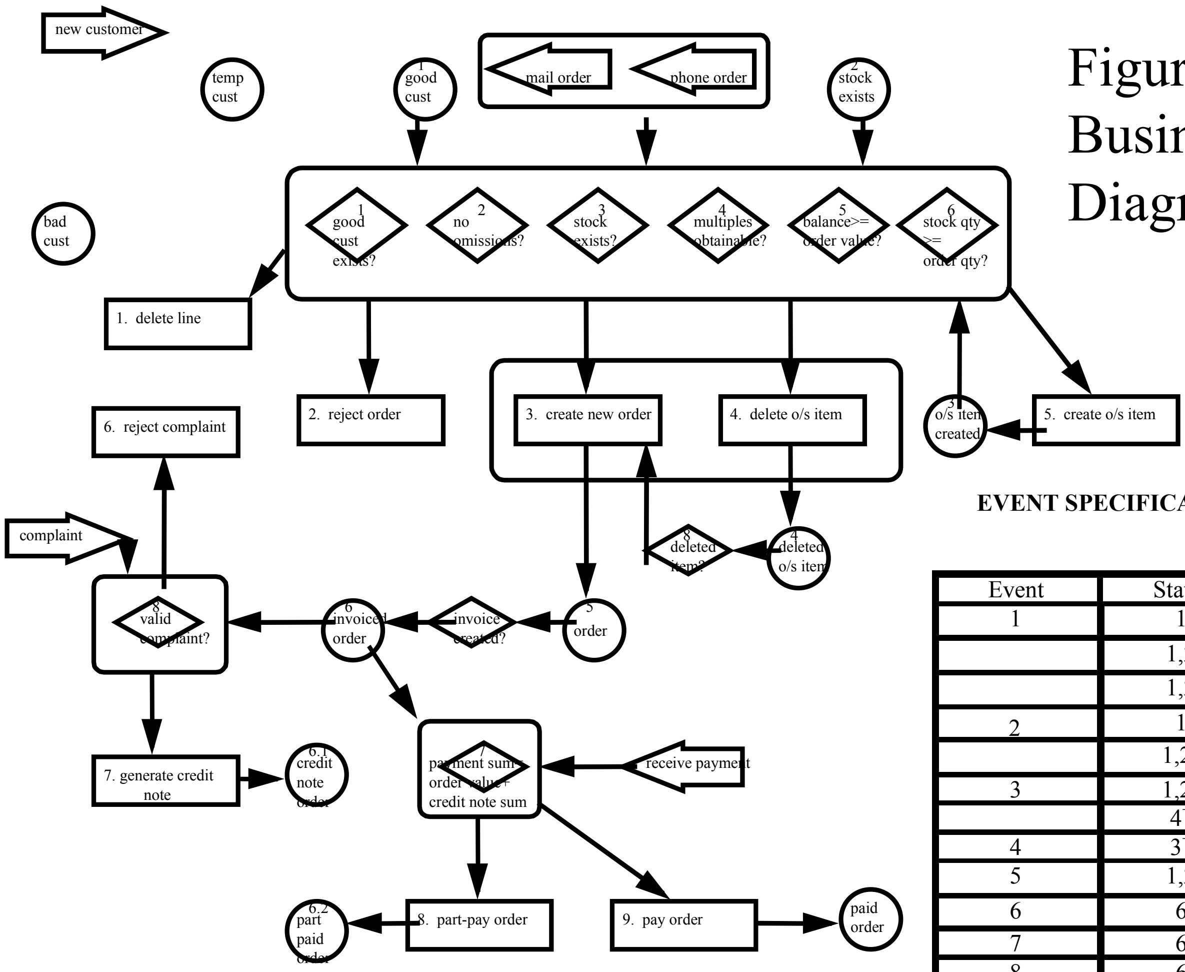


Figure 10.3 The Business Rules Diagram

EVENT SPECIFICATION GUIDE

Event	States	Conditions
1	1	$1,\overline{2}$
	1,2	$1,2,\overline{3}$
	1,2	$1,2,3,\overline{4}$
2	1	$\overline{1}$
	$1,2^{+}$	$1,2,3,4,\overline{5}$
3	$1,2^{+}$	1,2,3,4,5,6
	4^{+}	8
4	3^{+}	6
5	1,2	$1,2,3,4,5,\overline{6}$
6	6	$\overline{8}$
7	6	8
8	6	$\overline{7}$
9	6	7

10.6.4 Sub-States and Parallelism (5P4)

In the OLH (object life history) for the order object so-called sub-states had been identified. The process of identifying states had always been seen as a stepwise refinement process in the sense that as analysis proceeds continual refinement and therefore changing of ideas may occur. When OLHs are drawn at first there is often insufficient collective knowledge about the system to identify states in the OLHs categorically. It is only through the elaboration process that a fuller appreciation emerges. The problem here was that these sub-states appeared to occur in parallel with each other. This had been dealt with in Harel's paper and in fact is a common occurrence in real-time systems (Harel 1988). I elected to follow Harel's solution of using dashed lines in the OLHs to indicate parallel states and used a simple catenation system to code these states as sub-states in the BRDv2 itself. This issue, the wisdom of using sub-states and its implications are reviewed later.10.6.5.

10.6.5 Event Specification Guide (5P5)

Two points emerged under this heading. Firstly, an important difference appears in the layout of the event specification guide. The difference is to do with the fact the same event can be fired for different reasons. For example, consider figure 3 of the BRDv2 in appendix 1. Event 1 (delete line from order) has three ways in which it can be fired. These ways are if there are omissions on the order, if insufficient stock exists or if the quantity required cannot be made up from the multiples that are available for that stock item (eg 17 litres of paint cannot be bought in 5 litre multiples). Previously, the prerequisite table had not shown events and so the opportunity to cluster events had not arisen. It was decided to order the prerequisite table by event since it gave some sense of order to the table. The name of the prerequisite table has been changed to the event specification guide to reflect this difference.

The second point relates to post-conditions which appeared previously on the prerequisite table. These post-conditions indicated simultaneous events and delay events. These were dropped from the event specification guide as a column. Simultaneous events were able to be shown graphically by the use of the Harel blob and if required an extension to the Harel blob (not described) could be used to show delays between events.

10.6.6 Evaluation of Preparation Step against Intellectual Framework

In terms of reviewing the above issues against the W+W framework, the following emerged. Issue 5P1 (business rules versus information systems rules) was seen more clearly by extending the diagram used by Wand and Weber (1993). See figure 10.4.

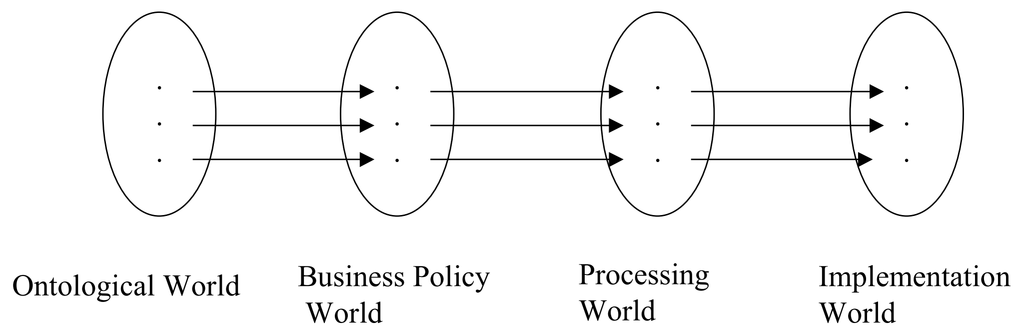


Figure 10.4 Extension of Mappings from the Ontological World

The assertion of three levels of rules implied that there needed to be three distinct mapping processes involved in a sequential manner. Issue 5P2 (trigger selection) was essentially an addition to the ontological world with corresponding additions in the design world. Issue 5P3 (Harel blob for simple condition) was an addition to the design world. However, one concern I had at this juncture was that the Harel blob was being used in different situations, ie as an encapsulator for conditions and as an encapsulator for trigger selection. On the other hand, in both situations the function performed was the same, ie encapsulation. Issue 5P4 (sub-states and parallelism) required that the ontological world had to be updated to accommodate a perspective that states within an object might exist in parallel. The decision to include it here was more of a test than a firm conviction that such a construct was necessary. If it did prove necessary, then clearly both the ontological and design worlds would have to be modified to ensure a clear and straightforward mapping. The last issue, 5P5 (event specification guide) in terms of the W+W framework was determined to be construct redundancy. This was because the constructs of event and state were being depicted in both the event specification guide and in the diagram itself. Arguably, this was construct redundancy. However, since one main purpose of the event specification guide was to show the logic of conditions (ie something which was not available from the diagram), this could not be done without referring to events and states. This was acceptable construct redundancy since it served a useful purpose.

10.6.7 Discussion in terms of Wider Literature

The use of three constructs ie state, event and condition appeared to map onto state, event and law in the representation model (table 7.2) very closely. Further, the separation of business rules into policy, processing and implementation (figure 10.4) provided a means to reflect the varying levels of abstraction and elaboration of requirements which are typical in IRD (Vitalari 1992). Overall, the work so far was beginning to confirm and be consistent with other work in the literature.

10.7 Summary of Progress with the BRDv2

At this point a revised paper entitled the BRDv2 was now written and was ready for use in the next phase of the research. Though ideas were still evolving, it was considered prudent to have created a revised paper which embodied the latest ideas on the diagram. It was now possible to proceed to a critical review of the research so far with a definitive statement of the latest version of the diagram.

Chapter 11

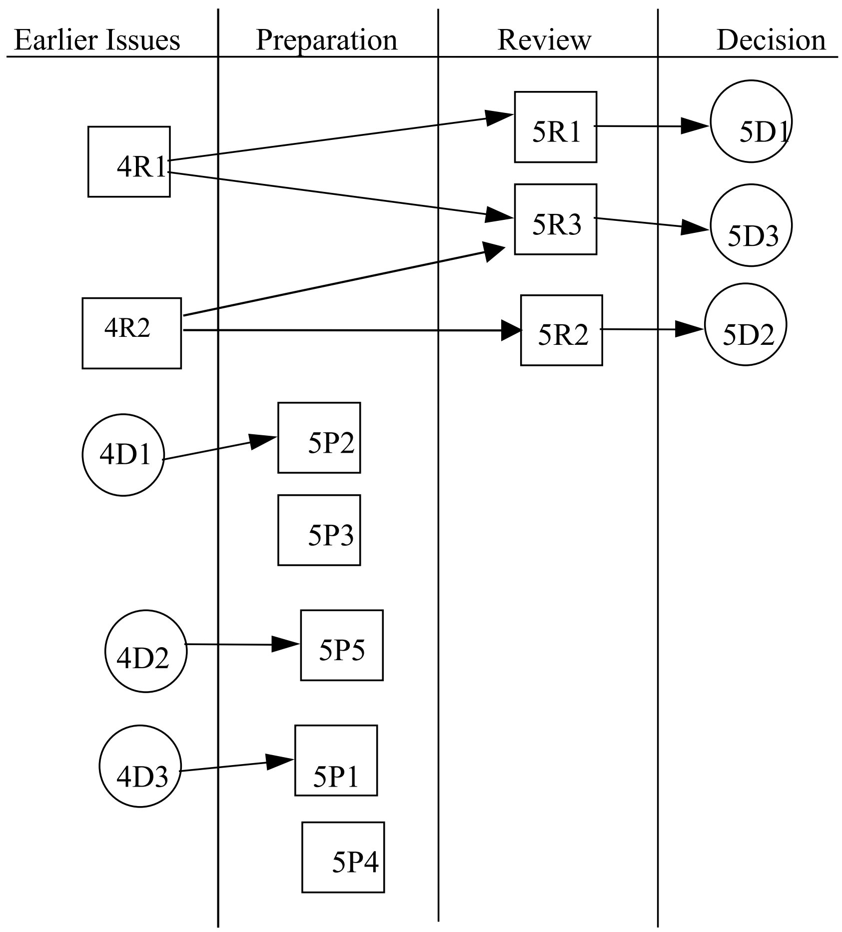
Critical Review of Research Approach

11.1 Overview

The purpose of this chapter is to critically review the progress of the research so far. In the previous four chapters, research progress had been documented in terms of the detailed development with regard to ontological ideas embedded in the methodology itself. In this chapter a more fundamental review takes place which critically examines the intellectual framework F, the methodology M and the area of application A. This is the activity referred to in figure 6.5 of the design methodology chapter as 'reflect and record learning in F, M and A'.

Figure 11.1 summarises research outcomes in this chapter. For convenience of reporting, the issues involved in preparing the BRDv2 (Business Rules Diagram version 2) were reported in the previous chapter. It will be noted that in figure 11.1 there are no progress codes for the application step since in this critical review there was no applied activity. However, while conducting the review it was decided to implement three decisions and these are coded in the review and decision columns in similar fashion as previously. These decisions resulted in a further version of the diagram being created and tested.

This chapter proceeds as follows. The next section critically reviews the success of the framework of ideas (F) employed by discussing its strengths and weaknesses and outlining changes in the framework considered useful for the next study. This is followed by sections which review the methodology (M) and the area of application (A) respectively.



Legend:

4R1	complexity
4R2	rule completeness
4D1	show conditions and triggers on diagram
4D2	prerequisite table
4D3	refine definition of rule
5P1	business rules versus information systems rules
5P2	trigger selection
5P3	Harel blob for simple conditions
5P4	sub-states and parallelism
5P5	event specification guide
5R1	messages as part of a business rule
5R2	one-to-many business rule
5R3	separating user and analyst diagrams
5D1	add message construct to business rule
5D2	implement one-to-many business rule
5D3	implement user diagram

Figure 11.1 Research Progress Map for the Critical Review

11.2 Review of Framework F

Figure 11.2 shows an extended version of the original W + W model. I constructed this model by reflecting on the preceding research. It highlights various categories of issue that had emerged in the research.

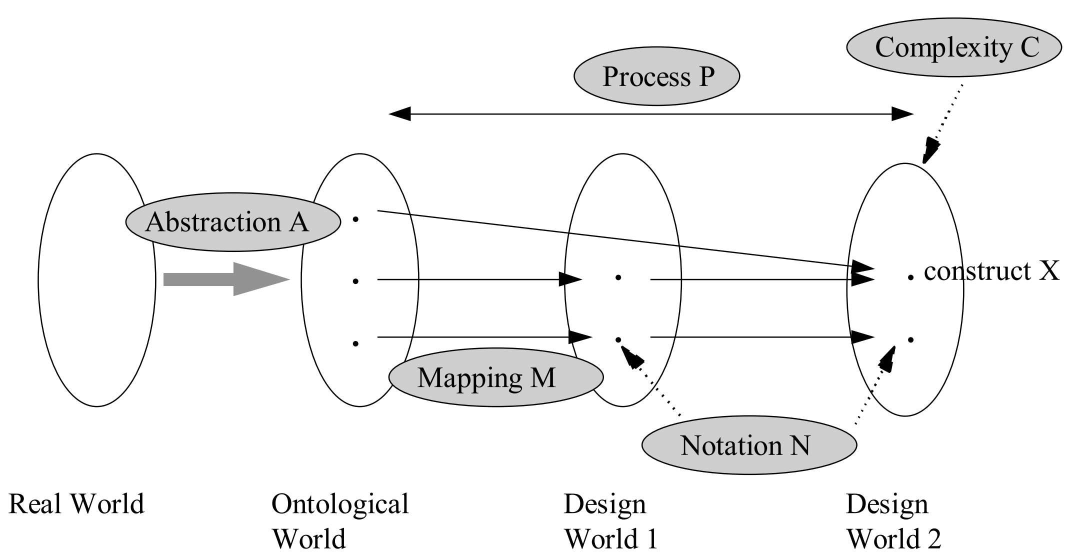


Figure 11.2 Extension of Wand and Weber's Model

Two aspects are particularly noteworthy. Firstly, the diagram shows an abstraction from the real-world to the ontological world. Whereas mappings can be argued to exist from the ontological world to the design world and from one design world to the next, the relationship between the real-world and the ontological world is better described as an abstraction. The ontological world contains a set of constructs which are considered to represent the real-world or some part of it. Other ontological worlds may similarly assert that they also represent the real-world. By declaring an abstraction between the real-world and the ontological world we focus attention on the purpose of the ontological world which is to represent what is considered important in the real-world with respect to the problem on hand. The second aspect is to note that sometimes constructs from both the ontological world and the design world can be mapped onto constructs in another design world. For example, in creating an object life history (OLH) in the Business Event Diagram, firstly an object model would have to have been constructed. However, in order to identify the states of an object, more information is required than is contained in an object model. In other words, additional information from the ontological world would be required as well as from the first design world in order to create a construct in a second design world. I refer to this as the **elaboration** problem. The elaboration problem was avoided in W+W's original paper because they did not show transitive mappings. Construct X in figure 11.2 illustrates this situation graphically though somewhat ambiguously. In figure 11.2, the thick arrow represents abstraction, single-headed arrows represent mappings, the double-headed arrow shows the span of effect of the process P category and the dashed arrows show the object to which category relates.

Figure 11.2 also contains a number of annotations. These annotations represent **categories** into which all the issues occurring in action research study 1 have been abstracted. Abstraction (A) is to do with the process of abstracting the real-world into the ontological world. So, where issues arose concerning whether a construct ought to exist in the ontological world, this issue was categorised as an abstraction issue. Mapping (M) is a category which deals with issues associated with mapping from one set (world) to another set (world) and are typically explained by the original W + W framework in terms of ontological incompleteness, ontological completeness, construct overload, redundancy and excess. Notation (N) is a category of issues concerned with notation, ie how best to depict a particular construct. Process (P) is a category which refers to issues to do with process steps in creating a design construct. The last category was complexity (C). This category is allocated to issues associated with the perceived complexity of a design world in terms of how easy such a world would be to be construct or use.

Table 11.1 contains an allocation of issues occurring in action research study 1 into the above categories. It will be observed that sometimes a single issue is allocated more than one category. This is because the nature of that issue was considered to have implications whose cause may be attributed to more than one category. The sub-sections below now review each category in turn.

Research Code	Description	Evaluation
1P1	event focus	A
1P2	object-oriented basis	P
1P3	group event notation	A, N
1P4	delay analysis step	P
1P5	event notation refinement	N
1A1	prerequisite flaw	M
1A2	union and intersection	M
1A3	alternative point of entry	A, M, N
1A4	fan out dependency	N
1A5	contingent dependency	N
1A6	simultaneity	M, N
1R1	IS events vs business events	A
1R2	rule completeness	A
1R3	state vs event	A
2P1	simultaneity	N
2A1	simultaneity	M, N
2A2	overkill	A, M
2R1	readability	C
2R2	rule completeness	A
2R3	manyness	A
2D1	extend prerequisite table	A, N
2D2	complex objects	A, M
2D3	Harel blob	M, N
3A1	readability	C
3R1	complexity	C
3R2	simultaneity	A, N
3R3	rule completeness	A, N
3R4	construction from complex objects	P
3D1	states and events	A
4A1	simultaneity	M, N
4A2	readability and the simple condition	C, N
4R1	complexity	C
4R2	rule completeness	A
4R3	IS rules vs business rules	A
4R4	manyness	A, N
4R5	direct vs indirect construction	P
4D1	show conditions and triggers on diagram	A, M, N
4D2	prerequisite table	N
4D3	refine definition of rule	A
4D4	direct approach	P
5P1	business rules versus information systems rules	A, M
5P2	trigger selection	A, N

5P3	Harel blob for simple conditions	N
5P4	sub-states and parallelism	A, M, N
5P5	event specification guide	M, N

Legend:

A abstraction issue
C complexity issue
M mapping issue
N notational issue
P process issue

Table 11.1 Allocation of Issues into Categories

11.2.1 Mapping (M)

Table 11.2 summarises the issues that were classified under the category of mapping (M).

Research Code	Description
1A1	prerequisite flaw
1A2	union and intersection
1A3	alternative point of entry
1A6	simultaneity
2A1	simultaneity
2A2	overkill
2D2	complex objects
2D3	Harel blob
4A1	simultaneity
4D1	show conditions and triggers on diagram
5P1	business rules versus information systems rules
5P4	sub-states and parallelism
5P5	event specification guide

Table 11.2 Research Codes Classified under Mapping (M)

This category of mapping is the category in which the W+W framework was expected to directly provide support. In other words, where errors or misapplications in mappings between the ontological world and the design world had occurred, conceptualising the mappings in terms of ontological incompleteness and so on was a way of determining the nature of these problems. Indeed this was straightforwardly so for issues 1A1, 1A2, 1A3, 2D2, 2D3, 4A1, 4D1, 5P1 and 5P5. It was also true for the other issues, ie 1A6, 2A1, 2A2 and 5P4 but these were slightly more complex.

Issues 1A6 and 2A1 involved simultaneity. Here it was required to show that events or states associated with taking copies of events for audit purposes would occur simultaneously. The dilemma in essence was that since no candidate for the notation could be found, it was not clear whether the problem was merely a notational one or something more fundamental involving the mapping between the ontological world and the design world. In reviewing issue 2A2 (overkill) against the W+W framework I had observed that the W+W framework had no sense of how **easy** it would be for analysts to successfully map from the ontological world to the design world. This observation is now revisited. If the ontological world contains clear and well-defined constructs then in practice it should be a straightforward matter to map onto the design world. This suggests that perhaps the ontological world while undertaking this research was possibly not well-defined in this area after all. As it turned out, the introduction of complex objects into the ontological world, effectively resolved the matter, thus verifying the assertion. Issue 5P4 (sub-states and parallelism) related to mapping between one design world (object life histories) and another (BRD). This in itself was a relatively straightforward mapping. However, it led to speculation about whether in the ontological world it was necessary (or desirable) to define parallel constructs.

All the issues in the above paragraph illustrate how focussing on mappings can lead to analysis and speculation about underlying causes of problems. It is noteworthy that such causes may in fact be outside the scope of the mapping itself, ie that using a framework does not limit speculation about possible causes of phenomena to causes inside the framework itself.

11.2.2 Complexity (C)

Table 11.3 summarises those phenomena which were considered to be related to the problem of complexity (or readability) in the diagram. Complexity and readability issues have been bundled together because in reviewing this issue it became clear that the factors of size, readability and complexity were strongly inter-related. For example, in 2R1 (readability) it was stated that '[the diagram] remains large and clumsy' and later 'if there was a way of making it [the diagram] more readable perhaps by reducing the number of states'. Similarly, when discussing complexity, the size of the diagram was also perceived to be a significant factor in statements such as 'no opportunity arose to reduce the number of states'.

Research Code	Description
2R1	readability
3A1	readability
3R1	complexity
4A2	readability and the simple condition
4R1	complexity

Table 11.3 Research Codes Classified under Complexity

Unlike the previous category, complexity is not addressed within the W+W framework. Since the original intellectual framework did not contain any formal ideas about complexity, it would have contradicted the philosophy of the framework to have introduced formal ideas **during** the first action research study. However, it is now necessary for the intellectual framework to be updated to include a way of measuring complexity and through measurement dealing with complexity and readability issues.

McCabe (1976) provides a well-established graph-theoretic measure of complexity (Pressman 1987). The metric was developed as an aid for testing and maintaining computer programs and is based on quantifying the decision structure within a computer program. The cyclomatic number V(G) of a graph is:

$$V(G) = e - n + p$$

where e is the number of edges,
n is the number of vertices and
p is the number of connected components.

The theory is based on the assumption that each program has one entry and one exit point. The complexity of a graph is defined as 'the minimum number of paths that can, in (linear) combination, generate all possible paths through the module' (Watson and McCabe 1996, p11) or less formally the number of independent paths through a graph.

There is a problem with the graphs in this research, because although they can be converted to control-flow graphs, they differ from most computer programs or modules in that they typically have more than one entry and exit point. However, it is still possible to quantify all independent paths through the graph by adding together the counts for each entry point. Table 11.4 shows the complexity measure for the four versions of the diagram described in previous chapters for the portbilling application.

Diagram	**McCabe complexity factor**
BED	79
BSDv1	79
BSDv2	79
BRDv1	79

Table 11.4 McCabe's Complexity Factor for each Version of the Diagram

On reflection, perhaps it is not so surprising that the cyclomatic complexity would be the same for each version of the diagram. This is because the number of independent paths is determined by the inherent logic of a business application and not the notation nor constructs chosen to represent it. However, this observation raises an important issue in terms of using such a metric to manage research progress. This is to what extent does McCabe's metric relate to the ability of an individual to **understand** and successfully work with a diagram. The assumption made is that the **number of independent paths** in a graph is directly proportional to the degree of difficulty one individual would have in **understanding** a diagram. Such an assertion is in my view incomplete. It takes no account of how easy it is for the individual to identify with the nodes (ie constructs) on the graph or indeed the number of different types of nodes on a graph. Recall in an earlier chapter, analyst X observed that he preferred to look for states first and then to identify events. This is clearly not reflected directly in McCabe's complexity factor.

Though not as well established as McCabe's metric, Rossi (1997) introduces a complexity metric which does take into account the 'different constructs' in a diagram. Based on the work of Rossi and Brinkkemper (1995), the metric seeks to take account of the inherent complexity in a diagram based on the number of different object types in a diagram, the number of different ways the object types may inter-relate and the total number of properties of object types in a diagram. The notational complexity of a diagram is defined as follows.

$$C = \sqrt{(X^2 + Y^2 + Z^2)}$$

Where C is the complexity,

X is the number of object types in the diagram,

Y is the number of relationship types in the diagram and

Z is the number of properties in the diagram.

Table 11.5 summarises the calculations for the diagrams in the first action research study.

Diagram	No. of Objects	No. of Relationships	No. of Properties	Complexity
BED	1	5	1	5.20
BSDv1	1	7	1	7.14
BSDv2	1	9	1	9.11
BRDv1	2	8	2	8.49

Table 11.5 Rossi's Complexity Factor for each Version of the Diagram

Some interpretation was required in the construction of table 11.5. The table was populated by examining the specification of each diagram as described in appendix 1. With regard to the BED, the number of relationships was considered to be five. Four of these were the four types of dependency identified between events, ie direct dependency, multiple direct dependency, selective dependency and contingent dependency. The fifth relationship was the alternative point of entry phenomenon. Clearly with the BED, there was only one type of object namely the event. The number of properties in a diagram is the total number of properties across all objects. In a dataflow diagram for instance, the process may have the properties of name, number and location, ie three properties. In the BED, there was only one property, ie the name of the event. With the BSDv1, the pattern was similar except that the number of relationships was extended to seven. This was because the four types of dependency were extended to six, ie to include complex predicate dependency and fan-out dependency. With the BSDv2, two further types of relationship were added. These were the symbols for simultaneity between states (ie the double-headed arrow) and the Harel blob. The Harel blob was considered a relationship type because it was a means of connecting states together, ie a relationship type. The BRDv1 contained two types of object, ie the state and the event and also it had two properties, ie one name for each type of object. The number of relationship types came to eight as follows. The phenomena of alternative point of entry, fan-out dependency, contingent dependency, simultaneity and the Harel blob were all retained making five types thus far. In addition three further types were defined as the simple event, the complex event and the complex multiple event.

In summary, there was an element of subjectivity in the way that the calculations were performed. For example, it could have been argued that the Harel blob was an object in its own right rather than a type of relationship. Further, no account was taken of the prerequisite table or indeed other diagrams used by the technique such as object life histories. Nevertheless, Rossi's complexity factor did provide a basis for comparing the inherent complexity of versions of the diagram, the underlying argument here being that the number of object types, relationship types and properties combine to influence the complexity of a diagram. In each case the complexity factor calculated was less than ten which compared favourably to the figures Rossi produced for different object class diagrams which ranged from about 13 to 21. In other words, in terms of Rossi's complexity factor each version of the diagram was simpler than the kinds of diagrams used by analysts undertaking object-oriented modelling.

Rossi's metric I shall refer to as the generic complexity factor of a diagram because it is calculated based on the number of different types of constructs and related concepts that a diagram may contain. McCabe's metric on the other hand is dependent on the inherent logic of a particular application and is therefore termed the application complexity factor.

In the next study, it is proposed that both metrics are incorporated into the intellectual framework. They will be used to help evaluate complexity and through evaluation it is hoped to better manage complexity. Interestingly, for computer programs, McCabe suggests an upper limit of 10, so clearly a factor of 79 would be considered extremely complex and therefore very difficult to understand especially for users.

11.2.3 Process (P)

Table 11.6 summarises those phenomena which had no sense of process steps.

Research Code	Description
1P2	Object oriented basis
1P4	Delay analysis step
3R4	Construction from complex objects
4R5	Direct vs indirect construction
4D4	Direct approach

Table 11.6 Research Codes Classified under Process (P)

This phenomenon is shown on figure 11.2 as spanning over all design worlds which is why the incorporation of transitive mappings into figure 11.2 makes the problem a little easier to illustrate. However, figure 11.2 is still weak in my view because such a diagram would not show alternate routes for developing a diagram well and therefore make it more difficult to analyse issues associated with routes or process steps. A richer model was required which explicitly showed steps and alternative routes through a development activity.

Glasson's model of systems evolution (1989) is a model which was considered suitable for adding to the intellectual framework. The model involves three concepts namely system evolution, states and deliverables. The concept of system evolution allows for the existence of different design worlds in W+W's terms, but further it allows for flexibility in the routes that are taken by developers. This concept is illustrated shortly. A state of an information system (in terms of its evolution) describes its position in relation to its evolution through the systems development. So at any given moment, an information system can be described as in a particular state. States may be decomposed into deliverables which are specific outcomes of development work eg, a screen or a diagram. Indeed, deliverables if large may be decomposed into sub-deliverables. Figures 11.3 and 11.4 illustrate these concepts in terms of the BRDv2.

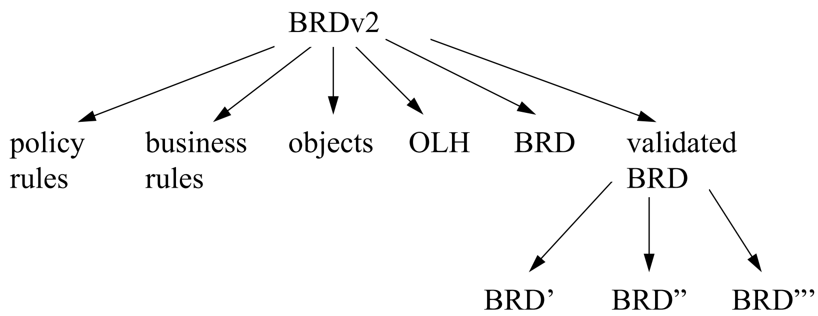


Figure 11.3 Structural Decomposition in Glasson's Model

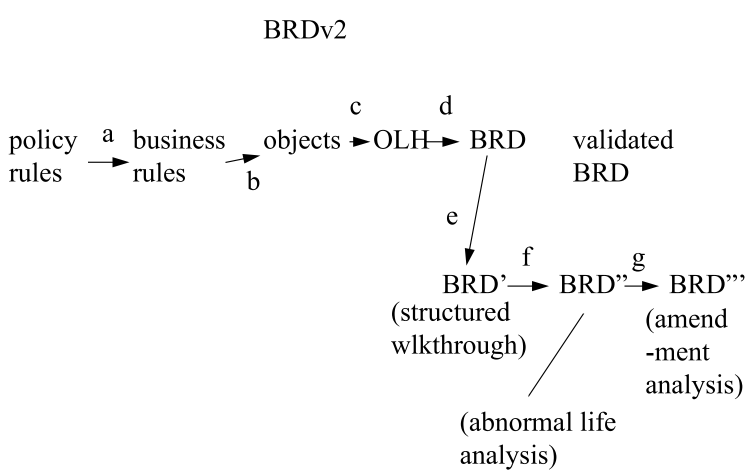


Figure 11.4 Development Sequence in Glasson's Model

In figure 11.3 the relationship between state, deliverable and sub-deliverable is shown. The BRDv2 is considered a state in Glasson's terms, ie either it has been constructed and therefore exists or it does not exist. If it has been constructed it implies that a number of deliverables have been constructed. Further, in figure 11.3 the validated BRDv2 (ie a deliverable) is decomposed into three sub-deliverables. These sub-deliverables represent versions of the BRDv2 which have undergone a structured walkthrough, abnormal life analysis and amendment analysis respectively. Figure 11.3

can therefore be seen to be able to describe aspects of the structure of deliverables in information systems development generally. However, that particular version of his diagram does not define the sequence of tasks. Figure 11.4 shows one possible sequence for developing the BRDv2. Indeed, this was the sequence suggested in appendix 1. Here arrows are annotated so that the sequence of constructing these deliverables is shown. Other sequences are possible. For instance, business rules may be identified before policy rules or even at the same time. Such variations are more easily shown on this type of diagram. Further, it is also possible to show situations where alternative deliverables are used in the development process. For example, earlier it was suggested that either a data model or an object model could be used as the source for creating life histories. Again, in principle, this type of diagram would have no difficulty in depicting alternative models as well as pathways.

In summary, therefore Glasson's model (1989) was able to describe a specific development process by defining the sequence of deliverables produced in that process. Alternative routes of development or alternative models used could also be depicted. So it was decided to incorporate this model into the intellectual framework. Further, the model was also able to describe aspects of the structure of deliverables in terms of their component deliverables.

Glasson's model appears to complement the W+W framework. Where questions of sequence of process arise, clearly Glasson's model is proposed as a way of conceptualising the issue; where questions arise concerned with the mapping between constructs in one design world to another, then the W+W framework appears a more appropriate vehicle for dealing with this. The next study will provide an opportunity to test the complementarity between the models as well as the ability of Glasson's model to deal with any process issues that arise.

11.2.4 Abstraction (A)

Table 11.5 shows issues which were classified under abstraction.

In order to conduct a review of abstraction issues they have been further categorised into sub-categories. These sub-categories are completeness, modelling focus, levels of business rule and other abstraction issues. The sub-categories were obtained from a content analysis of the codes classified as abstraction issues in table 11.1.

Completeness

This sub-category of issues refers to whether the constructs in the ontological world were considered to form a complete specification. Concern over this was raised in each iteration of the diagram (1R2, 2R2, 3R2 and 4R2) and indeed over the four diagrams a trend in adding new constructs was detected (3D1, 4D3).

Having thus detected a trend it raised the question about whether there was any other construct that might logically be argued to be necessary for a complete description. There was an argument for including a message construct and so it was decided to incorporate this into the next study in order to test the argument. The argument for this is summarised below as a review issue and is documented as a research progress item.

Research Code	Description
1P1	event focus
1P3	group event notation
1A3	alternative point of entry
1R1	IS events vs business events
1R2	rule completeness
1R3	state vs event
2A2	overkill
2R2	rule completeness
2R3	manyness
2D1	extend prerequisite table
2D2	extend prerequisite table
3R2	Simultaneity
3R3	rule completeness
3D1	states and events
4R2	rule completeness
4R3	IS rules vs business rules
4R4	manyness
4D1	show conditions and triggers on diagram
4D3	refine definition of rule
5P1	business rules versus information systems rules
5P2	trigger selection
5P4	sub-states and parallelism

Table 11.5 Research Codes Classified under Abstraction

5R1: Adding Messages to the Business Rule

In considering the business rule components of state, event, condition and trigger, there was a sense in which they were incomplete. The trigger can be viewed as an input signal to the system. Would the opposite (ie output signals from the system) also not qualify as part of a business rule? Figure 11.5 illustrates two different business rules.

In the first case the business rule is that lines are deleted from an order and a message is sent back to the customer to that effect. In the second case no such message is sent. Messages (ie signals leaving the system) constitute part of a business rule because they indicate part of the policy of the organisation in this regard. This allows business rules such as ‘customers are sent a list of those order lines deleted from an order’ to be captured in the diagram. It was therefore decided to include a message construct as a component of the business rule (5D1).

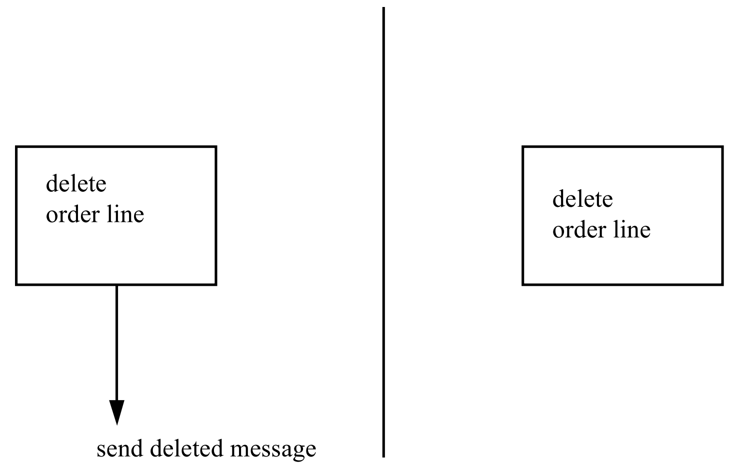


Figure 11.5 Two Different Business Rules for Deleting a Message

Modelling Focus

This sub-category refers to what the focus of the model (eg state, event etc.) is considered to be in the ontological world. In some respects this topic overlaps with the completeness topic above. However, discussion here contains further analysis.

Initially in 1P1, events were the focus of the diagram. In 1R3, a question was raised as to whether states rather than events were a better candidate for modelling. Next in 3D1 a more pluralistic approach was considered in terms of modelling both states and events. Later conditions and triggers had been incorporated as basic modelling constructs (4D1) and in the above sub-section messages had been provisionally added (5D1). However, since there had been a shift of emphasis towards modelling business rules an important question arose as to whether the same modelling focus and ontological constructs were still relevant. Expressed in terms of the W+W framework, if the ontological world has changed (and clearly it might have to since originally the purpose of the model was to aid information systems specification and now it was to describe business rules) then it would be necessary to review the ontological constructs in the new world (ie of business rules) in case they were inconsistent with the earlier world.

The literature review chapter had demonstrated that there is a wide diversity of opinion on what constitutes a business rule and therefore what might constitute the constructs of its ontological world. However the list of types of business rule introduced as anecdotal rules in chapter 2 could be considered a starting point for establishing whether typical business rules could be modelled using constructs in the ontological world and this is in fact what was done.

By reviewing the types of business rule in the BRDv2 paper, it was found that four of the five rules could be abstracted into the ontological world. Indeed, the first four rules had been depicted in the BRDv2. These rules were that customer orders may be received by telephone or post, that part-payment for invoices is permitted, that credit orders are granted if credit balance is greater than or equal to order value and that only good customers may be granted credit orders. The fifth rule however was unable to be defined in terms of existing ontological constructs. This rule was that one order may have many invoices. This led to documenting a further review issue.

5R2: The One to Many Business Rule

The BRDv2 could model all types of examples of business rules outlined except those of the form 'one instance of object X has a relationship to many instances of object Y' eg one customer may have many orders. While such rules could be depicted on a static object model through showing the cardinality of object relationships, it was decided to try to show this on the next version of the diagram in some manner (5D2) so that it could be argued that the BRDv3 was a repository for all the types of business rule identified in the academic paper.

Levels of Business Rule

Another sub-category of abstraction that emerged concerned levels of business rule. In 1R1, the issue of whether IS (information system) events or business events were to be modelled was raised. Later in 4R3 a similar discussion took place this time concerning

IS rules as opposed to business rules. In 4D3 a decision was taken to refine the definition of a rule to distinguish policy rules from others and in 5P1 three levels of rule were identified namely, policy, processing and implementation rules.

Figure 10.4 illustrated how these three levels related to each other. In terms of the extended W+W framework, these three levels represent three design worlds. The ontological world is mapped onto a business policy world and from there to a processing world and so on. However, figure 10.4 is incomplete in the sense that no account has been taken of the elaboration problem discussed earlier. Here, the processing world contains details of how each event is actioned. Such information is not in the business policy world but would have to be in the ontological world in order to generate an accurate processing model. Since the elaboration problem was considered to be a common occurrence, hereafter figure 11.6 will be used as the archetype for discussing future problems since it contains an example of the elaboration problem.

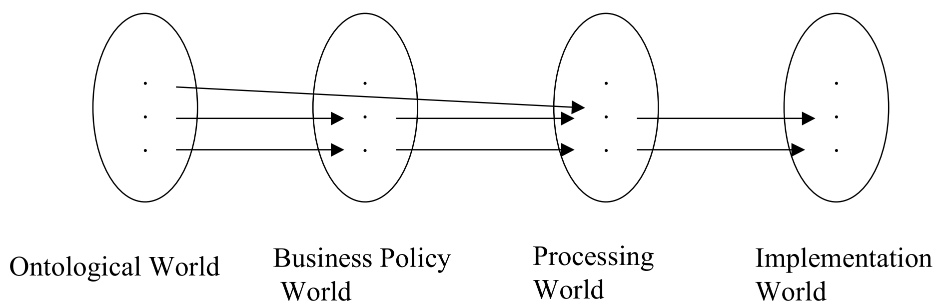


Figure 11.6 Extended Model showing Elaboration Problem

Other abstraction issues

The remaining issues in this category refer to whether fairly specific ontological constructs require to be abstracted into the ontological world. Compared to the modelling focus sub-category the issues under this sub-category do not question the validity of the modelling focus; rather they seek to provide a complete and consistent view of the modelling focus. These issues are the group event (1P3), alternative point of entry (1A3), overkill (2A2), manyness (2R3, 4R4), show conditions and triggers on diagram (4D1), trigger selection (5P2) and sub-states and parallelism (5P4). While it is clear that the W+W framework in itself did not provide specific guidance on these issues, figure 11.2 certainly provided a location for them and thus a starting point for analysis.

11.2.5 Notation (N)

Table 11.7 shows the research codes which were classified under notation.

Research Code	Description
1P3	group event notation
1P5	event notation refinement
1A3	alternative point of entry
1A4	fan out dependency
1A5	contingent dependency
1A6	simultaneity
2P1	simultaneity
2A1	simultaneity
2D1	extend prerequisite table
2D3	Harel blob
3R2	simultaneity
3R3	rule completeness
4A1	simultaneity
4A2	readability and the simple condition
4R4	manyness
4D1	show conditions and triggers on diagram
4D2	prerequisite table
5P2	trigger selection
5P3	Harel blob for simple conditions
5P4	sub-states and parallelism
5P5	event specification guide

Table 11.7 Research Codes Classified under Notation

In table 11.1 occurrences of notational issues appeared either by themselves or in combination with other codes. Codes 1P5, 1A4, 1A5, 2P1 and 4D2 were all codes where the notational issue was a straightforward one involving no other type of research code. Here the problem was simply one of how best to depict a design construct graphically. With regard to the other codes in table 11.5, these codes appeared in combination with other codes. The reason for this was that there appeared to be some doubt as to the exact nature of the problem. For example, research code 1A3 was the alternative point of entry problem. As an issue it raised questions in terms of abstraction, mapping and notation. That is whether it should exist in the ontological world (ie an abstraction issue), whether there existed a mapping to the design world (ie a mapping issue) and whether a suitable notation existed for it.

Mostly, the considerations of notation were relatively straightforward compared to deciding for example whether a construct ought to exist in the ontological world. One exception to this was the recurring problem of showing simultaneity. At times, for example in 2A1 (simultaneity), the problem was that no obvious way of showing simultaneity with states appeared available. Although the W+W framework in itself did not provide support for dealing with notational issues, I was comfortable that notational issues could be satisfactorily dealt with as they arose in the research without the necessity of acquiring additional theoretical support by extending the intellectual framework at this point.

11.2.6 Summary of Framework Review

Overall the choice of the W+W scheme as the initial basis for the intellectual framework F was considered appropriate. Not only had it helped to conceptualise specific mapping problems but it facilitated the construction of an extended model. Analysis of the extended model had provided a means for identifying categories of intellectual ideas (ie artefacts) that were missing and thus strengthened the view that the W+W framework was a good initial framework for this kind of research.

As a result of this review, two additional artefacts were added to the intellectual framework for use in the next study. These were the notion of measuring complexity and Glasson's model of systems evolution.

11.3 Review of Methodology M

To recap, the methodology here was a form of action research study which involved working jointly with an analyst in drawing a diagram using a real application that was very familiar to the analyst. The role of the analyst was that of co-researcher. He was involved in drawing the diagrams and in discussions on how well the diagram had managed to capture the essence of the application. Though responsibility for the detail of each version of the diagram was clearly mine, the opinions of the analyst were an important part of the discussions leading up to the creation of a new version of the diagram.

I consider that the process of involving an analyst using a real application and then working through several iterations to be a powerful way of developing new diagrams such as this one. In this study four versions of the diagram had been created in this manner.

There are two main strengths in this cyclic approach. These I shall term incremental re-inforcement and incremental anomaly handling. In incremental re-inforcement, later versions of the diagram reconfirm the conclusions or solutions of earlier versions. For instance, the alternative point of entry solution remained a solution throughout all subsequent versions and so this re-inforced the view that the solution was an appropriate one.

With regard to incremental anomaly handling, the cyclic process also allows for the testing of possible solutions where problems or anomalies are identified. One of the characteristics of action research described by Dick (1992) is that when anomalies or problems are found there is the opportunity to investigate them and correct them. The steps involved are examine anomalies, hypothesise, test in later phases and re-evaluate. Together these two incremental tactics contribute to the strength of conviction that can be placed on the latest version of the diagram in terms of the worthiness of the characteristics it embodies.

Overall, the methodology was considered robust for addressing this type of research question. However, two aspects warrant further discussion below. In the first sub-section, a distinction is made between the execution of the methodology itself and the ontological ideas being tested in the methodology. In the second sub-section, comments are provided at a more pragmatic level. They relate to reflections about applying the methodology. This is because a future study was planned and any lessons learnt may therefore be able to be applied to the future study.

11.3.1 Distinction between Methodological and Ontological Ideas

Throughout action research study 1, the methodology (M) had been the location for recording the constructs in the ontological world. Since it had been anticipated that the research was to confirm the original software engineering ideas, considering these detailed ontological constructs as part of the methodology had served to make them distinct from the framework of ideas and indeed had caused no practical difficulty in its application. However, philosophically, ideas about the ontology being tested were quite different from the methodology itself. Because of the importance of ontological ideas to the whole research effort, it was desirable to separate them quite distinctly from methodological issues. So it was decided to put ontological matters under the rubric of the framework of ideas (F) in order to more clearly distinguish methodological issues from ontological issues.

11.3.2 Reflections on the Application of the Methodology

Below, a number of different reflections on using the methodology are provided with a view to improving its application in future studies. In particular, a review of the process and tactics of interviewing the analyst takes place.

The semi-structured interview was divided into two main parts. In the first part questions were asked about the features or characteristics in that particular version of the diagram. These questions were asked in the first part of the interview in order to get the analyst's view without it being coloured by some of the issues which were to be raised in the second part of the interview.

In general I found that the early open-ended questions did not elicit the kind of responses which confirmed or shed light on a particular pivotal issue. While it was re-assuring to get positive feedback, this did not in itself provide substantive insight or alternatively contention. The questions in the last stage of the interview, especially follow-up questions, seemed to allow me to get much better insight from the analyst. Of course, inherent in this observation is the question of to what extent a question, however neutral it may seem, guides the interviewee into a particular frame of thinking. This is why general open-ended questions are favoured at least at the beginning of an interview.

The question of how willing an interviewee is to offer comments is important here. If the interviewee is prepared to offer deep observations from the open-ended questions then this provides a good argument for the weight that can be attributed to a particular point. However, where the interviewee is not offering deep or unsolicited observation, it may be necessary to probe further at the expense of guiding the interviewee along certain lines.

Although clarifying points from the diagram the analyst had drawn evoked discussion on many worthwhile issues, I judged it better in future interviews to handle clarifications outside the interview itself. If there has been an oversight on the part of the analyst, it tends to create a tension during the interview and perhaps even a desire to justify the oversight in some way rather than simply acknowledge it for what it is. However, given that clarifications are cleared up prior to the meeting, where such discussion has evoked an important issue, there is no reason why that issue can't be aired in the latter part of the interview if it hasn't already arisen in earlier discussion.

In the development of the BSDv2, a deliberate decision was taken to work more interactively with the analyst. Instead of preparing a paper as a benchmark to work from, the diagram was developed while working with the analyst. Whilst this is only an impression, my feeling is that this latter method was a better way of working. With the prepared paper, the ownership of ideas was mine and the analyst adopted a relatively passive role in terms of essentially confirming or denying my assertions. However, working with the analyst interactively made for much more sharing of ideas and therefore more joint ownership of ideas, albeit that most new ideas were still initiated by myself. So at least where the study is exploratory as was the case here, I consider it of value for the researcher to move towards an interactive relationship with co-researchers as quickly as the circumstances allow. This also explains why there is no preparation column for the BSDv2 and BRDv1 in the research progress map. However, this doesn't take away the need for preparation in those action research cycles, it simply changes how and when the preparation is undertaken.

The interview was undertaken at the end of the study after several iterations of the diagram had been completed. On reflection a better tactic would have been to have conducted interviews once each version of the diagram had been completed. This would have had two advantages. Firstly, the problems and issues concerned with that version of the diagram would have been fresh in both the analyst's and researcher's minds. Secondly, there would have been no 'hindsight' effect. The 'hindsight' effect in the context of this study is where the analyst had the benefit of the ideas and solutions incorporated in later versions of the diagram when responding to questions on earlier versions of the diagram. When this study was planned, it was not anticipated that there would be major revisions to the diagram or indeed a shift of focus and so the need for interviews at the end of each version was not envisaged.

An important outcome of this study were two conclusions that emerged during the action research process, ie that the focus of the research should shift to capturing business rules and that business rules comprise states, events, conditions and triggers.

The insight afforded by examining real-world context should not be under-estimated. In this study for example, the existence of status entities led to the speculation of the desirability of separating business rules from information systems rules. I doubt personally whether an approach such as laboratory testing would have revealed such an issue.

Though action research is a very flexible approach, it was not possible to undo events or do things differently once done. This was so in respect of the realisation that the analyst should have been interviewed at the end of each cycle. Because the analyst had been 'contaminated' by later ideas and solutions, the significance that could be attributed to his responses was therefore diluted. However, this clearly would not be a problem in future studies as long as interviewing takes place after each cycle. Overall while this review had identified a number of detailed improvements to the methodology, it is concluded that the general approach involving an analyst as co-researcher in the manner described is a sound one.

11.4 Review of Area of Application A

The lack of ability to generalise from this study is apparent. In the study, only three applications or cases were used. One was the researcher's own case study. The other two were real-world applications. A number of problems emerged in one application but not the others. For instance the problem of requiring an audit event for every portbilling event did not surface in the case study. Given the diversity of the domain of information systems applications, it is obvious that many more applications would be required before any degree of confidence that most or all business rule situations had been identified and therefore addressed. Consequently this was seen as a weakness in the study and one reason for seeking a new application.

However, there was another reason for changing the application. This was to do with involving users. It had always been planned that users would be involved in working with an analyst using the diagram. One major concern that had emerged was to do with the perceived complexity of the emerging diagram. I considered it hard enough for analysts and virtually unacceptable for users. However, there was a possible resolution to this and this is documented below as a review issue.

5R3: Separating the User and Analyst Diagrams

Reflection on this issue suggested there was a strong case for developing two separate diagrams. One diagram would be for the users to work with and the other for the analyst to work with. The user's diagram would be different in two major respects. Firstly, unlike the analyst who would require a picture of the whole system, each user or group of users would be limited to a single business rule on any one diagram. Secondly, it could be simplified for example by removing the event specification guide and other constructs arguably more needed by the analyst. It was therefore decided to implement this approach in the next version of the diagram (5D3). This was seen as assisting in the resolution of 4R1 (complexity).

Clearly, to test this proposal it was necessary to work with users and so coupled with the limited number of previous applications used pointed to the need to identify another area of application which would involve both analyst and user.

In the first action research study, it was decided to select an existing application with which the analyst was familiar, to allow focus on the specification of requirements (rather than the gathering of requirements). It had been planned in a later study to extend this research by using the diagram in the information requirements determination phase of a new information system. However, over the course of study 1, there had been a major shift of emphasis from a specification problem within systems development to modelling business rules. Indeed the current version of the diagram contained many constructs not in the original version. For this reason, it was decided to look for an application where it would be possible once again to focus on requirements specification (this time of business rules) as opposed to requirements gathering. An ideal area of application here would be one in which business rules were needed and used, yet where there was no formal recording of such rules.

11.5 Summary

This chapter has reviewed a model used throughout action research study 1 to frame and guide the research. The components of this model were an intellectual framework (F), an area of application (A) and a methodology (M) which was used as a vehicle for the testing of ideas. Each of these components were reviewed in this chapter.

The intellectual framework utilised was taken from the work of Wand and Weber (1993). It involves mappings onto scripts and mappings from an ontological world to a design world. This framework was regarded as highly appropriate to the original research question. The review of the framework was undertaken by identifying an extended model of W+W's model. Issues identified in previous chapters were categorised into one or more areas of the extended model. The review vindicated the original choice of framework as one which provided a good source of insight into specific problems that occurred. From the review it was also concluded that the extended W+W framework was a good vehicle for highlighting other kinds of problem. Two main decisions were taken regarding updating the intellectual framework. These were to develop ways of measuring complexity so that complexity could be better managed and to use Glasson's model of system evolution to better explain issues concerned with process and alternative routes of development.

With regard to reviewing the methodology as an instrument for exploring the research question, a number of observations were made. Firstly it was noted that the flexibility of the cyclic process affords the opportunity to solve problems and increase confidence in solutions. Secondly, in order to make future evaluation more straightforward, it was also decided to include ideas about the ontological world as part of the framework of ideas. Arguably, this is where ontological ideas more properly belong. Thirdly, in terms of the mechanics of the action research process, it was observed that more valuable feedback was obtained in the directed questions at the end of the interview and that clarifying points in an interview about the detail of diagrams is not recommended.

Fourthly, with hindsight a note was made that interviews should be conducted at the end of each iterative cycle. It was relevant to review lessons learnt in this phase of the project because a further action research study was planned and so where appropriate the lessons learnt here could improve the quality of future studies.

With regard to the area of application, it was concluded that there were too few cases used to generalise about the robustness of solutions proposed which pointed to the need to seek further applications to study. There was also a need to involve users in working with the diagram and this further strengthened the conviction to seek a new application area.

Over this critical review, a number of decisions were taken in terms of changing the diagram, ie affecting future research progress. These were to:

5D1 add a message construct to business rule,

5D2 implement to-to-many business rule and

5D3 implement a new user diagram.

This necessitated a re-writing of the latest version of the diagram.

Chapter 12

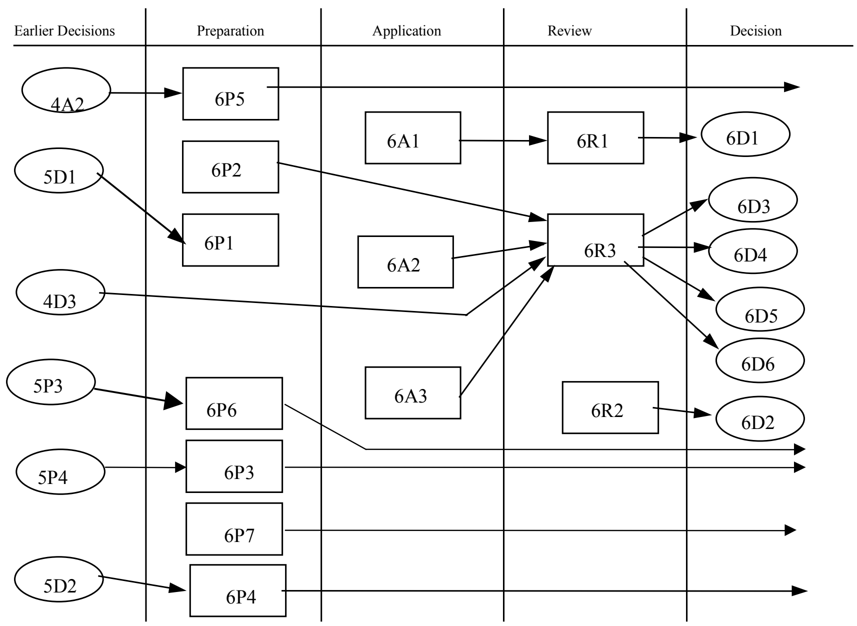
Action Research Study 2

12.1 Overview

As a consequence of review decisions in the previous chapter, the BRDv2 paper was rewritten to incorporate changes. Thus the new paper was referred to as the BRDv3 (Business Rules Diagram version 3) paper and was reproduced in appendix 1. Discussion in this chapter follows the action research cycle steps of preparation, application, review and decision used in the previous chapters. The first section outlines the main goals of this study and describes the structure of the study. This is followed by a section on client/system infrastructure, ie it provides some background to the actors in this study and the business application used. The next three sections discuss the action research cycle steps of preparation, application and review in detail. In the next chapter a final review of all of the conclusions from the whole of this research takes place, so in the final section only decisions which directly resulted from this study (ie action research study 2) are documented. Figure 12.1 summarises discussion in this chapter with a research progress map for this study.

12.2 Goals and Structure of Action Research Study 2

The principal goal of this action research study was to extend the focus of the research by working with potential users of the diagram. There were two ways in which this extension of focus took place. Firstly, I wanted to examine how an analyst might use the technique in practice. In the previous action research study I had led the research in the sense that I was directly interacting with the co-researcher.



Legend:

4A2	readability and the simple event
4D3	refine definition of rule
5P3	Harel blob for simple conditions
5P4	sub-states and parallelism
5D1	add message construct to business rule
5D2	implement one to many business rule
5D3	implement user diagram
6P1	categorising events, triggers and messages
6P2	flowchart 'look and feel' to the UBRD
6P3	the abolition of sub-states
6P4	the one to many business rule
6P5	the abolition of the simple event
6P6	refining the Harel blob
6P7	changes to the event specification guide
6A1	user-led UBRD construction
6A2	flowchart loops in the UBRDs
6A3	processing rules in the UBRDs
6R1	the need for guidance
6R2	the need for validation
6R3	the level of abstraction problem
6D1	guide users in construction of UBRD
6D2	retain steps for validation of diagram
6D3	clarify policy rule
6D4	change notation for sequentiality of policy rule
6D5	retain the pure BRD and table
6D6	flexible approach for UBRD

Figure 12.1 Research Progress Map for Action Research Study 2

Questions such as how an analyst might integrate the BRD into his or her work with users would be better answered if I took a more passive role. I ensured that a systems analyst was adequately trained in the technique by providing him with a copy of the paper and meeting with him to discuss or clarify aspects of the technique. The analyst was then expected to work with users developing the diagram. My role here was to support the analyst in this endeavour. I was not present at the meetings with users. However, I met with the analyst after these meetings and discussed any issues that had emerged.

Secondly, it had always been the intention that users would be expected to work with the diagram in some manner. A decision in the last chapter had been made to develop a separate diagram specifically for users. Clearly an important part of this study was to test the extent to which users were able to work with such a diagram.

The structure of this study involved an analyst and two users working on a business application. Working with the users the analyst attempted to create a UBRD (User Business Rules Diagram), ie a user-oriented diagram along the lines in the prepared paper. The analyst was asked to prepare a BRD (Business Rules Diagram) and accompanying EST (Event Specification Table) from UBRDs created in the sessions with the users.

At the end of this study the analyst and users were interviewed by myself and the transcripts of these interviews were used as one input to data analysis. Further, minutes of meetings held between the analyst and users were also used as input to data analysis as well as any diagrams produced.

The analysis of the study (ie the review step) was divided into three passes. In the first pass, feedback from the users was obtained by analysing their comments and comments from the analyst. The questions put to the users centred on aspects of the UBRD, so much of the thrust of the first pass reports on the issues and problems of incorporating the UBRD within the BRDv3. Further, it had been decided that the analyst was to be given access to the users' transcripts. The reason for this was to allow the analyst to comment on the users' feedback. Since the analyst had been present at these meetings and I had not, this approach provided an additional source of feedback on the process. Comments from the analyst were also included where these comments had a direct bearing on what the users had observed or stated.

In the second pass, the focus was on the BRD as a whole rather than only the aspects concerned with the UBRD. Since many of the features of the BRD concern the systems analyst working with the diagram without users after the UBRDs have been constructed, a separate pass examining issues and problems from the perspective of the systems analyst was considered appropriate.

The first two passes report the comments and observations of the participants. I have simply summarised their comments without attempting to synthesise broad conclusions. Detailed description of the first two passes appears in appendix 8. The third pass brings together what has been reported in the first two passes by identifying main issues or problems that emerged. For each main issue or problem identified there is argumentation to support the conclusions reached. The argumentation uses material from the semi-structured interviews, the minutes of the meetings held by the participants and the diagrams produced by the analyst as well as types of cognitive map summarising the thread of each argument.

The three pass tactic reflects the sequence in which I actually undertook analysis of this review step. Recapitulating the sequence of the three passes here demonstrates the process by which these issues and conclusions emerged. However, since the main argumentation is provided in the third pass, the description of the first two passes has been removed to appendix 8 as stated earlier.

12.3 Client/System Infrastructure

In this study three actors were involved excluding the researcher. These were one systems analyst and two users, hereafter referred to as analyst Y, user A and user B. The roles of the users were simply those of domain experts. They were made aware of the nature and purpose of the research and had agreed to take part in the study. An application had been chosen in which it was known that the business rules were not formally recorded in a central repository and the users themselves were aware of this.

> ‘Some are recorded as policies in the policy manual. Some of the business rules are not really recorded, probably in an ad-hoc fashion on the backs of forms and these sorts of things. In a few of our prospectuses a lot of these processes are involved. For the staff it’s

> just a matter of learning them on the job. So they are not really recorded in the one place. No, not all of them.' (User A).
>
> 'I think that's something that needs to be worked on in the future. There will be a policy manual for our office, but at this stage I think things are pretty haphazard. I'm not sure to what extent we do have things documented. I know that user A was working on some ISO9000 documentation, so I don't know if he flowcharted that. But other than that there isn't anything documented.' (User B).
>
> *Are they static or do they change a lot?* (researcher).
>
> 'Yeah, they're static they don't really change too much.' (User A).
>
> 'The rules are fairly static yeah. It's normally the procedure that changes.' (User B).

The role that the analyst was that of co-researcher. He was trained in the technique and then asked to use it in his organisation. Though he was knowledgeable about the business application himself (since he had worked with it), the primary question he was asked was whether the BRD could be used for capturing business rules especially in terms of the dynamics of user-analyst interaction. The analyst was asked how best to use the technique with users and what changes if any to the structure of the technique were warranted based on his experiences. Like the previous action research study, the analyst was not made aware of details of the (revised) intellectual framework being used.

12.3.1 Analyst Y

Analyst Y is a systems analyst employed by Curtin Business School which is a division of Curtin University of Technology. Analyst Y had twelve years experience in the computer industry and is familiar with popular approaches to systems development including dataflow diagramming, entity-relationship modelling and so on. He has an interest in formal methods and at the time of writing had just submitted a Masters thesis (part-time) on methodological selection.

12.3.2 User A

User A had about two and one half years experience in Curtin Business School. His job was totally concerned with the enrolment and registration of students. This involves preparing timetables and classlists as well as assisting in deciding on the process by which students enroll themselves.

User A has experience with personal computers going back to the eighties. During the planning stage of a new student records system, he was exposed to entity-relationship diagrams.

12.3.3 User B

User B's job title is that of undergraduate enrolments co-ordinator. She had been in that role for almost a year though had been connected with the Business School for about three years.

User B also has experience with personal computers. She has had some exposure to flowcharts in which she was involved in diagramming existing procedures.

12.3.4 The Business Application

The business application which is used in this study is the student enrolment system for the Curtin Business School. Each semester some 6000 students enroll in undergraduate and postgraduate courses. Students may be full-time or part-time. Their individual circumstances such as when they can enroll or when they can attend sometimes put quite difficult constraints on the enrolment system. There are also problems with limited availability of classroom and laboratory facilities. Thus the enrolments system is both a key system for the smooth operation of the Business School as well as a system which is constantly under some pressure and focus.

The scope of the student enrolment system covers all activity from sending appointment letters to students that they should begin the enrolment process through to actually recording their registration in specific lecture, laboratory and tutorial rooms. The student enrolment system interfaces heavily with a University-wide student records system which contains information about students' academic status including what subjects they have previously passed.

Student enrolments have been computerised in the University since the seventies, but the system in use in the Business School and currently known as the student enrolment system has been available on computer since 1993. In this system PCs or dumb terminals access data on VAXs in the University's main computer centre.

12.4 Preparation of the Business Rules Diagram version 3

Seven issues emerged during the preparation of the BRDv3 and these are now discussed.

12.4.1 Categorising Events, Triggers and Messages (6P1)

In the BRDv2, it had been proposed that candidate events could be identified by brainstorming. In the BRDv3, the same suggestion was retained. However, with the addition of the message construct, there was even more opportunity for users to confuse events, triggers and messages. Indeed, when constructing the list of events, I had to reconfirm to myself sometimes the distinction between say a trigger and an event. For example the candidate 'receive payment' could be construed as either a trigger (ie that the receipt of a payment triggers an event) or an event (ie that receive payment is taken to mean the act of recording a payment which is an event). A simple extension to the end of the brainstorming session was written into the narrative which required the group to categorise candidates as events, triggers or messages thus avoiding ambiguity at an early stage and also perhaps facilitating the identification of other associated candidates which had been overlooked.

12.4.2 Flowchart 'Look and Feel' to the UBRD (6P2)

A decision to separate user diagrams from analyst diagrams (5D3) had been taken in the previous chapter in order to simplify matters for users. There were two ways in which the user diagram could be simplified. These were firstly, that each user diagram would be limited to a use case and secondly that some of the constructs could be removed, eg those more relevant to the analyst. It was a simple matter to rewrite the paper limiting each user diagram to a single use case. However, deciding specifically which constructs to omit required further deliberation.

The event specification table could be omitted because all of the basic information contained in the table was already on the diagram. The information that was not on the diagram, eg information to do with manyness was arguably more of interest to the analyst.

As regards the constructs on the diagram itself, I had some reservation about the use of the Harel blob. There was a case that could be made for using the Harel blob to show trigger selection since this enabled a specific type of business rule to be explicated on the diagram. However, I was less comfortable, as far as users were concerned, with using it to show simultaneity and complex conditions. With regard to simultaneity, this was an issue that could be dealt with by the analyst in the same way as manyness could be established. In other words, the UBRD could act as a communication vehicle between analyst and user. The analyst could ask questions of users by referring to the UBRD. The answers to these questions would allow the analyst to establish whether events were simultaneous or whether manyness existed between states. I therefore decided that simultaneity could be omitted from the UBRD.

With regard to complex conditions, on the one hand, it was necessary to model complex conditions, ie to establish from the users the full range of questions that had to be asked concerning a business rule. On the other hand a Harel blob around a complex condition could be visually overpowering to users. In addition diagrammatically it did not indicate a sequence of conditions if a sequence had to exist. The alternative to using a Harel blob for the complex condition was to use the condition symbol as it is used in a flowchart. Here a result of a binary condition may be an input into other binary conditions and thus a path emerges. Further, sequence of conditions would then be specified. I decided that users might be more comfortable with this kind of procedural approach to explicating conditions and so the description of the UBRD was written with a flowchart 'look and feel' to it.

One last change was implemented in the UBRD. In the BRDv3, a state is shown as connected to the complex condition if it is interrogated in one or more conditions within the complex condition. This can be regarded as redundant since a question about the status of an instance of an object is effectively repeated in the condition. For example, in the case study a state of 'good customer' would be connected to the complex condition for accepting an order. One of the conditions in this complex condition would be 'good customer?'. This 'redundancy' can be argued to be worthwhile on the full BRDv3 because it is a strong visual signal indicating the connection of states of instances of objects with subsequent events in other objects. Having said that, these connections are something that could be established by the analyst quizzing users through the medium of the UBRD rather than having to show such connections explicitly on the UBRD. It can be argued that as long as all relevant conditions are shown in the complex condition, the existence of connected object instance states is automatically implied. However, with regard to post-states, I contend that the situation is different. Here, it is still appropriate to retain showing any state changes (post-states) that emerge from the use case on the UBRD. This is because these formalise that something significant has occurred in the use case which will change the way the system will subsequently respond to stimuli in the future and so state changes (ie post-states) were retained.

12.4.3 The Abolition of Sub-states (6P3)

In issue 5P4 the idea of sub-states in object life histories and the BRDv2 was rewritten in order to accommodate parallel states. As the paper was being written here however, I recognised that introducing sub-states contradicted the concept of states that had been used since the early versions of the diagram. From early on a state only existed in the diagram if it changed the way an object was to be treated by the system. For instance an (unpaid) invoice and a paid invoice would be treated differently from the point of view that unpaid invoices could receive payments, have credit notes generated and so on. This would not be true for a paid invoice. If this was the role and purpose of a state, what then was the distinction between a state and a sub-state? Either a sub-state would or would not affect the way an object would be treated by the system. If it did then it was a state; if it did not then it was not a state. It was therefore concluded that sub-states were unnecessary on the diagram itself although it was accepted that the need to show parallel states on the OLHs (object life histories) remained since at that juncture a detailed understanding may not exist, ie at an early stage of requirements definition stakeholders may not have fully thought through the implications of their assertions (Vitalari 1992).

Moreover, it was recognised that it was quite possible to construct the diagram without the use of sub-states. For example, there would still be events receiving payments and generating credit notes but these would not change the state of the object. Only when the criterion of full payment was met would the state of the object be changed. In principle this was no different from a situation in which stock had been delivered but where the revised level of stock was still insufficient to meet the order. In such a situation there is no 'state change' because the system will not necessarily respond differently to a future signal. Here, not until the order quantity is known (ie when a trigger of order is received) would it be possible to decide whether there is sufficient stock to meet that order item. In both cases, the event only happens when the condition is met. One consequence of this line of reasoning is that some events may not change states at all. It was decided to incorporate this consequence in the definition of a business rule derived from this research.

The question then arose as to what to write in the paper. On the one hand, the argument provided in the paper did not prove that the use of sub-states to depict parallelism could be avoided generally. It merely demonstrated that in this case it could be avoided. On the other hand, neither was I able to prove otherwise in the general case. Having said that, as the case study illustrated there were situations where sub-states might be considered necessary on the OLHs but later were seen to be unnecessary on the BRD. This is because some assertions might not be thought through at an early stage in business rules modelling. On balance, I felt it wiser to avoid the need to show sub-states if at all possible. So, I elected to write the paper on the basis that sub-states would not be shown on the BRDv3. Clearly, this should be reviewed at a later stage. This is one strength of the cyclic action research approach, ie the assertions of earlier cycles may be confirmed or disconfirmed in later cycles.

12.4.4 The One to Many Business Rule (6P4)

Resolution of this issue demanded finding a solution which enabled rules of the form 'one object X is related to many objects Y' to be described. It had already been recognised that inferring such rules from data models was problematic. For example, in the case study only good customers may have credit orders. It is not possible to state this categorically by simply inspecting a standard data (or object) model which has a one-to-many relationship between customer and order. In such a case, since there is nothing to signify otherwise it would be quite reasonable to infer that all customers, not just good customers, may have credit orders which is incorrect in this case study. Unless sub-typing is used, there is no precise means of expressing a rule such as 'only good customers may have many credit orders' on a data model. An alternative for modelling this kind of rule would be to use states since such rules often apply to states of an object.

I speculated that the key to solving this problem was in observing a stereotyped difference between intra-object and inter-object state cardinalities. Typically within an object, the states it can occupy are 'singular'. This would be arguably true even for composite objects. For example, in the order processing case study an instance of a composite object such as order could occupy the states 'order', invoiced order' and 'paid order', ie one instance of object order does not have many instances of the state 'paid invoice' for example. This is what singularity is defined to mean here. However, for the most part, when dealing with inter-object cardinalities there is typically (though not always) a different situation which I term multiplicity. For example, when it comes to interpreting the rule 'one good customer can have many credit orders' on the BRD there has to be a mechanism which allows second and subsequent orders to be accepted while the customer remains in good status. This would mean that several object instances would be initialised into a particular state, eg credit order exists. This is different from singularity where second and subsequent instances of an object state would not be acceptable. In summary therefore the above discussion has suggested that intra-object state cardinalities are typically singular (or one-to-one) and that inter-object state cardinalities are often multiple (or one-to-many).

My solution was to group objects or more accurately the states of the same object on the diagram by using dashed lines. Where in the diagram a relationship crosses a line multiplicity would be the default, ie it would be acceptable for one state in one object to have associations with many occurrences of a state in the other object. Figures 12.2 and 12.3 summarise this discussion graphically.

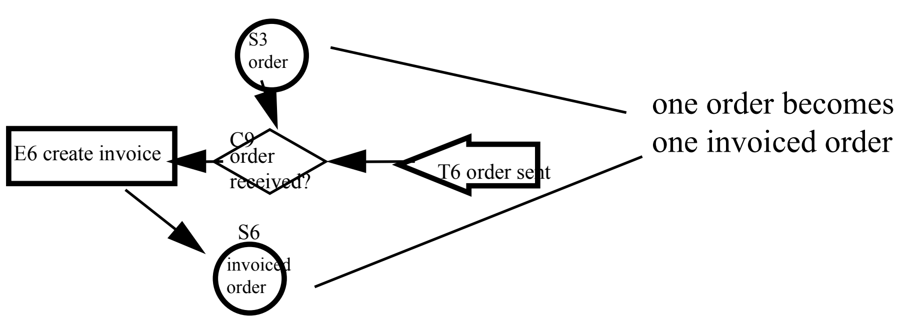


Figure 12.2 Singularity within an Object

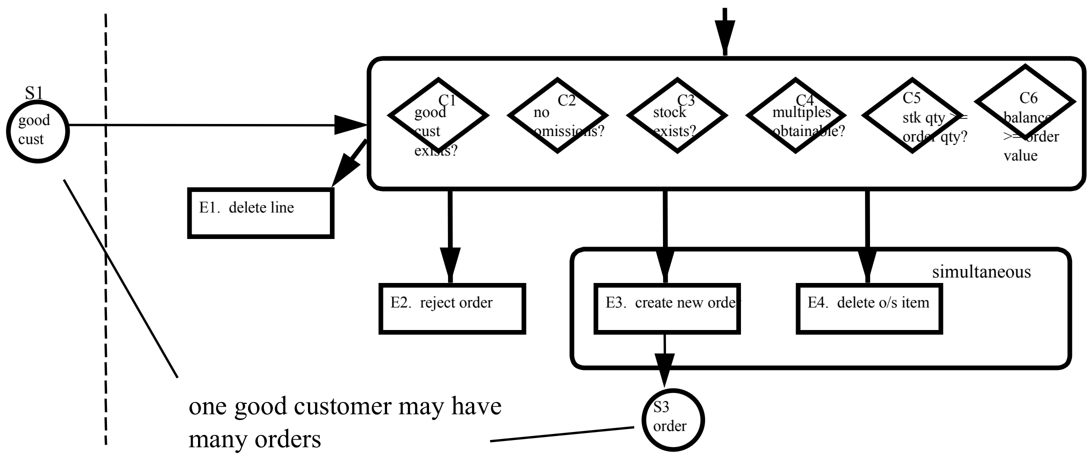


Figure 12.3 Multiplicity between Objects

Of course this does not deal with the situation where the cardinality between two objects is one-to-one. In such situations it would be possible to insert an additional condition which would prohibit more than one instance of a state being generated. See figure 12.4.

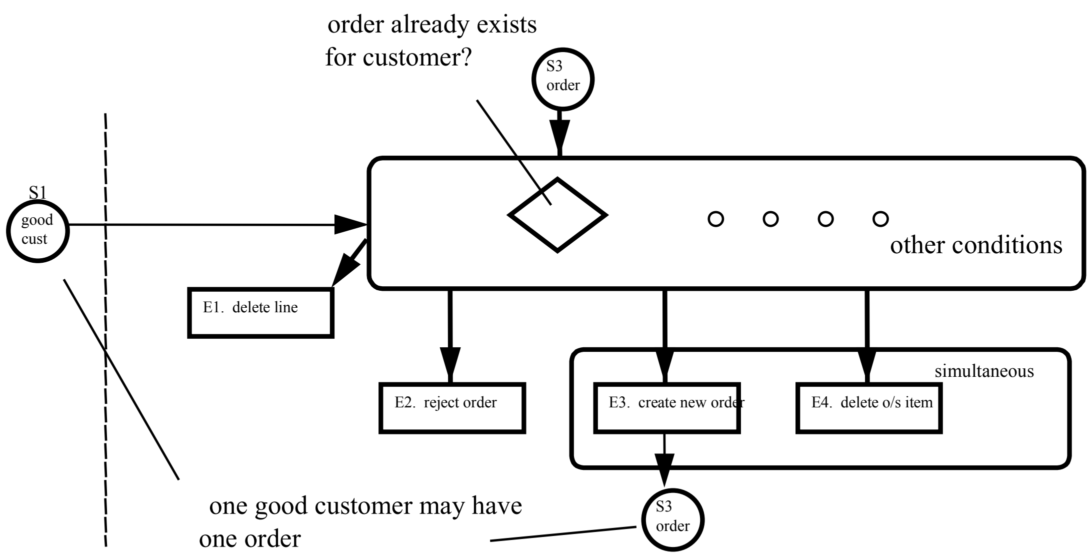


Figure 12.4 Bar on Multiplicity between Objects

As it turned out the 'one to many' problem was more complicated than the above discussion implied. However, at this time it was decided to implement this in the next version of the diagram since the goal of capturing all types of business rule on one diagram was considered very important. Though I was not totally comfortable with the solution, it had sufficient merit to test it out in this study with a view to reviewing it afterwards.

12.4.5 The Abolition of the Simple Event (6P5)

With the decision to develop a user diagram a more relaxed view was taken regarding previous efforts to reduce complexity. Because users would only see one business rule at a time on a diagram it was felt there was now no need to omit event boxes for simple conditions to reduce the number of symbols on a diagram (4A2). So, in future all event boxes would be shown on both user and analyst diagrams.

It was a simple matter to abolish the use of the simple event and redraw the diagram with the appropriate additional symbols. In doing so however, I noticed a certain repetitiveness in the way that triggers, states, events and conditions often repeat themselves. For example for the create invoice event in this particular case study, the pre-state is 'order', the trigger is 'order sent', the condition is 'order received', the event is 'create invoice' and the post-state 'invoice'. There still remained an issue to do with the trade off between expressing a business rule completely and the repetitiveness in the narrative description of its related constructs. Although minor, it was decided to look at this issue later.

12.4.6 Refining the Harel Blob (6P6)

The role of the Harel blob had become more complex in that it was being used as an encapsulator in different situations. Overall, I was concerned that its notation should be as clear as possible. Two straightforward solutions were proposed which helped to resolve this. The first solution was to label each Harel blob with a keyword such as ‘selection’ or ‘simultaneous’. The role and purpose of each Harel blob now became much clearer. The second solution was to omit the Harel blob in simple condition situations. The result of this was a slightly less complex diagram (because of fewer Harel blobs) and one which was clearer for analysts to read.

12.4.7 Changes to the Event Specification Guide (6P7)

The decision to develop separate models for users and analysts meant that there were fewer restrictions on what could be shown in the event specification guide since now only analysts would work with the event specification guide. It was decided that the event specification guide should contain as full a description as possible of the business rule and so triggers, messages and post-states were added. Further the logic of conditions could also be shown, ie whether each condition must be true or false for an event to fire. Thus the role and purpose of the event specification guide now became clearer. Whereas in the diagram only a limited amount of information was abstracted, the event specification guide contained a more complete account of each event. Also at this time the event specification guide was renamed to become the event specification table.

12.4.8 Evaluation of the Preparation Step against Intellectual Framework

Three issues were considered relatively minor in terms of the intellectual framework. Issue 6P1 was to do with categorising events, triggers and messages during the brainstorming step of developing the BRDv3. In terms of the intellectual framework, this was simply an aid to the abstraction process and therefore had no significant ramification for analysis here. Similarly, issue 6P5, abolish the simple event, had little impact at the framework level. Thirdly, issue 6P6, refining the Harel blob to include labels simply clarified the role of the Harel blob and made it easier to work with.

Issue 6P2, the flowchart 'look and feel', was more significant in terms of the framework. Here a flowcharting paradigm had been imposed on the UBRD. It was likely that the number of paths on the UBRD might be quite different to what they would have been if the Harel blob had been used. Figure 12.5 is a copy of a complex UBRD taken from the BRDv3 paper. The application complexity factor, ie the number of independent linear paths is 7. As an aside, note how much smaller this factor is compared to the 79 achieved for the whole diagram earlier.

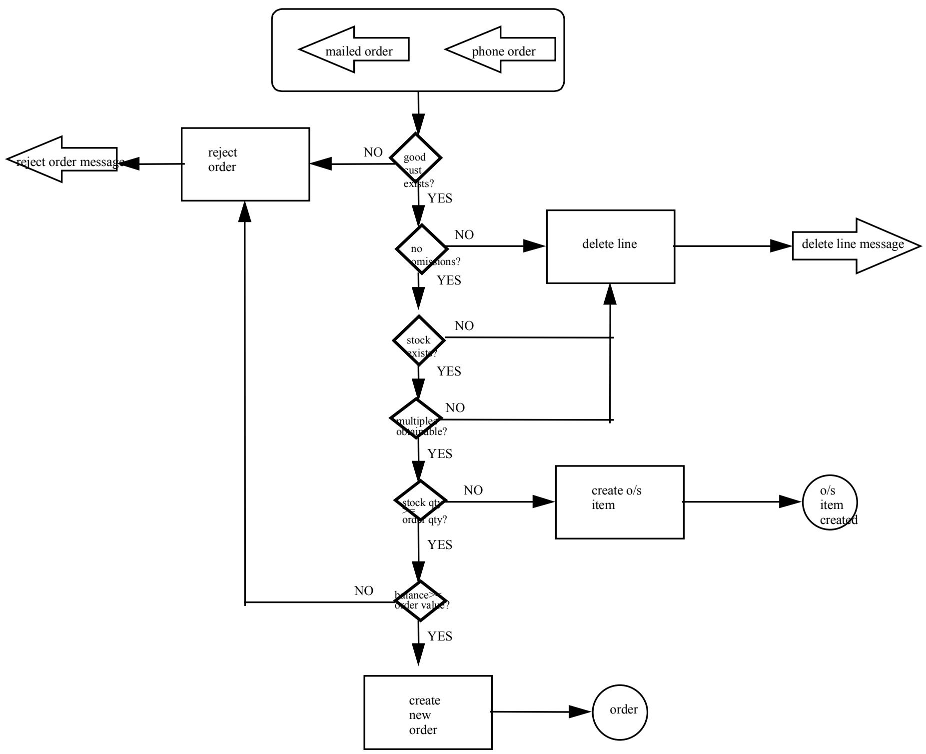


Figure 12.5 UBRD from Case Study

Figure 12.6 is a copy of part of the final BRD from the BRDv3 paper. Only those aspects which deal with the same UBRD have been copied onto this diagram, so it represents what a UBRD would look like if the Harel blob approach had been retained instead of the flowcharting style. Here the application complexity factor is only 4. Some assumptions were made in arriving at these complexity factors eg that pre-states (ie states S1 and S2 on figure 12.6) were not considered in calculating the number of independent paths. However, without the aid of the Event Specification Table, figure 12.3 doesn't show exactly how the outcomes are derived.

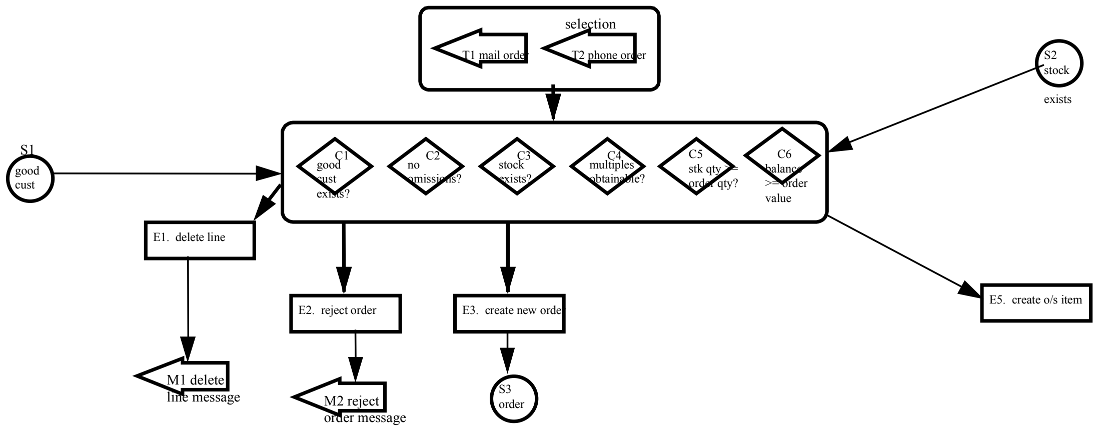


Figure 12.6 Harel Blob Equivalent of UBRD

Table 12.1 show the counts involved in calculating the generic complexity factors for the two diagrams.

Generic Complexity	X	Y	Z	C
‘flowchart’ UBRD	4	2	4	6
‘Harel blob’ UBRD	4	4	7	9

Table 12.1 Generic Complexity Factors for UBRD

The counts for the ‘flowchart’ UBRD were relatively straightforward with four objects (signal, condition, event and state), two relationships (arrow and trigger selection) and four properties (one for the name of each object). For the ‘Harel blob’ UBRD there were four objects also, four relationships (arrow, trigger selection, complex condition and simultaneous event) and seven properties (four for the names of the four objects and three for the names of the three uses of the Harel blob).

In summary, for complex UBRDs the application complexity factor is likely to be greater for ‘flowchart’ UBRDs since the ‘Harel blob’ UBRD tends to hide some of the detail of the logic of decisions. However, the generic complexity factor is higher for the ‘Harel blob’ UBRD since it has a richer notation. Since the ‘flowchart’ UBRD allows the detailed logic of a use case to be expressed on a single diagram and the application complexity factor is considered manageable even for complex use cases, it is concluded that the ‘flowchart’ UBRD is an appropriate device for working with users.

Issue 6P3 concerns the abolition of sub-states. Essentially this is an issue concerned with abstraction and mapping though there may be notational implications. This issue has been discussed in a previous chapter with respect to the intellectual framework. It is hoped that this study might shed some light on this issue.

Issue 6P4 is to do with developing a means to show ‘one to many’ business rules. It has possible implications in terms of abstraction, mapping and notation. Like issue 6P3, 6P4 is being tested in this study. In both cases, the intellectual framework provides little direct guidance to support decisions about these issues.

The last issue concerned changes to the event specification guide/table. In particular triggers, messages and post-states were added. Clearly, this was adding to redundancy since these constructs already appeared on the diagram. On the other hand the role of the event specification guide/table was now becoming clearer. It was now seen as a repository for information which is not to be shown on the diagram but necessary for the complete specification of the system. As such the redundancy was considered acceptable.

12.5 Application Step of Action Research Study 2

Three meetings with analyst and users took place over a period of seven weeks. For the first meeting the analyst had prepared worked examples of states, events, policy rules and so on based on the student enrolment system so that the users could obtain a good grasp of the concepts involved in constructing the UBRD. See appendix 6 which contains copies of minutes of each meeting. In the first meeting, after discussing the goals of the study and providing examples of the constructs, the users were invited to begin the brainstorming activity. This resulted in the creation of a list of candidate objects, an object life history for object student (with some assistance from the analyst), candidate business rules and candidate events and signals. These are documented in the minutes of the first meeting.

The users were generally very complimentary and comfortable about the way the meetings had been handled and what had been achieved as the following comments from the interviews of user A and user B indicate.

> 'Yep. Fairly good and interactive. We listed up all the business rules and then using your method we tried to diagram them which we did over a couple of hours and that seemed to be quite easy to understand and easy to do.' (User A).

> 'They were pretty good, pretty interactive, pretty productive. We had about three separate meetings between myself, analyst Y and user A on different occasions and that was spaced out quite evenly. They were very productive, we got a lot of ideas on paper and then analyst Y would take the knowledge he'd gained from us and he would put together notes and we would review those notes the following time we met and that was a pretty good system. Yeah.' (User B).

Three issues emerged during the application step, ie during the period while the meetings were being undertaken. These are discussed below.

12.5.1 User-led UBRD Construction (6A1)

After the first meeting with analyst and users, the analyst and researcher met. The analyst was of the strong opinion that, given the success achieved in the first session, the users were capable of developing the UBRD on their own with relatively little intervention from the analyst. He suggested that in the next meeting he would allow the users to attempt to draw the UBRDs with as little involvement from himself as possible.

In reviewing the deliverables from the first meeting, the analyst and I concurred that progress had been very good. Given that the meeting had lasted only one hour many candidate constructs had been identified. However, it was observed that there were several inaccuracies in what had been done, for instance some of the objects such as IT equipment and staff member were not likely to be relevant here (see appendix 6). Similarly, some of the candidate business rules such as ‘CBS (Curtin Business School) EO (Enrolments Office) Staff must provide a friendly service’ and ‘no queues’ did not come under the definition of policy rules used here. Nevertheless, it was felt possible that with relatively little guidance the users might begin to see these inaccuracies for themselves and thus correct the specification accordingly. So it was decided to allow the users in the next meeting to attempt to construct the UBRDs with as little interference as possible by the analyst. This was a departure from the original paper in which it was implied that the analyst’s role would involve more supervision and guidance than was now to be attempted. However, it was very much in line with the spirit of the study which was to explore the dynamics of analyst-user interaction. The question of who drives the elicitation process is often referred to in the literature as the locus of control (Kim and Courtney 1988) and more recently Flynn and Davarpanah Jazi (1998) describe an approach to user-led construction of requirements.

The outcome of the second meeting was not so positive in the sense that the directions and interpretations taken by the users were inconsistent with the concepts of business rules diagramming. For instance they identified an enrolment form object and defined states for it including ‘blank’, ‘filled in’, ‘authorised’, ‘processed’ and ‘filed’. The analyst and I agreed that the enrolment form or rather the presentation of the enrolment form was just a trigger to the enrolment system rather than an object. Further, the states identified were irrelevant as states in a business rule. As far as the business rule is concerned, students are either enrolled or not. Some of the states identified by the users would be conditions in the business rule, eg an enrolment form must be completed and authorised before a student can be enrolled. The biggest concern from this meeting was that unguided, the users had spent a lot of time generating OLHs and UBRDs whose states, events, and conditions were not germane to the problem situation. This was echoed in a comment by the analyst in his interview regarding the dilemma he faced during the meetings.

> 'Yes we did. I had a particular challenge with the OLHs myself in terms of clarification because from my own experience I could see where we had specified certain states or certain instances of objects or possibly certain events that should have been triggers or I guess certain errors... We tried to end up with a set of rules with the users and I was more or less torn between telling them up front look this isn't going to work because these things aren't necessary and you'll find they don't do anything once we get through this a bit further and them actually trying to figure that out for themselves. So I ended up with a policy that said that's fine put this in there and we'll look at that later and try to defer that to see if they can discover for themselves that such things weren't necessary. I tried to hold back as long as possible before actually saying anything like look if you actually look at what states are happening and what the events are this thing isn't actually doing anything and therefore it's either outside the model boundary of what we're talking about because it doesn't actually change state or do anything, there isn't any point in modelling it. Once I actually explained that and gave some examples such as we had a staff availability state or in some cases they had staff as being objects which if we're not going to do anything with them there wasn't any point in modelling that and it was arguable whether staff member was an object in any case. So once I gave them a few pointers in that direction many of the states, events and objects fell away to be pruned down to the meat of it so to speak.' (analyst Y).

It was decided that in the third meeting that the analyst would try to guide the meeting in terms of correcting the accuracy of the constructs being modelled as far as was reasonable in the circumstances. In other words the analyst would correct their suggestions rather than start from scratch with his view. Further, it was agreed to limit discussion to one part of the enrolment system, ie the registration sub-system in order to obtain some UBRDs from which a (partial) BRD could be constructed quickly. Two further issues emerged at this point, ie they surfaced in the minutes of the second meeting and continued into the minutes of the third meeting. The first issue was to do with flowchart loops in the UBRDs and the second is concerned with processing rules in the UBRDs.

12.5.2 Flowchart Loops in the UBRDs (6A2)

The examples provided to the users had not involved looping back to earlier states or for that matter conditions or events. It would be possible to loop back to an earlier state, eg for a UBRD which deals with moving customers from good to bad status and back again. But loops should not logically go back to conditions or events! However, in the minutes of the second meeting (appendix 6), loops back to earlier events were drawn by the users in the UBRDs for appointment letters and for enrolments. This incident raised two questions. The first question was whether feedback loops were **necessary** for showing policy rules. The answer is no, at least in terms of the scenarios that occurred here. These instances can be satisfactorily modelled by showing a message leaving the system. The business rule can be re-initiated by another trigger, so there is no need to show looping back at the policy rule level at least for this type of problem. The second question is whether feedback loops are **desirable**. Here we enter into a discussion on preference and style. In my view loops add an element of visual complexity which if not absolutely necessary should be avoided. However this issue will be elaborated later.

12.5.3 Processing Rules in the UBRDs (6A3)

It was also becoming clear from the diagrams in the minutes of the second and third meetings that processing rules were being expressed in these diagrams (appendix 6). A good example of this was in the UBRD for registration in which student number and PIN (personal identification number) were being inserted as conditions in the UBRDs. Moreover, in the same diagram instances of quit and print events are shown.

12.5.4 Evaluation of Application Step against Intellectual Framework

The first issue 6A1, concerned user-led UBRD construction. Glasson’s model allows for each deliverable to be defined in terms of who is responsible for its construction. So, to that extent, the intellectual framework accommodates the issue. However, at a more fundamental level, none of the components of the intellectual framework provided any assistance as to whether it would be better to have users lead the construction or not.

The second issue 6A2, concerned flowchart loops in the UBRDs. Later in this chapter a comparison is performed by examining the application complexity factors for two diagrams (ie one with flowchart loops and one without). However, this issue also raises the question of whether feedback loops were necessary or merely desirable. Again, the intellectual framework provided no help with this.

The last application issue concerned processing rules in the UBRDs (6A3). It was surmised that the users were drawing these diagrams in terms of how they viewed the human activity system. In other words, because they had been working with a computerised information system for a long time the distinction between what was business and what was computer system did not exist, ie the computerised information system was the human activity system at least to them. In terms of the intellectual framework this was an abstraction issue. However, this issue also needs to be considered in terms of Glasson's model, because the original construction path in the BRDv3 paper did not take into account other possible construction paths such as developing a model of processing rules directly and without defining policy rules first.

In terms of the goal of exploring business rules diagramming and analyst-user interaction this was critical. The chief concern was whether a correct and valid BRDv3 could be constructed from UBRDs which contained processing rules. Consequently it was agreed that the analyst would construct two diagrams. The first diagram would be a 'BRD' based on the UBRDs and in it the analyst would incorporate the perspective of the users, ie that processing rules were part of the policy rules. In doing so the analyst would try to capture the spirit of that perspective and so it would contain processing rule information. The second diagram was to be a straightforward BRDv3. Here the question of whether the BRDv3 could be constructed **from UBRDs** containing processing rules would be answered. Both diagrams would be then be evaluated in the review. As far as the intellectual framework was concerned, once again the phenomenon could be mapped using Glasson's model, but the framework itself did not provide guidance on the issue.

12.6 Review of Step of Action Research Study 2

The text from the semi-structured interviews was scanned and themes of interest highlighted in the first and second passes of the transcripts (appendix 8). The issues discussed below emerged from analysing these themes. Although this seems simple and straightforward, in practice this was a time-consuming, challenging and at times frustrating activity. At any rate, eventually the cognitive maps shown below were completed from this analysis. These maps summarise the arguments as well as link back to details in the first and second passes.

12.6.1 The need for guidance (6R1)

Figure 12.7 is a cognitive map summarising the main influences resulting in the conclusion that analyst guidance was necessary in constructing the UBRD. The term ISA in figure 12.7 represents the phrase 'is a kind of' and is used to describe the nature of relationship between bubbles in a hierarchy.

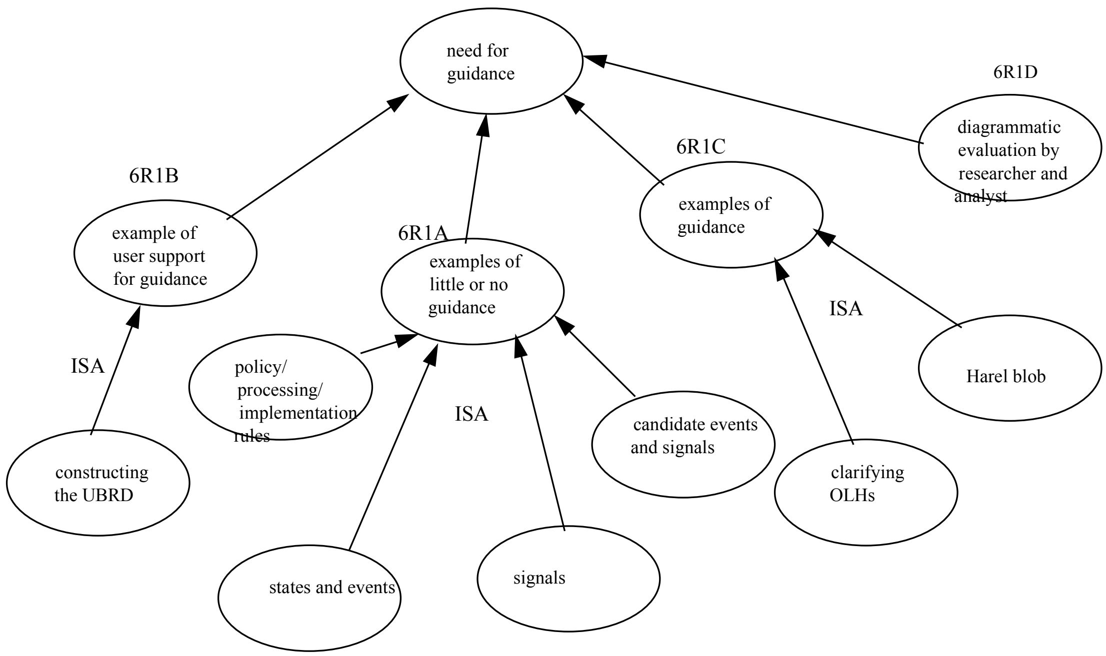


Figure 12.7 Cognitive Map supporting the Need for Guidance

This issue initially surfaced during the application step when it was decided to allow users to lead the construction of the UBRD (6A1). Feedback from the semi-structured interviews together with evaluating the diagrams produced suggest that there was a need for users to be guided in the construction of the UBRD. The support for this from the semi-structured interviews is summarised in table 12.1.

feature being evaluated	pass	participant	characteristic phrase	code
policy/processing/ implementation rules	1	analyst Y	it was difficult for them to determine in which category a rule might fit	6R1A
states	1	analyst Y	I tried to hold back as long as possible before actually saying anything	6R1A
signal	1	analyst Y	my impression in speaking and working with them was that that was an area that they had trouble with at least initially	6R1A
candidate events and signals	1	analyst Y	actually they had some trouble with that when they were generating objects and candidate events	6R1A
constructing the UBRD	1	user B	Analyst Y gave a lot of good input I guess because he's got a lot of first hand knowledge with the whole diagram	6R1B
Harel blob	2	analyst Y	I was surprised and pleased in the way that the users took to using the blob as a concept of aggregating the different elements together	6R1C
clarifying OLHs	2	analyst Y	So once I gave them a few pointers in that direction many of the states, events and objects fell away	6R1C

Table 12.2 Characteristic Excerpts suggesting the Need for Guidance

Table 12.2 has four columns. The first column indicates the answer in the semi-structured interview from which a characteristic excerpt is taken. This is followed by two columns that indicate in which pass the comment was recorded and who the participant was. The fourth column contains characteristic excerpts. A characteristic excerpt is a phrase which typifies the perspective of the whole statement or as Miles and

Huberman (1994, p148) put it '*to get at the essence of the interview material appropriate for each cell*'. Characteristic excerpts should aim to encapsulate why a conclusion was reached by the researcher. The last column contains a cross-reference to the cognitive map.

Table 12.2 contains three types of supporting evidence. The first four rows (down to candidate events and signals) are excerpts which illustrate what happens when there is 'little or no guidance' as far as the analyst is concerned. The next comment from a user implies a positive disposition to being guided. The last two rows are comments from the analyst which indicate that when 'appropriate guidance' is provided that good results follow. These types of supporting evidence are used in figure 12.5. To assist in reading the cognitive map these types of supporting evidence are codified as follows. Examples of little or no guidance is coded as 6R1A, user support for guidance as 6R1B and examples of appropriate guidance as 6R1C.

Further the UBRD produced by the users and documented by the analyst in the minutes of the second meeting was very different to the researcher's and analyst's (6R1D). This resulted in a decision taken in the debriefing meeting afterwards for the analyst to guide users more in the third meeting.

12.6.2 The Need for Validation (6R2)

Figure 12.8 is a cognitive map which summarises the main argumentation which supports the need for validation. As before the ISA relationship should be read as 'is a kind of' with the same meaning.

On three separate occasions during the interview the analyst made the proposal that a walkthrough could be conducted with the users looking at the event specification table as a means of validating the diagram. While more powerful evidence might have been obtained if there had been time to run formal validation sessions such as structured walkthroughs and so on, it is still noteworthy that the analyst's experiences in the construction of the UBRD and BRD led him to the conclusion that

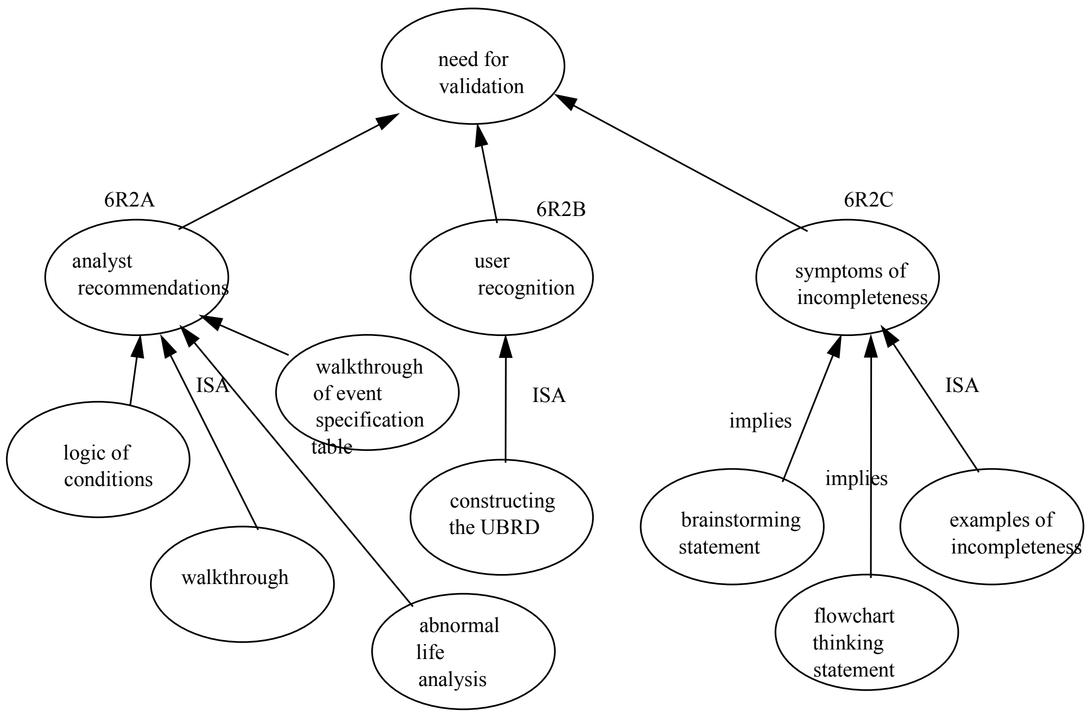


Figure 12.8 Cognitive Map supporting the Need for Validation

validation sessions would be valuable in terms of improving the completeness and correctness of the diagram. Table 12.3 summarises the evidence from the semi-structured interviews which support the argument for the need for validation. The format of table 12.3 is similar to that of table 12.2.

Table 12.3 contains three types of evidence supporting the case for the need for validation. The first four entries represent the analyst's recommendations (6R2A). The fifth entry is a recognition from a user that the UBRDs are incomplete (6R2B), which also supports the need to validate these diagrams in some way. The last two entries are statements from elsewhere in the interviews which imply there may be a need for validation. In other words they could be categorised as symptoms of incompleteness in the models produced (6R2C). Further, table 12.4 contains a number of specific statements from the interview in which an omission or oversight is specifically exemplified. Such examples of incompleteness reinforce the argument for the need for validation.

feature being evaluated	pass	partic-ipant	characteristic phrase	code
logic of conditions	2	analyst Y	as to whether those UBRDs that were deconstructed were still valid based on that experience of the business rule	6R2A
walkthrough	2	analyst Y	That would have been a very valuable experience because that may have actually picked up certain events or parts of the model that aren't expressed in the diagram currently	6R2A
abnormal life analysis	2	analyst Y	they didn't tend to explore events or states that were modelling abnormal life analysis as you've called it	6R2A
walkthrough of event specification table	2	analyst Y	that would converge the event-spec table back to business rules and try and validate that those rules were the ones we had originally	6R2A
constructing the UBRD	1	user A	I mean it's not 100% but certainly at least the major components of the process	6R2B
brainstorming	1	user B	Analyst Y couldn't keep up with us	6R2C
flowchart thinking	1	user B	we found that it was a bit more complicated to diagram things like that	6R2C

Table 12.3 Characteristic Excerpts suggesting the Need for Validation

12.6.3 The level of abstraction problem (6R3)

The term ‘level of abstraction problem’ is used to refer to the problem of whether only policy issues should be captured on the diagram. Experience in developing early versions of the diagram at the Port Authority suggested that processing and implementation rules could be omitted in a diagram to be shown to users thus reducing the complexity of such diagrams. For example, there was no need to show copies of transactions made for audit purposes when trying to discover the ‘basic’ rules of a human activity system. However, the student enrolment system highlighted a different scenario altogether. Here users had been using a computerised system for so long that they tended not to distinguish between that which is policy and that which has been implemented in order for the information system to function. Information systems rules may be either processing or even implementation rules.

feature	**pass**	**participant**	**specific instance**	**code**
clarifying OLHs	2	analyst Y	a staff availability state or in some cases they had staff as being objects	6R2C
walkthrough	2	analyst Y	what it doesn’t model is them moving from class to class	6R2C
walkthrough	2	analyst Y	nor does it model a student getting out of a class	6R2C
abnormal life analysis	2	analyst Y	what was going to happen if a student wasn’t validated?	6R2C
abnormal life analysis	2	analyst Y	If they didn’t have a valid appointment, what actually happened to them?	6R2C

Table 12.4 Specific Statements from Interview in which Omissions are identified

Figure 12.9 is a state-event network for the level of abstraction problem. A state-event network is simply a modified cognitive map. Miles and Huberman (1994) suggest that the use of the state-event network is especially useful for assessing the causal dynamics within a particular case. Its purpose here is to allow for two types of node to be described. Bubbles remain as before, ie a fact, belief or concept considered important in a logical argument. A rectangle in the diagram is an event. The event of interest here was the test involving two diagrams. The state-event network was chosen for its ability to show inter-relationships between events and state. Once again the ability to add additional steps into the action research process such as the additional task (event) of drawing two diagrams, illustrates the flexibility of action research as a tool.

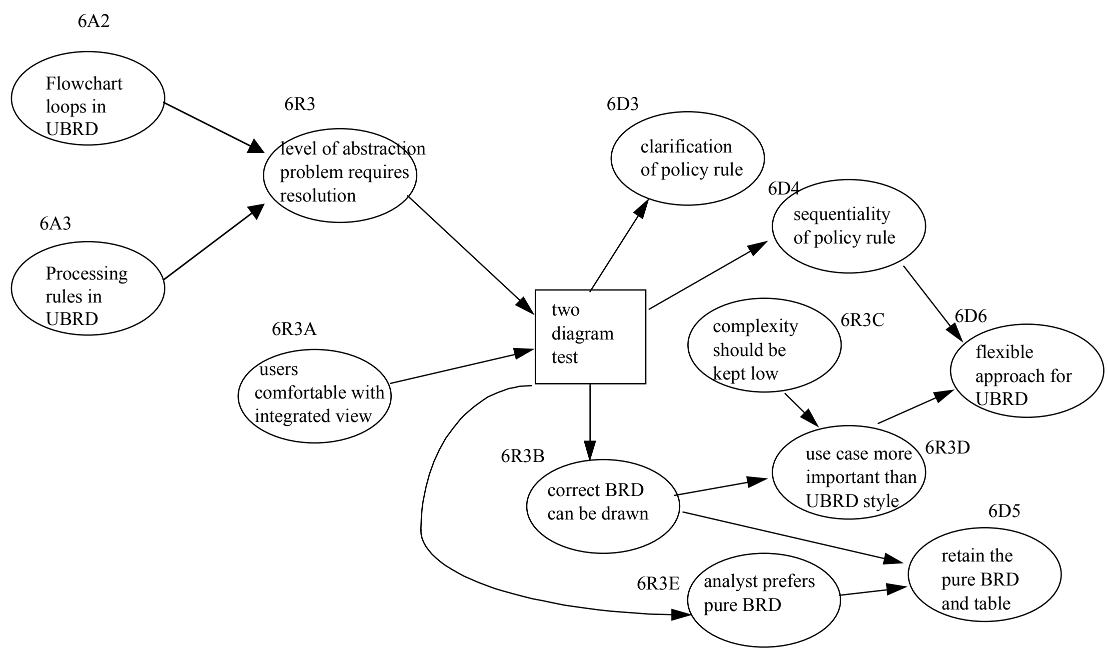


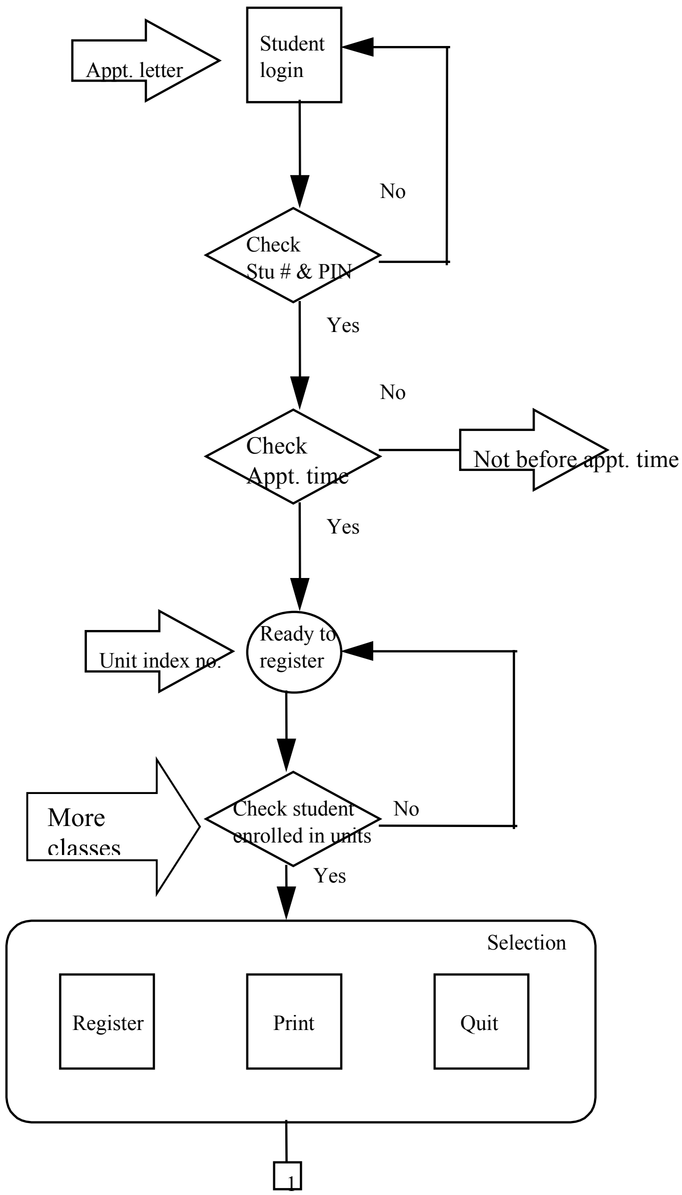
Figure 12.9 State-event Network for Levels of Abstraction Problem

The event in figure 12.9 is a consequence of a decision taken during a debriefing session between the analyst and researcher and is discussed later. Table 12.5 contains excerpts which will be used in the following paragraphs to provide a commentary.

Issues 6A2 (flowchart loops in UBRD) and 6A3 (processing rules in UBRD) resulted in a recognition that the level of abstraction problem required resolution (6R3). From entries 1 and 2 in table 12.5 a strong conclusion was that users were generally comfortable with an integrated view (6R3A), ie a diagram which depicts policy and processing rules. This view emerged during the first and second meetings with the analyst. It was decided that the analyst would attempt to draw two diagrams. The first diagram was to be a diagram which, in the analyst's view, reflected the way that the users perceived the business rules of a sub-system of the student enrolment system. This diagram will be referred to as the alternative diagram. See figure 12.10.

feature	pass	participant	characteristic excerpt	code
computer vs business system	1	user B	Without the computer you can't really have the enrolment system	6R3A
computer vs business system	2	analyst Y	so I think that they didn't tend to distinguish between the information system that they used and the overall conceptual business rules	6R3A
trade-off	2	analyst Y	it actually breaks up the conditions into multiple blobs	6D4
trade-off	2	analyst Y	My personal preference is for the second one because from a conceptual point of view it's cleaner	6R3E

Table 12.5 Feedback from Interviews on Level of Abstraction Problem



Appt. letter
Student login
No
Check Stu # & PIN
Yes
No
Check Appt. time
Not before appt. time
Yes
Unit index no.
Ready to register
More classes
Check student enrolled in units
No
Yes
Selection
Register
Print
Quit
1

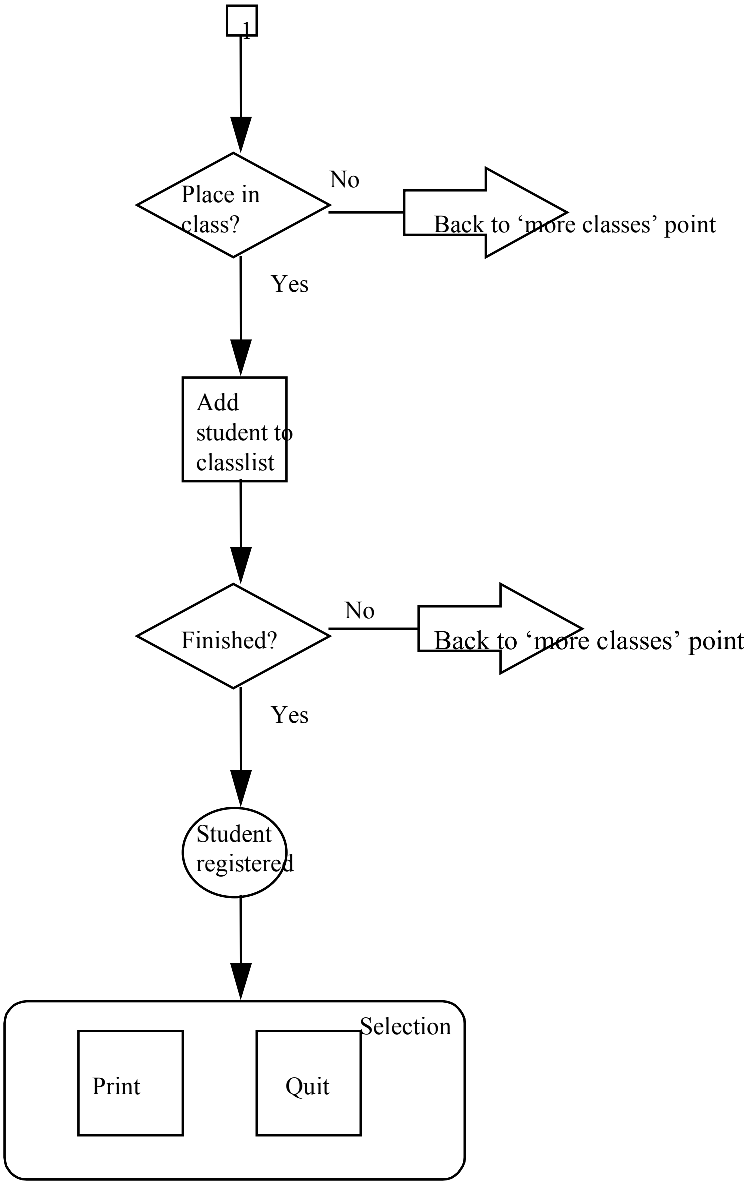
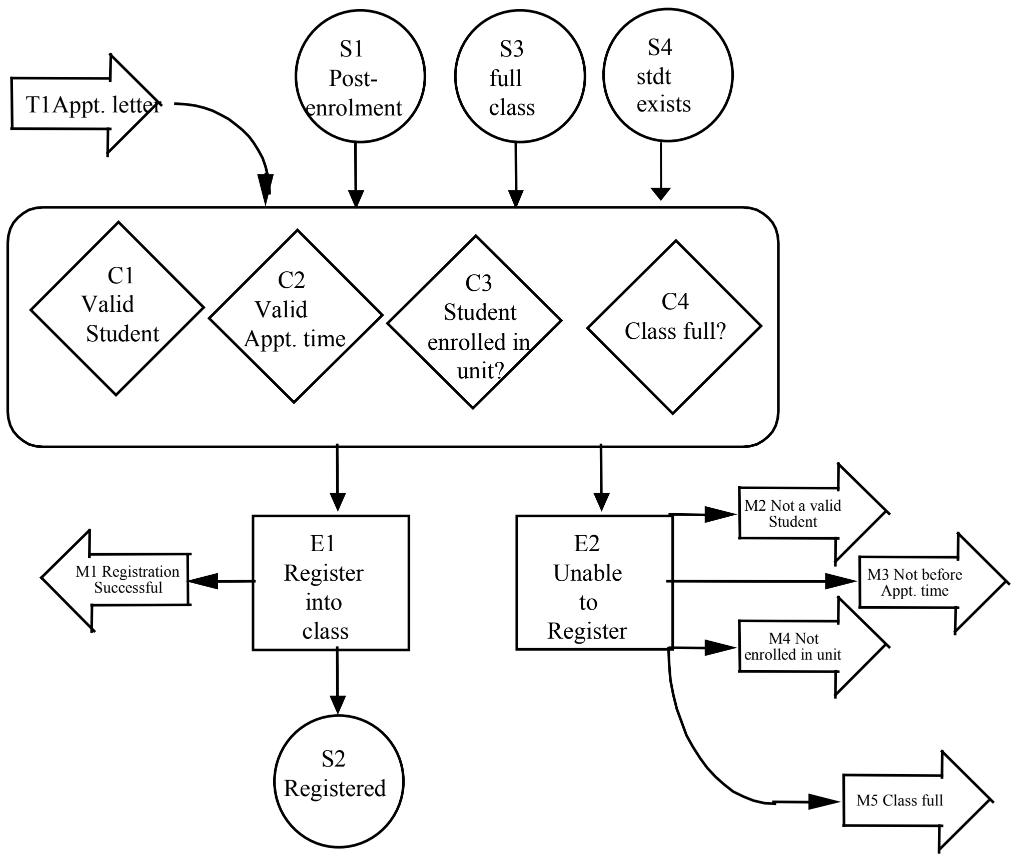


Figure 12.10   Alternative BRD for the Registration Sub-system

It should be noted that the analyst did not consider that this diagram was complete or correct, but rather it was the analyst's view of what the users perceived as reflecting the business rules. For example there is no state which indicates that the student is registered into a class. The second diagram was a straightforward BRD. This diagram will be referred to as the pure diagram. See figure 12.11.



Here, it is assumed that there is a student object which has been identified separately from an enrolment/registration object. Clearly, if the enrolment/registration had been subsumed within a complex student object, a different diagram would have resulted.

Figure 12.11 Pure BRD for the Registration Sub-system

Analysis of the two diagrams drawn produced a number of conclusions. These concerned the clarification of policy rules (6D3), business rule sequentiality (6D4), the retention of the pure BRD and table (6D5) and adopting a flexible approach with the UBRD (6D6).

The need to clarify policy rules (6D3) is exemplified in the following argument. Consider the situation where a student presents to the student enrolment system in order to register into classes. The computer checks that student number, PIN and appointment time are all valid. The policy rules associated with this part of the registration system are that 'student must exist', ‘student must be enrolled in that unit’, that ‘registration can take place only after a designated appointment time' and that 'classes must not be full'. In order for these rules to be checked it is necessary for the information system to confirm the identity of the student. This is implemented by the student entering student number and PIN (Personal Identity Number). Once the unit code is entered the system will check if the appointment time is valid, if the student is enrolled in that unit and if the class is full. It is assumed that the student object stores the student number and PIN and further that the enrolment/registration object stores the appointment time.

The above example highlights an important distinction between a policy rule and a processing rule. There are only four policy rules and they may be expressed at a high level of abstraction. For example the statement that students must be enrolled in a unit does not indicate how a valid student will be confirmed eg it does not indicate whether a student will be given one or more opportunities to enter a student number and PIN. On the other hand, when defining processing rules, it would be necessary to define the number of attempts available to enter student number and PIN. So it can be seen that processing rules may be elaborations of existing policy rules as well as be new rules such as copying transactions for audit purposes.

I consider it vital that an organisation has the opportunity to decide at what level to define its business rules. In this example, the organisation by omission is stating what is not policy. In this case, the rules for how a student's identity is validated is **not** a matter of policy. Whether it should or should not be is another matter. The point is that it is important to allow a way of inferring there is no business policy about certain aspects of a state change (business rule). In such cases the processing rule would have to pick this up.

It was concluded that the dichotomy here was not so much between business and computer system which was the label used earlier to describe this phenomenon but between policy rule and processing rule. This is why this section has been entitled the level of abstraction problem. The clarification with regard to a policy rule is documented as 6D3 on the state-event network and the research progress map.

Another conclusion which emerged from the drawing of the two diagrams concerned business rule sequentiality (6D4). In one sense this is another clarification of a policy rule. Consider entry 3 in table 12.4 where the analyst claims that one advantage of the alternative UBRD is that it maps out the sequence of steps in a use case and therefore users are likely to find this intuitive since it reflects the sequence of events they are used to in the existing computerised system. This observation by the analyst raises the question of ubiquity of sequence as far as business rules are concerned. It may be argued that some business rules require sequence, ie that it is a policy rule that condition A is tested before condition B. For example in the case study used throughout this thesis, credit availability must be checked before stock availability. On the other hand, it may be the case that a policy rule does not require sequence. In other words, at the level of business policy it may be decided that such matters are best left open-ended. Analyst Y's solution of using the Harel blob is therefore a way of visually accommodating situations where sequence is considered irrelevant at the policy level. However, the notation in the event specification table would have to be extended since currently the comma between conditions implies sequence.

The next two conclusions use an intermediate conclusion which is that the analyst was able to produce a correct BRD from UBRDs which contained both policy and processing rules (6R3B), ie figure 12.11. There is only one minor variation in this diagram compared to the way I would have drawn it. That variation is in the way that many different messages emanate from a single event. I would have separated these into different events. However, this is just a different way of looking at an event since this particular event does not involve a state change. The important point is that exactly the same information concerning states, events, conditions etc could be extracted from either the diagram I would have drawn or the one the analyst drew. So in that sense they are equivalent.

The third conclusion is to do with analyst preference and this emerged from the analyst evaluating the two diagrams he had drawn.. As far as the analyst was concerned he stated he preferred the ‘pure’ BRD, ie the diagram following a strict interpretation of the rules for drawing the BRD as evidenced in entry 4, table 12.4 (6R3E). Coupled with the conclusion that it was possible to draw a correct BRD from variants of user diagrams (6R3B) this led to a decision to retain the original version of the BRD and its event specification table for use with systems analysts (6D5).

The last conclusion concerns adopting a flexible approach with the UBRD. In previous chapters the issue of complexity and the implication that the complexity of any diagram to be shown to users should be kept as low as possible (6R3C) has been widely discussed. This was a position which was felt to have a strong bearing here. However, another position which was felt to be equally important here was that a correct BRD could be drawn from variants of user diagrams (6R3B). From these two it was concluded that it was more important to limit the complexity of a user diagram by working with use cases than it was to be concerned with the exact style of diagram adopted with users (6R3D). This conclusion coupled with the conclusion on business rule sequentiality (6D4) in turn led to a position that a flexible approach in terms of style of UBRD diagram should be adopted by the analyst when drawing the UBRD (6D6).

12.6.4 Evaluate Review Step against Intellectual Framework

Both issues 6R1 (the need for guidance) and 6R2 (the need for validation) emerged from the data analysis. Therefore expressed in terms of Checkland’s intellectual framework, applying methodology (M) to application area (A) and subsequently reviewing it had generated these conclusions. Though the framework of ideas (F) had no sense of guidance or validation criteria, it was still possible for such ideas to emerge. This point is discussed further in the next chapter where the intellectual framework is critiqued.

The third issue was the level of abstraction problem (6R3). It involved the analyst drawing two diagrams. One was a ‘pure’ BRD which followed the rules laid down in the BRDv3 paper. The other was an ‘alternative’ BRD which tried to capture the way it was perceived that users viewed the problem situation. In terms of showing alternative routes to developing the BRD, whether the pure BRD route was taken or whether the alternative route was taken first and then converted into a pure BRD could be handled satisfactorily by Glasson's model. One important conclusion that emerged from this issue was that a pure BRD could be drawn from an alternative BRD.

In terms of the application complexity factor, the pure BRD (figure 12.11) had a factor of 5. The calculation for this used broadly the same assumptions as before. However, note that the path through event E1 which generated message M1 and state S2 was considered a single linear path (as both outcomes always happen as part of event E1). The complexity factor for the alternative BRD (figure 12.10) was 19, though the logic at times was sketchy and there were clearly some omissions on the diagram. Notwithstanding the problems with the drawing of this diagram, as predicted the pure BRD was less complex than the alternative since the pure BRD did not contain processing rules for example relating to quit and print. Thus, comparison of the two diagrams in terms of the application complexity factor also suggests that the pure BRD should be retained.

A number of conclusions emerged from this issue. These were that

- a correct BRD can be drawn from an alternative BRD (6R3B),
- the analyst prefers the pure BRD (6R3E),
- focusing on the use case is more important than UBRD style (6R3D),
- processing rules may be elaborations of policy rules rather than different rules (6D3),
- sequentiality in a rule may be required (6D4),
- a flexible approach to using the UBRD is required (6D6) and
- the pure BRD and table should be retained (6D5).

Of these 6R3B, 6R3E, 6R3D, 6D6 and 6D5 are conclusions which are drawn from applying the methodology to an application. As such they may be considered knowledge or insight obtained from this research though the extent to which they can be said to be generalisable to other applications is debatable. 6D3 relates to an abstraction issue and later how a mapping from policy to processing rules might be obtained. 6D4 has abstraction, mapping and notational ramifications.

12.6.5 Discussion in terms of Wider Literature

The use of the UBRD relates to the representation model (table 7.3) in the following way. The UBRD is essentially a device for reducing the complexity of the notation to a single instance of what Wand and Weber (1993) refer to as a transformation. This is also linked with the concept of systems and subsystems in the representation model. Thus the use of the constructs of state, event, condition and the decomposition of BRD into UBRDs all reinforce the legitmacy of the value of the representation model.

The concepts of trigger and message however, in my view are not catered for in the representation model and therefore further work is required to establish how these constructs might integrate into a richer representation model. Clearly, as the literature review confirmed, the constructs of trigger and message occur in some models and either this is an omission in the representation model or there is an alternative mechanism for providing the same information. For example, the idea of an external event has some similarity with the concept of a trigger and so could be an alternative, but in the BRD it is not a type of event but a different 'thing'.

12.7 Summary

The main goal of this study was to extend the research to consider more closely how users and analysts could work with the BRD. In terms of research progress three types of outcome occurred. The first type of outcome was that some of the problems or issues which existed at the beginning of the study were resolved. The second type of outcome was that some outstanding issue which emerged during the study remains unresolved at the end of the study. This may be due to a limitation of the study, for example perhaps the study did not contain aspects which allowed a meaningful examination of that problem to take place. Thirdly, during the course of the study new issues emerged.

There were two issues in the first category, ie issues outstanding at the beginning of the study which were resolved during the study. The first of these was issue 6P1 which was concerned with categorising events, triggers and messages. The minutes of the meeting confirm that this addition was incorporated into the construction process and in my evaluation this was done accurately. No adverse comment was received from any participant regarding problems with categorisation. This is a relatively minor amendment and it is concluded that this addition was successful.

The second issue was to do with adding the message construct as a component of the business rule (5D1). Previously it has been argued from a theoretical perspective that a message was a valid component of a policy rule and examples were provided to support this claim. From the semi-structured interviews it will be recalled that the users considered this an aspect that was '*easy to follow*' and '*made perfect sense*'. Indeed one of the users commented that the signal *was 'one of the symbols that I [she] remember[ed] the most'*. However, the analyst did feel that they had some problems with the concepts initially although this was later assessed as confusion over identifying them rather than understanding them. It is concluded that the message construct should be incorporated as a component of a business rule and given that it is proposed that some degree of analyst guidance is to be provided, no major difficulty is seen in identifying or using messages.

The second category of outcome was where some aspect of an issue still remains unresolved. Five issues fall into this category. The first issue was concerned with changes to the Harel blob (6P6). The change to labelling each blob with a key word was considered satisfactory. There was no difficulty in rewriting the paper. Further, the analyst reported he had '*no trouble with the blob in using it, certainly not in the instances ... used ..., in terms of selection or parallel states or complex conditions or* *simultaneous events'*. However, there is still the issue that as currently specified, it is not possible to show exceptions in a selection construct. So, this issue is ongoing.

The second and third issues in this category are the abolition of sub-states (6P3) and the one to many business rule (6P4). Both these issues were not resolved in this study. The part of the particular business application used did not shed any light one way or the other. Both these issues are referred for discussion in the next chapter.

The fourth issue is to do with changes to the event specification table (6P7). The changes were to add trigger, message and post-state columns to the table. Further, the logic of conditions was also to be shown. The analyst felt that the table could be shown to users as a means of validating the diagram. The changes to the event specification table were considered appropriate. Although arguably redundant since some of this information is repeated graphically in the diagram, it is my view that the entries form a more complete statement about each event which can be argued to be beneficial. However, these entries can be argued to be incomplete. Further, there is also the issue of extending the notation of the event specification table as a result of the business rules sequentiality issue (6D4). Consequently, this matter is considered ongoing.

The fifth and last issue in this category is the question of the abolition of the simple event (6P5). In the BRDv3 every event explicitly shows all connected conditions, triggers, messages and pre- and post-states regardless of simplicity. Though this is a minor amendment and in itself caused no difficulty, the observation that much of a business rule is specified repetitively still remains a concern and is therefore categorised as ongoing.

The last category of outcome was new issues that emerged during the study. Three major issues emerged. During the application of the new diagram an issue concerning user-led UBRD construction (6A1) led to the conclusion that there was a need for analyst guidance in constructing UBRDs (6R1). A second set of issues surrounded the level of abstraction problem (6R3). Issues during the application step which were related to this were flowchart loops in the UBRDs (6A2) and processing rules in the UBRDs (6A3). Analysis of this issue resulted in four decisions being taken. These are that policy rules need to be clarified (6D3), that non-sequentiality of business rules is permitted (6D4), that the pure BRD and table should be retained (6D5) and that a flexible approach for the UBRD can be taken (6D6). The last issue in this category is to do with the need for validating the diagram (6R2). The analyst's experiences and well as comments from the interviews supported the need to validate the diagram to improve completeness and correctness (6D2).

This study had a strong social focus. The study examined how the BRD might be used by users and a systems analyst on a business application and it sought to obtain feedback on their experiences and preferred work practices. Though much of the study gathered data which can be regarded as social, this is not to say that technical considerations or issues were ignored. Such technical matters remained underpinning the discussion at the social level. For instance, the level of abstraction problem was both a technical and social problem. Further, the analyst was able to provide direct feedback on technical aspects. So although there was a focus on social issues, the study was also a further iteration of the whole technique. This further iteration was undertaken by a different analyst on a different business application in a different organisation.

The study was successful in the sense that feedback was obtained which permitted broad conclusions to be made about the value of the diagram. For example, the users and systems analyst were overall quite positive about the use of the technique and its ability to assist in modelling business rules, some existing problems were resolved and others were able to be progressed during the study. However, no progress was made on other issues such as the abolition of sub-states. Also, I had hoped to obtain feedback on how my ideas on validation might be accepted. Unfortunately, because of the time spent on exploring the issue of user-led construction of the UBRD it was not possible to run sessions on abnormal life analysis and amendment analysis.

To summarise, two issues were considered resolved. These were:

- categorising events, triggers and messages (6P2) and
- adding the message construct to the business rule (5D1).

At least some aspect of five existing issues was considered ongoing. These are:

- implementing changes to Harel blob notation (6P6),
- the abolition of sub-states (6P3),
- the one to many business rule (6P4),
- the event specification table (6P7) and
- the abolition of the simple event (6P5).

Some very important issues emerged during this study regarding the relationship between users and analyst and their respective roles. Two of these concerned:

- the need for guidance (6R1) and
- the need for validation (6R2).

A third issue emerged which was referred to as the level of abstraction problem (6R3). This resulted in four decisions being made. These were:

- to clarify the definition of a policy rule (6D3),
- to permit non-sequentiality in a business rule (6D4),
- to retain the pure BRD and table (6D5) and
- to maintain a flexible approach to the UBRD (6D6).

The next chapter analyses the findings of this research as a whole.

Chapter 13

Analysis of Findings

13.1 Overview

This chapter has one purpose and two parts. Its purpose is to inform and explain what has been learnt in this research. There are two parts to this discussion. In the first part, a discussion is provided at the level of what has been learnt about business rules and the BRD (Business Rules Diagram). Since, there is a great deal of detail to be reported, this necessarily takes up a large portion of this chapter. In the second part discussion focuses on what has been learnt about how to use action research to design new diagrams. In particular, there is a critique of the use of the intellectual framework.

The first part of the chapter has six sections. In the first of these a model is introduced as a means of partitioning the discussion. The next section looks at what has been uncovered about business rules at a high degree of abstraction, ie it reviews the definition of a business rule proposed in chapter 2 and discusses the justification for the main components of a business rule. This is followed by a discussion on policy rules. Policy rules are a unique characteristic of this research. Arguably they represent a severe limitation of the definition of a business rule. So it was considered important to discuss this in full. The next three sections describe what the research has uncovered concerning the three main deliverables of the technique, namely the UBRD (User Business Rules Diagram), BRD (Business Rules Diagram) and EST (Event Specification Table).

The second part of this chapter examines what has been learnt about action research and in particular how action research might be used to assist developers of new diagrams. Much has been stated earlier in this thesis about how an intellectual framework should be used to guide and frame the research. In this chapter a final evaluation of the role and place of an intellectual framework is conducted and as such this represents the final step of a learning framework described in chapter 6.

13.2 Introduction to what has been learnt about Business Rules

In chapter 3, the research question was introduced as an endeavour to develop a technique which can model business rules in organisations. There are many implications to such a statement including questions such as what is meant by a business rule, who will be involved, in what way, what training is required and so on. Given the nature of the research question it was arguably very difficult to conceive of all such implications in advance of the undertaking of the research and this was also discussed in chapter 3. Business rules are discussed at two levels. Firstly, section 13.3 examines what has been uncovered at a high level of abstraction. Colloquially, this might be termed 'the big picture'. In chapter 2, a definition of a business rule was proposed. This definition was a synthesis of observations and conclusions about the nature of a business rule which were constructed over the course of this research and as such represents some important aspects of what has been uncovered. Also at this level of discussion, it is appropriate to review the main components or constructs of a business rule and the evidence for their inclusion as constructs. This is because these constructs are fundamental in defining what a business rule is.

Secondly, the research is discussed at a lower level of abstraction. The research has led to a number of detailed issues and conclusions which follow if one accepts the general goals and directions outlined at that higher degree of abstraction. For instance, at a high degree of abstraction I have asserted that modelling events is important. At a lower level of abstraction questions to do with the simultaneity of events have had to be addressed. Of course a discussion at this lower level has to be interpreted in the context of the discussion at the higher level of abstraction and so such discussion serves to both support and elaborate earlier discussion.

Figure 13.1 is a model of how the research is viewed at the lower level of abstraction. The figure suggests there are a number of levels of abstraction of a business rule. Of direct relevance to this research is the level associated with policy rules and by implication the delineation between policy rules and processing rules. Section 13.4 reviews the findings of this research with regard to policy rules. Further, there are three main deliverables in the technique. These are the UBRD, the BRD itself and the EST. These deliverables are relatively independent of each other in the sense that for example some aspect of the UBRD may be changed without it necessarily affecting the BRD or the EST. Of course other aspects are not so independent. For instance figure 13.1 implies that UBRDs are constructed first of all and then combined to form the BRD and further that the BRD is used to define the EST. However, for the purpose of review here sections 13.5, 13.6 and 13.7 consider research progress in each of these main deliverables respectively. Additionally there is the problem that some conclusions apply to more than one section. For example, the conclusion about the notation for alternative point of entry applies equally to the UBRD and BRD. Where this is so, this has been discussed in section 13.6 in the BRD section.

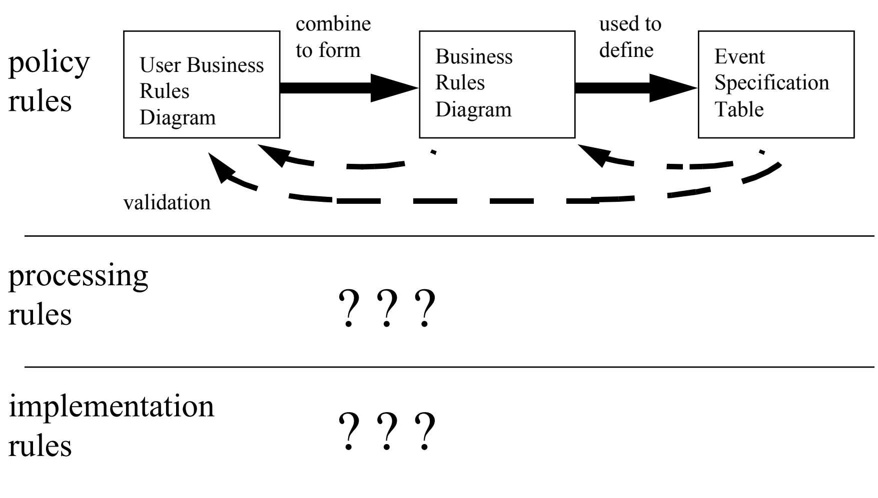


Figure 13.1 Model of Main Components of the BRD Technique

In each of these sections the same general approach to justifying the conclusions is taken. In a paper by Whetton (1989) entitled 'what constitutes a theoretical contribution' he identifies four main building blocks of theory development. He terms these the 'what', 'how', 'why' and 'who, where and when'. The 'what' represents a statement of what the research has uncovered. This is done typically in the form of a conclusion. The 'how' is a statement of how that conclusion has been arrived at. In order to relate this discussion to previous chapters, the research progress codes in the research progress maps are used to summarise the history of progress with regard to that topic. This permits cross-referencing to the place in the research where the topic was discussed in full. The 'why' is a statement explaining the rationale for the conclusion. This section therefore attempts to articulate an explanation for each conclusion. The fourth building block is the 'who, where and when'. In essence this section will try to delimit the conclusions reached by providing any limitation or qualification that applies to that conclusion. Of course the whole of this research is qualified by the generally accepted limitations of action research which have been detailed in chapter 5 on research methodology. In addition in chapter 6 in describing research design the characteristics of this research project were outlined. Of particular relevance here is the limited number of studies undertaken, the number and type of the business applications examined, the relatively few individual participants and so on. In this chapter only where a particular limitation is identified which is over and above these more general constraints, will it be discussed here specifically. Although the next chapter deals with future research, occasionally there is an obvious deduction for future work which can be directly made from discussion in this chapter. Where this is the case, an additional section has been added briefly summarising such work.

13.3 A Review of the Business Rule.

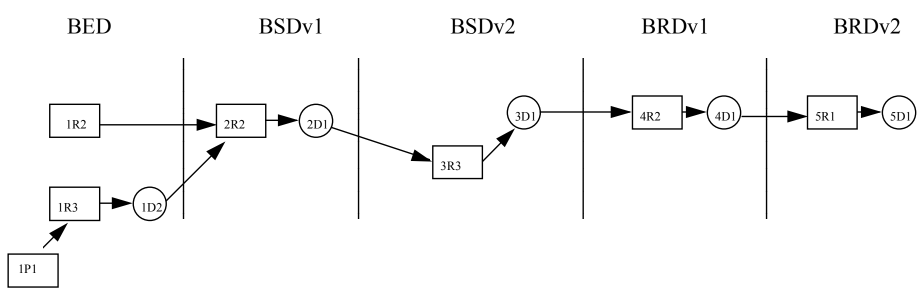
Before reviewing the business rule in detail I should like to defend the choice of term 'business rule'. I have considered at length whether it would have been better to have renamed the diagram and technique the Policy Rules Diagram because other kinds of 'business' rule such as processing rules and implementation rules have been specifically excluded. It would perhaps have made for less ambiguity if I was then able to use the term business rule as the all-encompassing term of which policy rules, processing rules and implementation rules are components or aspects. On the other hand there is also a possibility of confusion or ambiguity over the term policy rule. One possible interpretation of a policy rule is that it relates only to rules prescribed at the very highest level of an organisation. For example perhaps only directors or vice-presidents would be permitted to define policy rules. Of course that interpretation of the word policy is not how it has been used in this thesis. A policy rule here captures characteristics of a human activity system regardless of who prescribes them or at what level in the organisation they relate. Many policy rules might be considered mundane, routine and operational. The important point however, is that such rules represent the core rules of a human activity system. On balance I have decided to retain the term business rule as the descriptor of the diagram because I feel it more closely communicates the nature of what is being modelled. Just like any other technique, there has to be definitions of terms associated with the technique and so the qualification that business rules are limited to what I have defined as policy rules would necessarily become clear when working with the technique.

What

In chapter 2, a business rule was defined as follows. '*A business rule is an explicit state change context in an organisation which describes the states, conditions and signals associated with events that either change the state of a human activity system so that subsequently it will respond differently to external stimuli or reinforce the constraints which govern a human activity system.*'

How

The flow of argumentation that led to the conclusion that the state, event, condition and signal constructs are necessary in defining a business rule is now traced. See figure 13.2. The first diagram was the BED. This diagram had the event as the only modelling construct and therefore centre of modelling focus (1P1). However, an issue concerning the ability of only events to explain the full circumstances of transitions between events was identified. I coined the term rule completeness to describe this problem (1R2). The BSDv1 in contrast used the state as the single modelling construct in the model and therefore the centre of modelling focus. The decision to attempt this (1D2) had arisen out of a discussion of the merits of states versus events (1R3). However, after constructing the BSDv1 the rule completeness issue still remained a problem (2R2) so it was decided to extend the prerequisite table in an attempt to include more information about a rule, ie pre- and post- conditions were added (2D1). During the review of the BSDv2 it was concluded that focussing on a single modelling unit was insufficient to meet the rule completeness issue (3R3) and so the BRDv1 employed the tactic of modelling both states and events (3D1). The rule completeness problem still remained as an issue however (4R2) and so the in the BRDv2 conditions and triggers were added so that in total four constructs were being employed on the diagram (4D1). An argument was made when reviewing the BRDv2 that demonstrated the need to add the message construct, ie an output signal for completeness (5R1) resulting in four main types of construct being used to depict a business rule (5D1), ie states, events, conditions and signals.



Legend:

1P1	event focus
1R2	rule completeness
1R3	state vs event
1D2	change to state
2R2	rule completeness
2D1	extend prerequisite table
3R3	rule completeness
3D1	states and events
4R2	rule completeness
4D1	show conditions and triggers on diagram
5R1	messages as part of a business rule
5D1	add message construct to business rule

Figure 13.2 Research Progress Map leading to Need for Four Constructs in the BRD

Why

A recurring theme in this research has been to recognise incompleteness in a business rule. From the earliest versions there were many signs and many attempts to address this problem as the above research codes illustrate. At each stage, the decision to add an additional construct was made on the basis that it aided completeness. In other words, without a particular construct a business rule was argued to be incomplete. This was so even before the refocussing to a business rule took place. As the analyst in the second action research study put it when asked about this '*Interesting question. Given that we actually used all those constructs what would I do without?'*.

There are a number of phrases in the definition of a business rule. The reasons **why** they are included in the definition statement are now provided.

The first phrase is the phrase '*explicit state change context'*. I believe there is a great deal of confusion or at least difference of opinion about exactly what is meant by a business rule. In this thesis it has been demonstrated that what many consider are business rules are what actually partial business rules. Firstly, it was demonstrated that some statements which are considered to be business rules are simply inaccurate. For example, the statement 'one customer has many credit orders' is inaccurate in the order processing case study because only good customers are entitled to credit. Secondly, another qualification to what many people would consider a business rule is that for example, the statement 'good customers may have many orders' is also inaccurate in the order processing case study because customers also need to have sufficient credit balance to pay for the order. The point is that such statements as 'only good customers may have many orders' are only part of a complete statement of what organisational policy might be for that particular event to occur. In terms of the above observations, a more complete statement of a business rule has been constructed in this thesis because any one cluster of states, events, conditions and signals brings together and makes explicit all partial statements of relevance to that rule.

The second phrase in the definition is '*change the state of a human activity system so that it will respond differently to external stimuli'*. One aspect that distinguishes the BRD from many (but not all) others in the literature is the state construct. Without the state construct as it has been used in this technique, there is a great danger in losing sight of what is important in the model. States are those anchor points that define events in a human activity system from an outside observer's perspective. Clearly other definitions and uses of states and events are possible so the question is why is my concept of state so crucial here. The answer is that a state is a perceived conceptual unit of measure of progress in a system. Recall that we have identified objects in a system because we see these as helpful conceptual entities reflecting the way we look at and understand a system. States are a further elaboration of that view. They represent conceptual milestones of progress within an object reflecting the stages we see as important in an object's life and therefore the whole system. Without a state as an anchor in a business rule, there would be basis for defining progress in terms of external conceptual yardsticks.

The phrase '*change the state of a human activity system so that it will respond differently to external stimuli'* was constructed to encapsulate the above ramification about a state. It is important that the state change can be defined in terms of what an outside observer may perceive and hence the reference to outside stimuli. Note also that as well as defining aspects of a state, this phrase has implications for the event. In other words, both state and event are defined reciprocally, ie states determine events and events determine states..

The next phrase to be considered is '*[events that] reinforce the constraints which govern a human activity system'*. Figure 13.3 is a copy of a business rule used earlier. Here the business rule is expressed in the form of a UBRD. As a general rule, it would be expected that in a business rule at least one state change would occur though some business rules can be far more complicated as figure 13.3 illustrates. For example, there are paths through this business rule which involve no state changes. The particular path taken depends on for example object states, attribute values and values supplied in the trigger. Consequently, some paths may involve no state changes. The phrase *'reinforce the constraints that govern a human activity system'* was chosen to legitimise situations where no state changes take place. After all such partial rules are still aspects of a business rule.

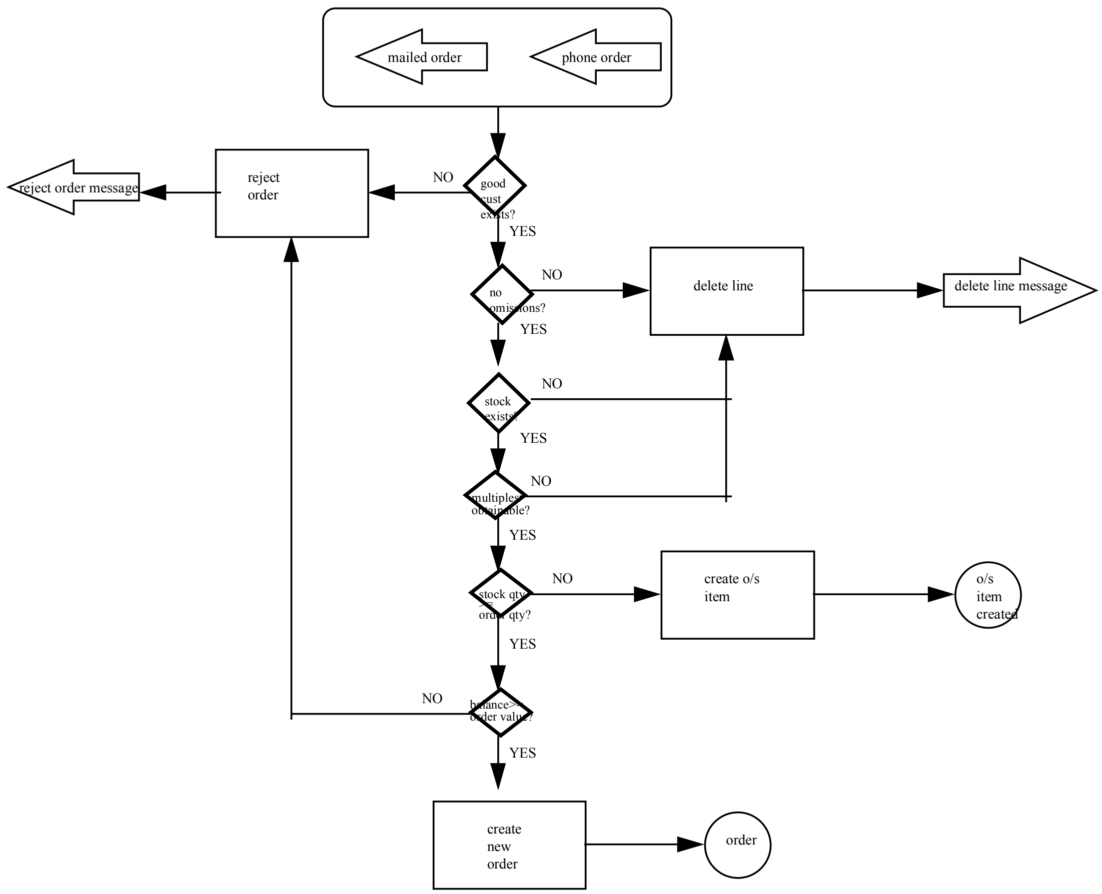


Figure 13.3 A Business Rule in the Form of a UBRD

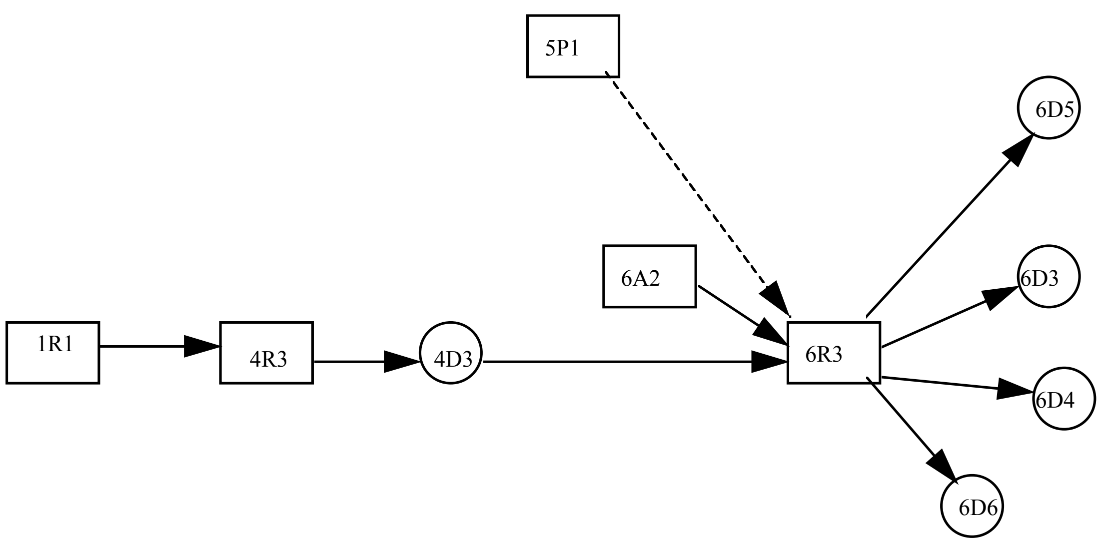
13.4 A Review of the Policy Rule

What

This research has demonstrated that it is possible to develop a model of policy rules using broadly the steps outlined in the BRDv3 paper and that further a model of policy rules is a useful first base in fully elaborating business rules. The evidence that it is possible to model policy rules is based on the fact that the systems analyst was able to produce a correct partial BRD from the sessions held with users. This was achieved in the face of what was arguably a diversion into exploring whether users were able to draw their own UBRDs with little guidance from the analyst. The steps outlined in the paper are a simple and straightforward method of constructing the BRD especially when it is considered that the analyst was using the technique for the first time himself. Of course assertions about modelling policy rules generally would have to be tested by further study or experimentation which followed exactly these steps, but nevertheless the fact remains that a correct partial diagram was able to be produced.

How

Figure 13.4 summarises the flow of argumentation with regard to policy rules graphically. Dashed lines indicate indirect or weak association not recorded in earlier research progress maps. In action research study 1, the issue of 'information systems events versus business events' was raised (1R1). This was in connection with the BED. It was noted that the existence of operational masters significantly



Legend:

1R1	IS events vs business events
4R3	IS rules vs business rules
4D3	refine definition of rule
5P1	business rules versus information systems rules
6A2	flowchart loops in the UBRDs
6R3	the level of abstraction problem
6D3	clarify policy rule

6D4 change notation for sequentiality of policy rule
6D5 retain the pure BRD and table
6D6 flexible approach for UBRD

Figure 13.4 Research Progress Map leading to Consolidation of Policy Rules

increased the number of symbols on the diagram and it was further suggested that this led to an increase in the complexity of the diagram. This issue remained unresolved until in 4R3 when reviewing the BRDv1 it was decided to redefine rules concerning operational masters in the Port Authority as processing rules rather than policy rules (4D3). At that stage the terms used were business and information system events rather than policy and processing rules. At this point also it was decided to move the rules concerning copies of transactions for audit purposes to processing rules.

Later, in the critical review, it was speculated that at least three levels of business rule can be identified, ie policy, processing and implementation rules (5P1). However, it was decided to focus only on policy rules in this research. So of particular relevance was how to distinguish between policy rules and other rules especially processing rules since these were considered to be the next lower level of abstraction to policy rules.

In the final action research study further issues emerged. In 6A2 the question of whether it was permitted to have flowchart loops in the UBRD was discussed. Amongst others, this issue was explored in 6R3 under the heading of the level of abstraction problem. While it was decided to be flexible in how users chose to draw the UBRD (6D6) it was considered that the pure BRD would be retained for use by the analyst (6D5). It was argued that adding loops increased the complexity of the diagram and that further at an appropriate level of abstraction, the device of output and input signals permitted the rule to be adequately described without recourse to loops. For example in figure 12.9 the existence of a trigger (ie input signal) of appointment letter and a message (ie output signal) of 'not before appointment time' was considered enough to indicate that the student may return at a later time to register into classes.

Thus it was decided that loops could be avoided in the pure BRD at least in the business applications so far investigated.

There were two further outcomes of the level of abstraction problem (6R3) of relevance here. In 6D3 an aspect of a policy rule was clarified in the following sense. It was observed that policy rules and processing rules may not be separate with respect to one another but rather processing rules may be elaborations of policy rules. The example given in chapter 12 was where students must be enrolled in a unit. Here it was argued that a processing rule elaborated from this policy rule would have to check whether the student was a valid student and that involved checking whether a student's PIN (Personal Identification Number) was correct. This additional check was an elaboration of the basic policy rule that students must be enrolled in a unit. The second clarification of a policy rule was with respect to sequentiality of policy rules (6D4). It was concluded that there needs to be a way of accommodating situations in policy rules where the sequence of conditions is important and also where the sequence is unimportant. Moreover, in order to cover the range of situations likely to occur in organisations, it was considered vital to allow flexibility in declaring a business rule. This would sometimes mean that processing rules would be elaborations of policy rules as well as separate additional rules.

Why

In this section the case that policy rules are an important first milestone in specifying the business rules associated with a human activity system is justified. A policy rule is a particular type of abstraction of a business rule. Abstraction here is the process of viewing aspects of what is perceived to be a reality or a perspective in an organisation. Abstraction is problematic in the sense that there are many possible ways of abstracting and many possible levels of abstraction. The 'how' section above has described the detail of issues associated with policy rules and the history of how these issues were clarified, but it does not in itself justify why policy rules are important in modelling business rules. So a justification is required which explains why policy rules represent one good initial model of business rules.

The Wand and Weber (W + W) model (1993) introduced earlier presented a view of development which involved many worlds or milestones in the development process. These worlds started from the 'real-world' and moved progressively to the 'machine world'. It seems reasonable to conclude that an early model involving business constructs and not more computer related constructs would be consistent with that view of development. So I argue that a model of policy rules is an obvious candidate for a first model of business rules.

However, the second action research study uncovered a different sequence of model development. Here, it was discovered that users were viewing the computerised system as the human activity system (with justification). On the face of it, this seemed to go against a traditional view of information requirements determination, since arguably much of what was contained in these UBRDs was design information. However, recall that a pure BRD was able to be constructed from the design oriented UBRDs. De Marco (1978) identified a similar phenomenon. De Marco's approach involved drawing 'physical' dataflow diagrams which captured much of the physical attributes of the existing system in much the same way as my users could be argued to be using the UBRD. The physical diagrams were then logicalised by removing these physical attributes so that a purely 'logical' (ie more abstract diagram) was obtained which was then used in subsequent analysis. In essence the same approach had been taken within the second action research study. Here the design oriented UBRD was distilled into a more abstract partial BRD. Having captured the policy rules in the BRD, a design model could be developed which may or may not reflect the original design oriented UBRDs. So, whether design oriented UBRDs are drawn or whether more classical UBRDs are constructed, it is concluded that policy rules in the BRD can be obtained from either.

Who, where and when

There are still questions over the distinction between policy and processing rules. For instance, it is unclear if rules regarding copies of transactions for audit purposes must always be demoted to processing rules. It is possible such decisions may be application dependent. To generalise and say that all audit rules should be demoted to processing rules is not justified. If the focal system were an accounting or even an auditing application then audit rules would very likely be policy rules.

Some of the evidence supporting this conclusion comes from a single case. To quote Pettigrew (1985, p242) '...*case study approaches cannot offer generalizability in the statistical sense but even single case studies are capable of developing and refining generalizable concepts and frames of reference'*. The same argument applies in action research study where phenomena in one organisation or application may be claimed to potentially exist in other organisations or applications, eg audit rules and processes exist in many applications and many organisations. Nevertheless, since this particular conclusion is a fundamental to the whole technique, a degree of caution is warranted.

13.5 A Review of the User Business Rules Diagram

Who, where and when

The same limitation applies to all conclusions in this section. So, rather than repeat it in each of the sub-sections, it is stated once at the outset. The decision to construct a separate diagram for users occurred relatively late in the research, ie at the end of the critical review (5D3). Therefore only action research study 2 was able to gather feedback regarding this important aspect of the model and that therefore is a limitation on the potency of any conclusions obtained. However, it will be recalled that user and analyst feedback was overall quite positive about the process of developing a UBRD and about the kind of symbols used and further the complexity metrics were quite favourable for the UBRD.

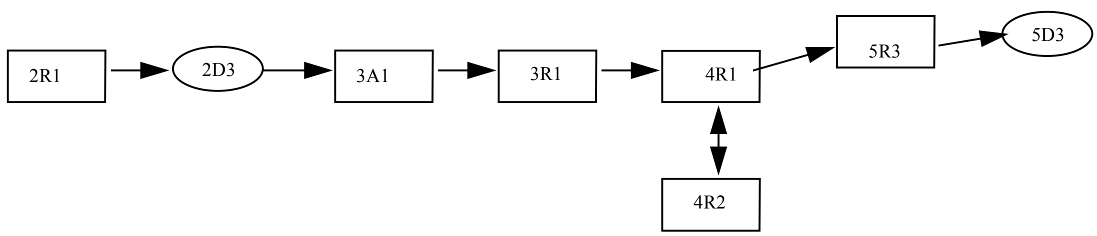
13.5.1 The Need to have Separate User Diagrams

What

It is concluded that it is better to have separate user diagrams which are limited to a single state change context.

How

Figure 13.5 summarises the history of the research leading to this conclusion. In action research study 1, there had been a great concern over the complexity of the diagram. Many working issues had been identified leading to a review of complexity in 4R1. These working issues were 2R1, 2D3, 3A1 and 3R1. It was also noted in 4R2, that the issue of complexity traded off with the issue of rule completeness (4R2). These issues were reviewed in 5R3 where separating user and analyst diagrams was considered. In 5D3 it was decided to implement this idea.



Legend:

2R1	readability
2D3	Harel blob
3A1	readability
3R1	complexity
4R1	complexity
4R2	rule completeness
5R3	separating user and analyst diagrams
5D3	implement user diagram

Figure 13.5 Summary of History of Decision leading to a Separate User Diagram

Why

The decision to implement a user diagram was made against the backdrop of concern over complexity in the diagram. At that time complexity metrics were employed to provide an impression of how complex a complete diagram (ie not a single business rule) might be. A UBRD since it is based on a single business rule (often a use case) by definition is only a subset of a complete BRD and therefore must be less complex. Figure 13.6 shows a very simple UBRD. Using the same basis for calculation as previously, the application complexity metric (McCabe 1976) is one which is very simple indeed. Such very simple UBRDs were very common in both the Portbilling application and the order processing case study.

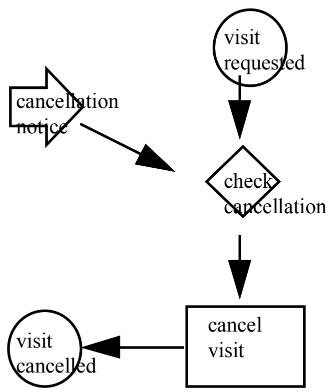


Figure 13.6 Simple UBRD from Portbilling Application

Figure 13.7 is also taken from the order processing case study and represents the most complex UBRD to be found in the research. Here the application complexity metric is 9. The important observation here is that overall the complexity of UBRDs are significantly lower than previously when the full diagram was used even though each business rule is now expressed in a rich manner involving all four types of construct. However, the generic complexity factor (Rossi 1997) remains at 6 (table 12.1). It will be recalled that in Rossi's paper the generic complexity factors for object class diagrams was around 12 to 14 and so 6 is considered quite reasonable given the inherent complexity of a business rule.

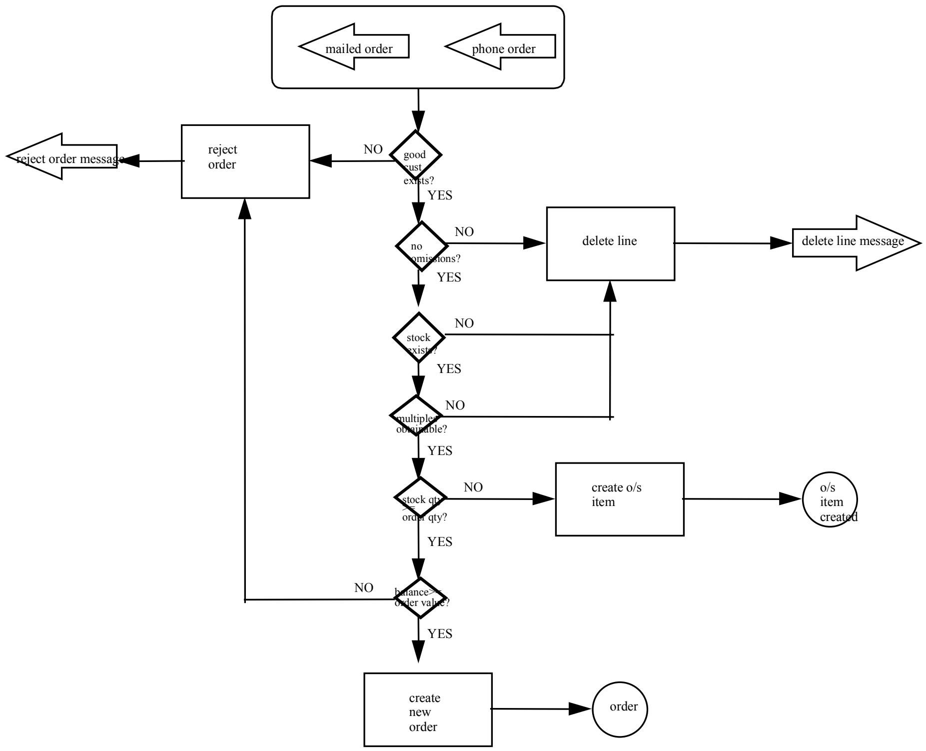


Figure 13.7 Complex UBRD from Order Processing Case Study

13.5.2 The Need for a Flexible Approach with the UBRD

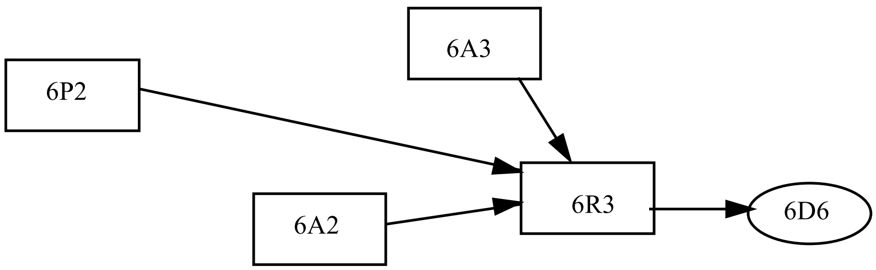
What

The notation for a user diagram should be flexible to accommodate a range of circumstances.

How

Figure 13.8 summarises the milestones of research progress leading to this conclusion. In preparing the academic paper prior to working sessions with users it was decided to design the UBRD with a flowchart ‘look and feel’ to it (6P2) because it was felt that this might well be intuitive to users. However, note that flowchart ‘look and feel’ tactic did not necessarily imply the use of flowchart loops; rather it implied only the sequencing of conditions. Having said that it was discovered that where these users were given leeway in the drawing of the UBRD they chose to employ flowchart loops (6A2).

Further, it was also discovered that they incorporated processing rules into the UBRD (6A3). This led to a major review of what was termed the level of abstraction problem (6R3) which resulted in amongst other conclusions that a flexible approach in working with the UBRD should be taken (6D6). Such an approach would permit a variety of styles of UBRD and would be contingent upon the preferences and experiences of the actual users themselves.



6P2	flowchart ‘look and feel’ to the UBRD
6A2	flowchart loops in the UBRDs
6A3	processing rules in the UBRDs
6R3	the level of abstraction problem
6D6	flexible approach for UBRD

Figure 13.8 Research Progress Map for Modelling Issues associated with UBRD

Why

The detailed rationale for concluding that a flexible approach was warranted is discussed fully in chapter 12. Figure 13.9 is a copy of the figure used in chapter 12 to summarise the main logic of the arguments used. It is provided here to the reader as an aid to memory only since much more detail appears in chapter 12.

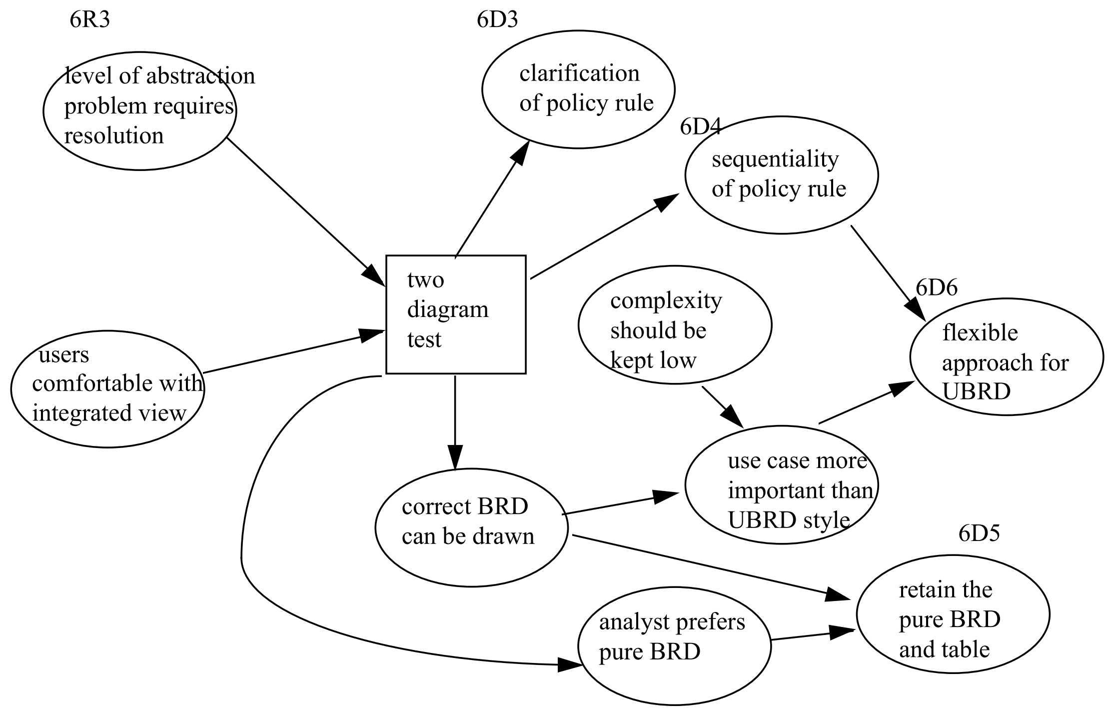


Figure 13.9 State-event Network for Levels of Abstraction Problem

There are two reasons why a flexible approach with the UBRD is sound. The first is that there may already be an existing computerised system to which users have been exposed in contrast to other situations where there is no pre-existing computerised system. Such a difference was demonstrated to be crucial in terms of the way that users perceived the human activity system. Any technique which purports to model business rules must therefore be flexible enough to accommodate the range of situations in organisational life today. The second reason is that users themselves have varying degrees of exposure and familiarity with modelling techniques. For instance one user in action research study 2, had some exposure to flowcharting, the other to entity-relationship modelling. Such exposure may have led to either positive or negative dispositions towards these techniques. So, by permitting a flexible approach to the style of UBRD drawn, the individual preferences of users can be accommodated in a constructive way thus earning commitment to the whole process. Such thinking is consistent with a post-modernist view (Drucker 1991).

Although it is concluded that a flexible approach would be permitted it is still appropriate to offer a stereotype notation, since there may be situations arising in which users have no preference or previous exposure to modelling notations. This strengthens the position of the technique as a flexible tool since there is additionally a fallback notation available and further the fallback notation can be used as an exemplar to orientate users.

13.5.3 The Need for Guidance (6D1)

What

In chapter 12, a conclusion was reached that users required some degree of guidance in the construction of the UBRD.

How

Figure 13.10 is a copy of the cognitive map used in chapter 12 to summarise the main arguments in support of the need of guidance.

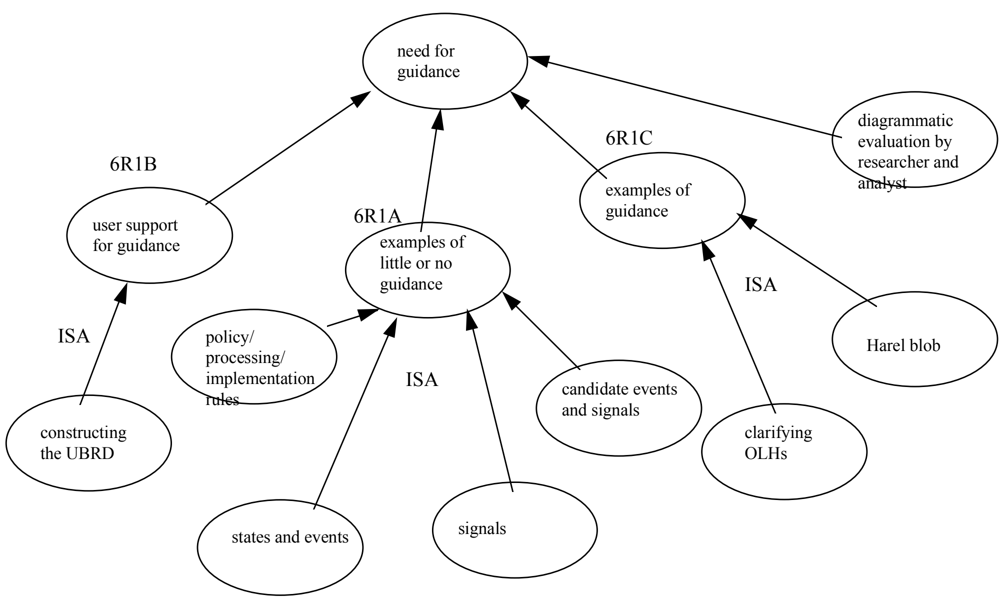


Figure 13.10 Cognitive Map supporting the Need for Guidance

It will be recalled that it wasn't originally planned that users would construct the UBRD by themselves. However, after the first meeting there was a feeling that the users understood the concepts of the UBRD sufficiently well to suggest that they may have been able to construct the UBRD with little or no guidance from the analyst (6A1). After the second meeting which was held a few weeks later, it was seen that such optimism was perhaps unfounded thus leading to a conclusion in the review stage (6R1) that guidance was in fact a necessary facet after all (6D1).

Why

Users vary in many ways. Some are independent; some are not. Some are highly computer literate; some are not. Perhaps a better term for this conclusion might be guidance support. In guidance support it would be necessary for the analyst to establish the degree and type of support if any required by the user group. Once that is established the analyst would then support the construction of the UBRD by supporting the group in those aspects only that it needs support. For instance suppose a particular group has already used the technique before. It may then be unnecessary for the analyst to do little more than refresh the memories of the participants, before embarking on the UBRD construction. In conclusion, since this particular group of users needed some support, it is reasonable to assume that groups of users in general may need some degree of guidance. The analyst will need to decide the degree and type of support required in each circumstance.

13.5.4 The Need for Validation (6D2)

What

It is concluded that validation steps need to be planned into the construction process of the technique. The need for validation refers to the need for review sessions with users the aim of which is to confirm or improve the quality of the deliverables.

How

Much of the empirical support for this conclusion came from the analyst who on three separate occasions during the interview suggested some form of walkthrough in order to ensure completeness and correctness in the diagram (6R2). Figure 13.11 is a copy of the cognitive map used in chapter 12 to summarise the main arguments in support of the need for validation.

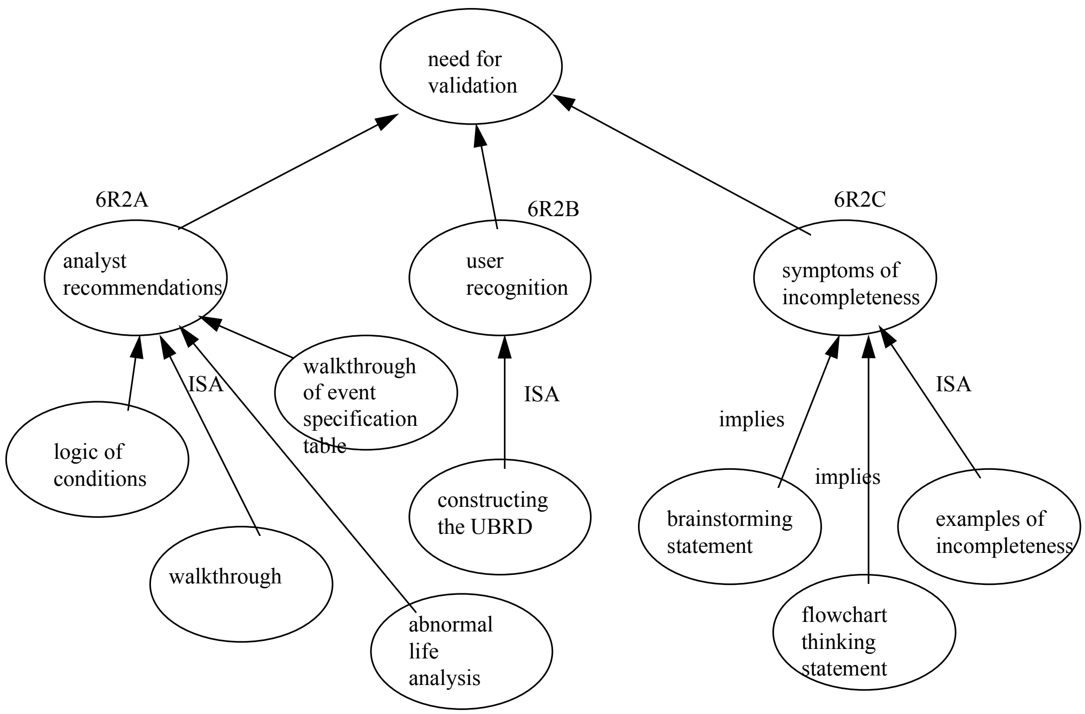


Figure 13.11 Cognitive Map supporting the Need for Validation

Why

Figure 13.11 indicates three main lines of argument which lead to this conclusion. The first line of argument comes from the analyst himself who recognised the need for validation through his reflection of the sessions conducted with users. The second argument came from one of the users who also recognised the need for some form of checking or validation of the correctness of what was done. The third line of argument came from what may be termed the symptoms of incompleteness, ie those side-effects which will become apparent if validation is not undertaken. A detailed discussion of these arguments appears in chapter 12.

The need for validation was seen as an important aspect of the construction process. The original SDD paper (McDermid 1990a) had explicated the role that validation could play in improving the quality of the diagram. Regrettably it was not possible to test the power of validation in improving quality in this thesis. Clearly this is a matter for future research and will be discussed in the next chapter.

Figure 13.1 shows arrows which indicate the need for feedback to earlier deliverables. It illustrates that in order to construct the BRD the analyst may need to go back to the users and use the UBRD as a medium for discussion and obtain answers to questions which then allow the BRD to be constructed. Similarly, as asserted earlier the EST contains more detailed information than the BRD and therefore the analyst in constructing the EST may have to refer back to the BRD or even users through the UBRD to obtain answers to questions in constructing the EST. So two occasions for validation are identified. The first is when constructing the BRD; the second in constructing the EST.

I consider validation to be an important mechanism for ensuring the completeness, correctness and consistency of the deliverables constructed (Moody and Shanks 1998). The evidence from the semi-structured interviews in terms of analyst and user statements as well as the symptoms of incompleteness is consistent with the view that validation is required. In the previous paragraph two obvious points were identified where validation could profitably take place.

Future research

It will be recalled that the analyst in action research study 2 proposed that the EST could be used as a vehicle for conducting walkthroughs. While the feasibility of such assertions would have to be tested before any detailed conclusion would be warranted, there is clearly merit in comparing the consistency between these three deliverables in some way as a means of ensuring consistency and correctness. This phenomenon of cross-checking between models is apparent in other approaches for instance SSADM (Downs et al. 1988).

13.6 The Business Rules Diagram

Given that there are three deliverables to this technique it follows that the BRD deliverable need not capture every feature or facet of a business rule. Specifically the EST can specify some aspects more concisely and directly than the BRD. The role of the BRD then becomes one which the analyst uses to build a merged picture from the individual UBRDs and is therefore not one in which it is attempted to capture all aspects of business rules. Thus it should represent a good visual description of many important aspects of business rules but not every aspect. If further information is required the EST can be examined to supply this information. The specific details of what the EST contains which cannot be found in the BRD are discussed in the next section. In this section only issues directly pertinent to the BRD or issues common to all three deliverables are discussed. The issues discussed are notational issues, the Harel blob, sub-states, construction issues and complexity.

13.6.1 Notational Issues

This section discusses straightforward notational issues associated with the BRD technique generally. Where an issue is of greater import than just its notation then this is discussed separately. For example simultaneity as an issue has notational implications. However, the question of simultaneity had a wider significance and was not only concerned with the symbols with which it would be drawn.

Notational issues were identified as a class of problem and were shown as a category in figure 11.2 as an extension to the original Wand and Weber model (1993). Every research code documented in the research progress appears in at least one section in this chapter. This section on notational issues groups all notational codes which otherwise would not appear elsewhere. To that extent, the conclusion recorded here is not necessarily of major import, but rather a bi-product of the action research process and is provided here for completeness.

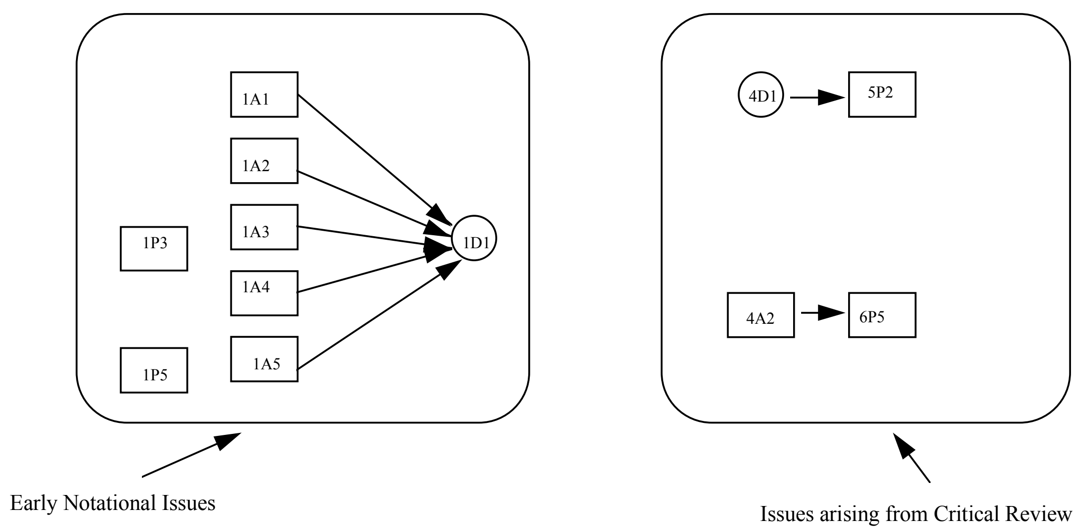
What

It is concluded that major changes in modelling focus tend to bring about a need for changes in notation and that such changes take time to settle down.

Two broad sets of notational issue were identified in this research. The first set were identified early in the first action research study and were concerned with the settling down of the changes initiated at that time. The second set of notational issues arose in the critical review and to a large extent reflected issues which had more to do with the change of focus in the research which was taking place at that time.

How

Figure 13.12 summarises notational issues from the research.



Legend:

1P3	group event notation
1P5	event notation refinement
1A1	prerequisite flaw
1A2	union and intersection
1A3	alternative point of entry
1A4	fan out
1A5	contingent dependency
1D1	change notation
4A2	readability and the simple condition
4D1	show conditions and triggers on diagram
5P2	trigger selection
6P5	the abolition of the simple event

Figure 13.12 Notational Issues associated with the BRD

There were two occasions when sets of notational issues arose. The first set occurred early in the research and represented problems concerned with the settling down of the notation generally. There was a need to depict that a number of events were to occur effectively simultaneously. This was resolved by inserting such events into a single bubble (1P3). In an attempt to reduce notational complexity a single symbol (circle) was used to describe all events (1P5). Previously three different symbols had been used depending on the type of event. There was also a flaw in the notation used to describe the prerequisite table (1A1). I had over-simplified a style of notation taken from De Marco (1978) and subsequently discovered that it was wrong to do so. A similar line of reasoning led me to consider the power of the graphical constructs and whether the combinations of union and intersection were sufficient (1A2). A minor enhancement was incorporated into the graphical notation to permit alternative points of entry into the diagram (1A3). Issue 1A4 concerned fan out dependency which was a simple graphical extension which aimed to slightly reduce visual complexity in the diagram. Lastly, a change to the way contingent dependency was shown on the diagram was instituted in an attempt to make the notation slightly more intuitive to analysts with experience of entity-relationship modelling (1A5).

The second set of notational changes occurred during the critical review and later. However, these changes really arose from the decisions taken at the end of action research study 1. Having introduced the trigger as a component of a business rule (4D1), there was a need to show situations where one of a number of triggers was possible (5P2). This phenomenon was called trigger selection. Lastly, in an attempt to reduce visual complexity, it had been decided only to show events if they were non-trivial (4A2). Because the UBRD was about to be introduced it was felt that this was now unnecessary and if the simple event were abolished this would result in more uniform display (6P5).

Why

What is evident here is the learning cycle which underpins the action research cycle. The **decision** to implement changes is followed by further **testing** of these changes which are then **reviewed**. If they meet the needs of the situation then no further action is required. If not, then further action, possibly in the nature of further change is required.

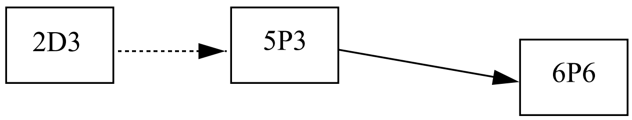
13.6.2 Harel blob

What

There is justification for using the Harel blob for showing trigger selection and complex conditions and these two uses therefore justify its inclusion as a construct in the BRD.

How

The Harel blob was incorporated into the diagram at the end of the second iteration in action research study 1, ie while making decisions concerning the BSDv1 (2D3). In the following iteration (ie the BSDv2) it was found that the combination of the Harel blob with the complex object appeared to be an effective way to model the problem efficiently. Since that time the Harel blob has remained an integral part of the technique. Figure 13.14 shows the history of issues associated with the Harel blob. While redrawing the diagram just before the critical review, it was noticed that it was strictly necessary to use a Harel blob round conditions which were relatively straightforward, ie where one of two possible events might occur. Thus this was rewritten explicitly into the revised paper (5P3). However the wisdom of this was reviewed in 6P6 where it was decided to omit the Harel blob in such situations for consistency's sake.



Legend:

2D3	Harel blob
5P3	Harel blob for simple conditions
6P6	refining the Harel blob

Figure 13.13 History of Issues associated with the Harel Blob

Why

This research has revealed two conclusions regarding the Harel blob. The first is that there is a need for a construct such as the Harel blob in describing business rules. The second conclusion is that there is still more work to be done in defining exactly in what circumstances to deploy the Harel blob and then in defining the most efficient notation for this purpose. A review of the Harel blob now follows in which these two conclusions are justified.

The Harel blob performs an encapsulation function. Originally it was used in combination with the complex object as a means of reducing the number of arcs on the BSDv2. Here the diagram captured only states and within a complex object, groups of states would often be predecessors to other states, eg in the Portbilling application portbill cancelled or portbill released for invoicing were predecessor states to the client cancelled state. However, with the shift to a focus on business rules and the relegation of processing and implementation rules, most of the circumstances in which the Harel blob was being used to advantage also disappeared. However, the shift to focussing on business rules also led to four constructs being used as opposed to earlier attempts where only one or two constructs were being modelled. It was found that the Harel blob could potentially be used to assist specification in each of the four constructs. These are now discussed.

The Harel blob has been used to show trigger selection, ie where any one of a number of triggers may initiate a business rule. It was argued that the ability to show trigger selection is an important facet of a business rule since it allows useful items of detail to be specified eg that either faxed or mailed order will be accepted.

The Harel blob has also been used in conjunction with conditions. Some conditions were termed complex. This was defined to mean that more than one condition contributed to a business rule. In action research study 2, an issue concerning the sequentially of a business rule arose. Here it was speculated that some business rules may require sequentiality in the specification of a complex condition whereas others may not. The particular notation for this requires further work. However, the need to show a number of conditions and to encapsulate them in some way was demonstrated by the fact that the business applications contained complex conditions which clearly need to be modelled.

Effective simultaneity of events has also been shown using the Harel blob. Whether this should continue as an aspect that is to be modelled at this level of abstraction or whether it might be better depicted at say the processing rules level is discussed in the section on simultaneity, but the broad position here is that the research has not demonstrated satisfactorily that it has to be modelled at the policy rules level.

The Harel blob was used in OLHs to capture parallel states, although it was argued that this was not necessary on the BRD itself. The question of sub-states is discussed shortly, but as with simultaneity it is not clear from the research whether this has to be shown.

Further research

The role of the Harel blob in showing simultaneity and parallel states needs further attention and so these two aspects are not included as part of a justification for the Harel blob. Also further work is required to clarify the notation as regards showing the sequentiality of a business rule.

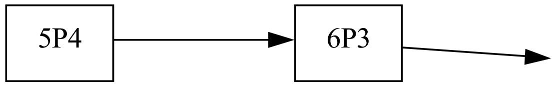
13.6.3 Sub-states

What

The research has been inconclusive with regard to the use of sub-states.

How

Figure 13.14 shows the history of issues associated with the need for sub-states. In 5P4 it was argued that at the time of specifying OLHs there may be insufficient knowledge about some states to be able to clearly distinguish them with respect to each other and so the use of sub-states to indicate parallelism between states was introduced. However, in 6P3 it was noted that the introduction of sub-states was inconsistent with the philosophy of states that had been applied up to this point. Specifically, it was argued that a state was an atomic unit of progress in the model and therefore a sub-state was either a state or not. So it was decided to reinforce this view by writing this into the description in the accompanying academic paper (6P3).



Legend:
5P4 sub-states and parallelism
6P3 the abolition of sub-states

Figure 13.14 Issues associated with the Need for Sub-states

Why

Identifying states is a very subjective activity. On the one hand it can be argued that credit-checked orders, overdue orders and the like are different stages in an order processing system. Thus they have different properties and so deserve to be described as states. On the other hand I have demonstrated that business rules can be modelled satisfactorily without using such (sub-)states. However, they can also be satisfactorily modelled with such states. So which is better and why? On what basis is better defined? Of course Occam's razor can be invoked which is to argue that if one can model successfully without a construct, then this is simpler and therefore better. Since an argument cannot be provided for why the use of sub-states can always be avoided, it is concluded that the issue of whether sub-states are required for modelling business rules remains unresolved. In other words, the range of applications examined thus far has allowed solutions to be proposed which have avoided the need for sub-states. However, that does not prove that there is no need ever to have sub-states in the BRD. There may well be other applications in which showing sub-states has some advantage.

Future research

In the meantime, using sub-states on the BRD will be avoided until a situation is discovered where it cannot be avoided. So more examples of real business applications need to be investigated to gain more experience with this problem. Equally, more investigation at the theoretical level is also warranted. One useful avenue of research might be to look at situations in the real-time domain where parallelism is modelled.

The question there however, is to what extent such domains have a bearing on human activity systems.

13.6.4 Construction Issues

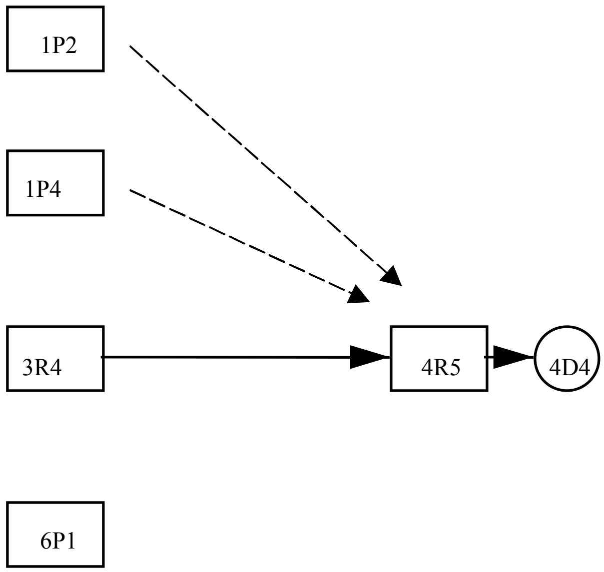
What

This research has shown that it is possible to construct a BRD from UBRDs in the manner outlined in the BRDv3 paper. An important corollary to this is that with due qualification it is considered that there is sufficient evidence to speculate that a third approach to developing information systems, involving business rules as the initial and prime modelling medium, is possible.

How

Figure 13.15 shows construction issues associated with the BRD generally (as opposed to issues specifically associated with the UBRD). In 1P2 it was speculated that the BRD could be constructed from an object model as opposed to a data model. This was demonstrated in the subsequent iteration of the diagram by using object models and OLHs as the starting point for the construction process, but only in the case study description. Later it was speculated that the diagram could be constructed without drawing either a data or object model at all, ie direct from complex objects (3R4). This was reviewed in 4R5 and in 4D4 it was decided to test the direct approach in the next iteration of the diagram. The direct approach was the basis for construction in action research study 2.

In 1P4 the steps of the construction process were modified to include an additional step examining delays between events. The outcome of this step was to identify situations where events could occur effectively simultaneously. However, a detailed examination of simultaneity occurs elsewhere in this chapter. Lastly, in 6P1 a minor alteration to the construction process was suggested in which candidates were categorised into events, triggers and messages.



Legend:

1P2	object-oriented basis
1P4	delay analysis step
3R4	construction from complex objects
4R5	direct vs indirect construction
4D4	direct approach
6P1	categorising events, triggers and messages

Figure 13.15 Construction Issues associated with the BRD

Why

The argument for this main conclusion here is that a correct partial BRD was able to be produced by the analyst. This was further corroborated by the analyst and the users in their interviews when they endorsed the construction process generally. Figure 13.16 shows the cognitive map in which this conclusion was discussed.

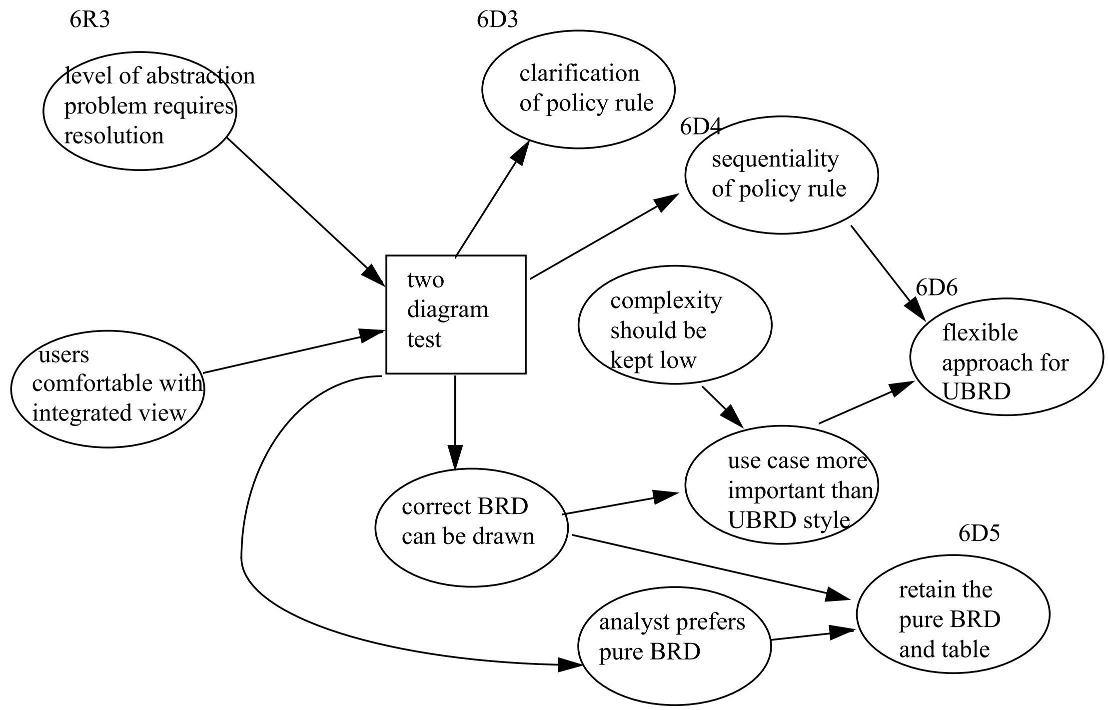


Figure 13.16 Cognitive Map supporting Conclusion that a correct partial BRD could be drawn

Detail of the construction steps has been included in all of the versions of the diagram. The relative lack of construction issues emerging from the research I believe is partly explained by the attention given to construction throughout the research. Of course I do not contend that I have developed the only or the best method for constructing the BRD. Nevertheless, it is still justified to be positive about construction generally to the extent that a demonstrably correct but partial BRD was obtained from the process outlined.

One novel aspect of this construction approach is the relatively direct approach taken to construction. Only a basic outline of each object and its object life history was required to begin construction of the UBRD. I can find no evidence in the literature where a similar direct approach is taken. Such an approach if it can be verified across a broader spectrum of cases has important implications for systems development. Traditionally there has been a debate about whether one should follow a data-based approach to developing information systems or a process-based approach (McDermid 1990b). Even with the development of object-oriented thinking there is still, in my view, an implied binary perspective in terms of the attributes and methods implicit in an object. This research has raised the following interesting issues. These are:

- whether states of objects should be considered as fundamental as the constructs of attributes and methods,
- whether states can be identified before attributes and methods and
- whether a third approach to developing systems can be construed as 'business rule' based development.

Who, where and when

A significant limitation of this conclusion is that it is based on the production of a partial BRD by one analyst in one study. Clearly, the production of a complete BRD in several studies by different analysts would make this a much stronger conclusion.

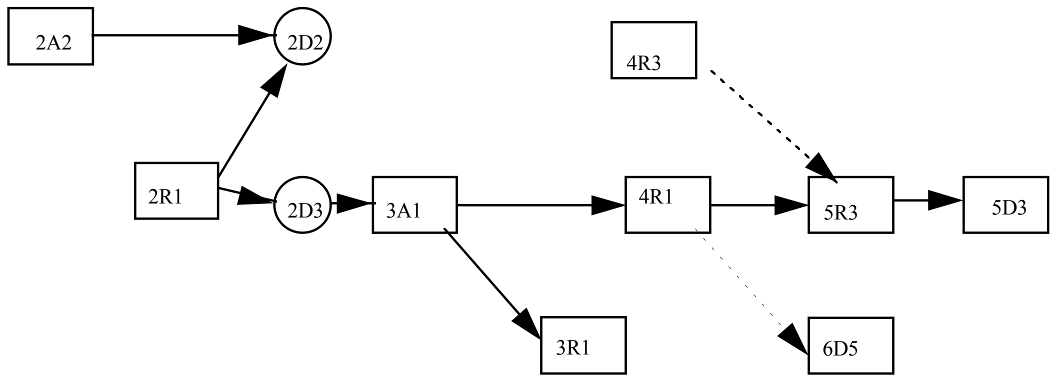
13.6.5 Complexity

What

A necessary degree of complexity is inherent in modelling business rules, but it is possible to introduce tactics to manage complexity.

How

Figure 13.17 shows the history of issues associated with complexity. Because complexity trades off with completeness it would have been possible to depict a more comprehensive diagram than figure 13.17 showing the inter-relationship between complexity and completeness. However, for the purposes of this review it was decided to keep figure 13.17 simple. Further, other minor issues have been excluded from this discussion which are connected with complexity where they are covered elsewhere. For example, issue 4A2 which is entitled 'readability and the simple event' is dealt with elsewhere.



Legend:

2A2	overkill
2R1	readability
2D2	complex object
2D3	Harel blob
3A1	readability
3R1	complexity
4R1	complexity
4R3	IS rules vs business rules
5R3	separating user and analyst diagrams
5D3	implement user diagram
6D5	retain the pure BRD and table

Figure 13.17 History of Main Issues associated with Complexity

In issue 2A2 the potential problem of overkill was raised. Overkill itself is simply a problem concerned with redundantly identifying states in two or more objects. However, a detailed examination of this issue caused speculation that use of complex objects would reduce the probability of overkill occurring. Later in the review step of the same iteration of the study, a general concern over the number of symbols and therefore the readability of the diagram was raised (2R1). These two issues led to the conclusion that the way forward was to use the complex object as a means of reducing complexity (2D2). In the same review step it was also decided to use the Harel blob as a means of reducing complexity. It was considered that the Harel blob would reduce complexity by reducing the number of arcs on the diagram and by providing a way of visually grouping like items together (2D3). In 3A1 a first attempt at redrawing the diagram was made using only the Harel blob, ie without the complex object. This was not successful. It was found that the combination of the Harel blob and the complex object seemed to be much better at reducing complexity. In 3R1 a review of complexity was undertaken. Although it was felt that inter-object complexity had been reduced it was also clear that the latest diagram was more complex than earlier versions. In 4R1 the issue of complexity was discussed again where it was concluded that problem of complexity was getting worse. In 4R3 it was argued that only showing business rules (policy rules) would have an important impact on complexity especially in the light of the trend to include more constructs in the diagram.

In 5R3 the possibility of separating user and analyst diagrams was raised. It was proposed that by limiting a user diagram to a single business rule that complexity for users would reduce dramatically. This was implemented in 5D3. However, complexity in the analyst's diagram still remained high. In the critical review chapter, analysis indicated that with four constructs complexity became very high indeed. Figure 13.18 is a copy of figure 12.9 in which the steps and conclusions leading to the decision to retain the pure BRD and table for analysts was made (6D5). Of particular note is the view that a correct BRD was able to be drawn from versions of UBRDs prepared by users.

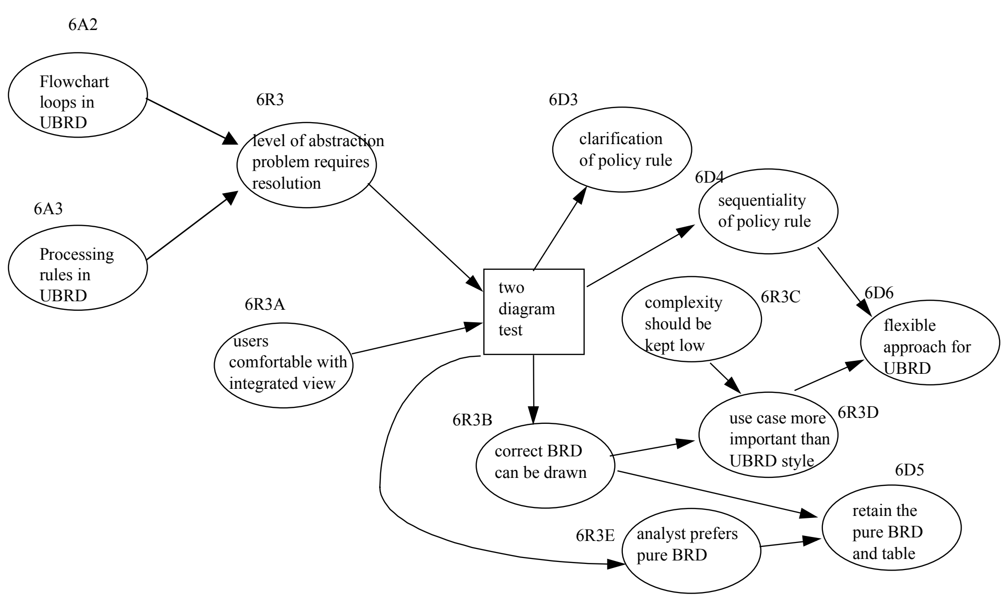


Figure 13.18 Cognitive Map supporting the Conclusion to retain pure BRD

Why

Complexity in various forms has been an issue which has been present throughout this research. Aspects of complexity sometimes emerged disguised under different labels. For example, the term readability was used at times to refer to anticipated problems associated with reading the diagram essentially due to the number and variety of constructs in the diagram.

The struggle with complexity has been to keep complexity as low as possible. However, behind the issue of complexity was a trade-off between complexity and rule completeness. It has to be understood that if a problem is complex then any diagram which adequately models that problem is also likely to be complex. This research has identified that the issue of complexity is inherent in the nature of a business rule as it has been defined in this thesis. In other words given that it is agreed that a business rule comprises elements of state, event, condition and signal then it follows that a degree of complexity is consequent in any diagrammatic technique attempting to model business rules. This is based on the observation of the growing levels of complexity that ensued as each new construct was introduced into the diagram. A corroboration for this was the systems analyst's view from the second action research study in which he stated he preferred the pure BRD and table thus implying that a degree of complexity is probably inevitable given the problem on hand.

13.7 The Event Specification Table

The event specification table would typically be the last deliverable to be completed within the BRD technique. It contains more detailed information on a number of facets of business rules and these are not shown in the BRD. There are two reasons why certain features are not shown on the BRD. The first is to avoid complexity in the diagram, ie although it may be possible to show certain information in diagrammatic form it has been decided not to do so since it would increase complexity. An example of this is a decision discussed shortly to show simultaneity in the event specification table but not in the diagram. The second reason is where it is not possible (at least easily) to show such information on a diagram. An example of this is showing manyness on the diagram. Again this is discussed shortly. Research progress with respect to the EST is summarised under three headings. These are the emerging role of the EST, manyness and simultaneity.

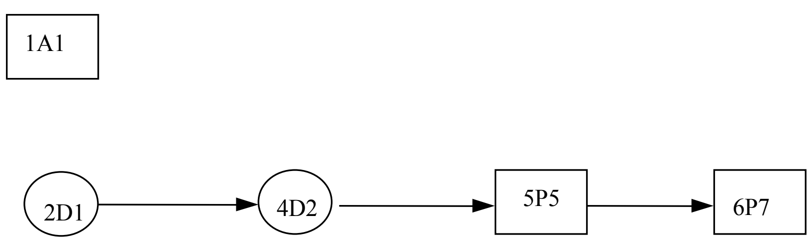
13.7.1 The Emerging Role of the EST

What

There is evidence which points to the need for a repository which contains a complete specification of business rules. Because of the potential complexity of a business rule, it is concluded that it is not practical to show complete specifications diagrammatically and that the EST is a candidate repository.

How

Figure 13.19 traces the history of issues associated with the EST with the exception of manyness and simultaneity which are dealt with separately.



Legend:

1A1	prerequisite flaw
2D1	extend prerequisite table
4D2	prerequisite table
5P5	event specification guide
6P7	changes to event specification guide

Figure 13.19 History of Issues associated with the Event Specification Table

Issue 1A1 is shown on this diagram for completeness. It refers to a notational problem with the prerequisite table. In issue 2D1 a decision was taken to extend the prerequisite table. The extensions were to include pre-conditions such as ‘customer’s credit greater than order value’ and post-conditions such as simultaneous events or time delays. These extensions were added to address a problem of ontological incompleteness perceived at that stage. In issue 4D2, the role of the prerequisite table was extended to show manyness since it was found difficult to do so on the diagram.

In 5P5 a further refocussing of the role of the prerequisite table was taking place. Foremost was the idea that the event is the central construct around which the other constructs in the business rule are connected. It was on this basis that the prerequisite table was renamed the event specification guide. Sorted by event, each row in the guide represented one path through a business rule, which in itself was an acknowledgement that different paths existed within a business rule. Clustering paths by event was considered a good way of showing completeness in a business rule over a few rows in the table. At that time, showing simultaneity and time delays was dropped from the table because it was felt unnecessary for these aspects to appear in both the table and diagram.

In 6P7 some additions to the event specification guide were discussed as a consequence of separating the user and analyst diagrams and it was decided to incorporate these into the renamed event specification table. These changes were to add triggers, messages and post-states to the table. The rationale for adding triggers and messages was a straightforward reflection of the fact that these were components of a business rule and so for completeness ought to be added to the table. Thus each cluster of rows representing an event was becoming much more of a complete specification of a business rule. Similarly adding post-states also completed the specification for each path associated with an event.

Why

Over its lifetime, the EST has been also known as the prerequisite table and the event specification guide. As the prerequisite table it was used to build up information on prerequisite events or states prior to constructing the diagram itself. As the event specification guide its role was more of a complement to the diagram in which information about manyness and the logic of conditions was shown.

This research has detected a trend which points to the role that the event specification table can play in capturing business rules. The trend has been towards recording aspects of business rules which are not considered advisable on the diagram itself. The previous discussion on complexity is relevant here since the difficulties of showing all aspects of a business rule on a single diagram contribute to the conclusion that some other solution has to be found. Using the event specification table as a complete and detailed repository of business rules is one solution to this problem and as such would be a key deliverable in later stages of developing an information system. In the BRDv3 completeness in the EST is defined to mean recording events, triggers, messages, conditions, pre-states and post-states. Further, manyness can be inferred from events, pre-states and post-states and simultaneity can be inferred from occurrences of more than one event in the same row.

Future research

Given that it is concluded that the role of the event specification table is that of repository two matters follow. The first is to do with the logic of conditions. As it is currently specified, the conditions column simply lists conditions and indicates in bold (previously with a bar) whether the logical negative of a condition applies. In 6D4 it was recognised that business rules may not be sequential and so a notation for this would need to be devised for the event specification table. Further, it is speculated that some business rule conditions may be related to each other using non-trivial combinations of union, intersection and negation as well as sequentiality and non-sequentiality. Such speculation is reasonable given the discussion in 6D4 and in the absence of evidence to the contrary. So it would seem sensible to devise a notation which accommodates such contingencies eg a more comprehensive and mathematically-based notation.

The second matter concerns post-conditions. Following the trend towards completeness, it would seem logical to specify post-conditions in the event specification table also. Post-conditions refer to the specification of changes in the value of any attribute stored in any object. For instance in accepting an order the quantity in stock of an item would be reduced by the quantity ordered and would be one consequence, ie a post-condition of the event accept order. Including post-conditions would re-inforce the role of the event specification table as a repository of information concerning business rules. Such post-condition information would not be available in the diagram.

In one respect it should be recognised that this move towards completeness in specification will bring the BRD closer to some of its competitor models discussed in the literature review. For instance, using pseudo-mathematical expressions, Kung and Solvberg (1986) were able to express post-conditions amongst other characteristics in their diagrams. So far I have taken the view that the tabular approach (eg the EST) is more user-friendly than a pseudo-mathematical notation. However, the move to specifying post-conditions suggests that this assumption should be re-examined. The over-riding criterion of course should be how well the EST or alternative is able to perform its function in a social context. Further research is required to establish the relative merits of a tabular approach against a pseudo-mathematical approach taking account of user-friendliness as well as expressiveness.

13.7.2 Manyness

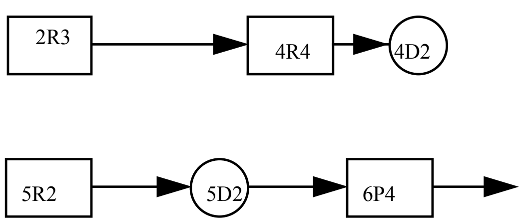
This sub-section deals with the issue of manyness. It is discussed here because it is recommended that manyness is recorded in the EST and not in the BRD. The issue referred to as the one-to-many business rule is also discussed here because it is related to the manyness issue.

What

Manyness should be shown in the EST and not in the BRD.

How

Figure 13.20 shows the history of issues directly associated with manyness. In 2R3 I coined the term manyness to describe the problem concerning the number of instances of an object which contribute to the execution of an event. At that time neither the prerequisite table nor the diagram (BSDv1) had any mechanism for showing manyness. I argued that manyness was a possible candidate for modelling since it added extra relevant information to the specification. This issue remained unresolved until 4R4 in which it was proposed to add a symbol to the prerequisite table in indicate manyness. This was decided in 4D2. When the focus of research changed to the business rule a related problem emerged which was referred to as the one-to-many business rule (5R2). It was observed that a common type of (partial) business rule is of the form one instance of object X is related to many instances of object Y and in 5D2 it was resolved to attempt to find a graphical solution for this. In 6P4 the rationale of the solution to the problem was described. This involved the use of a line and the graphical rule that the default for relationships crossing the line was of cardinality one-to-many. Though not totally comfortable with this solution it was decided to implement it anyway in the hope that the final action research study would shed some light on the problem.



Legend:

2R3	manyness
4R4	manyness
4D2	prerequisite table
5R2	one-to-many business rule
5D2	implement one-to-many business rule
6P4	the one to many business rule

Figure 13.20 History of Issues associated with Manyness

Why

The problem with showing manyness is a topological one. Diagrams can be very powerful visual aids because employed judiciously graphical symbols depict only the minimum necessary to describe a phenomenon. This research has uncovered that some business rules are complex in the sense that there are many possible routes through a business rule and that some of these routes may lead to state changes. The example of the business rule accept order in the order processing case study illustrates this well. It follows that if manyness is to be depicted graphically it is necessary to be able to trace each route separately through the business rule with respect to manyness. However, simply adding a manyness symbol to the state symbol is insufficient since many routes might involve that state but in a different way as far as manyness is concerned. It is therefore concluded that manyness should not be shown on the diagram but rather as an aspect of the EST. In the EST a separate row is given over to each route in each business rule, so it considered feasible to depict exactly the nature of manyness pertaining to that route.

Who, where and when

It would have been ideal if it had been shown possible to capture 'one-to-many' type business rules diagrammatically, because then it would have been possible to conclude that all types of business rule identified as a typical list in chapter 2 were able to be modelled graphically. A limitation of the project is that it has not been possible to capture all types of rule diagrammatically. Most types of rule have been captured on the diagram, but one is only detectable from the EST.

13.7.3 Simultaneity

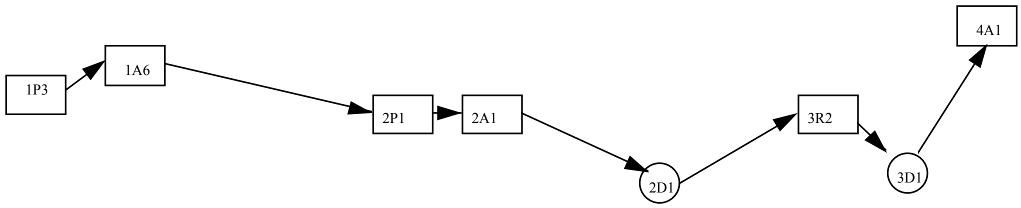
The problem of simultaneity has persisted throughout this whole research. Like manyness it has been argued that simultaneity is a useful property in describing business rules. Simultaneity concerns where two or more events effectively have to be executed at the same time. An example from the order processing case study is where because of abstraction into certain objects, the deletion of outstanding order-lines has to happen at the same time as the creation of normal order-lines.

What

The research is inconclusive regarding if and where simultaneity should be shown. As an interim measure it is proposed that simultaneity can be depicted in the EST.

How

Figure 13.21 shows the history of issues associated with simultaneity. In 1P3 a group bubble was used to encapsulate simultaneous events. In 1A6 a problem arose in the BED because some events had to be repeated several times. For example, the event 'create portbill audit' had to be listed in three different bubbles and this was considered undesirable. The move to modelling in states created a problem in that the group event was no longer a modelling construct. The alternative proposed was to use a double-headed arrow (2P1). However, in applying this to the business application (2A1) this caused problems with the audit states. Here the problem was a topological one. On the one hand it was preferred to declare each state only once on the diagram. On the other hand the use of the double-headed arrow implied that four states were simultaneous when in fact it was required to show that three pairs of two states were simultaneous. To avoid this three identical states had to be shown. This led to a decision in 2D1 to use the prerequisite table to show simultaneity. This was reviewed in 3R2 and considered successful. However, at the time it was still felt that the ability to show such information on a diagram in principle was desirable if it was feasible to do so and so also during this review codes were added to a diagram as a test. In 3D1 there was a decision to model in both states and events. One of the advantages of this decision was to be able to group simultaneous events within the same bubble. In 4A1 it was seen that although for inter-object simultaneity the use of the group bubble was effective, it didn't help with situations of so-called intra-object simultaneity eg where audits are conducted for each major event in an object's life. So it was decided, although clumsy to retain two methods for showing simultaneity, ie the group event bubble and the double-headed arrow.



Legend:

1P3	group event notation
1A6	simultaneity
2P1	simultaneity
2A1	simultaneity
2D1	extend prerequisite table
3R2	simultaneity
3D1	states and events
4A1	simultaneity

Figure 13.21 History of Issues associated with Simultaneity

Why

The reason why there is a need to show simultaneity in a system arises because a system has been partitioned into objects. Perhaps in some systems the partitioning will be such that there are no simultaneous events. However, as the business applications in this research have demonstrated, even systems partitioned into objects which have low coupling with respect to each other may well have simultaneous events. So it can be argued that simultaneity is a consequence of partitioning which itself is a fundamental approach to solving many types of problem including any approach which involves object-orientation as a solution, ie simultaneity is a consequence of object-orientation.

Perhaps a more pertinent question here is whether it is necessary or desirable to show simultaneity at this level of abstraction. Although it can be argued to be not necessary, it is much harder to answer whether it is desirable. To answer this would involve examining the other lower levels of abstraction identified, namely the processing and implementation levels. It would then be possible to compare the relative merits of showing simultaneity at different levels of abstraction.

If it is decided that it is to be specified at the policy level, the question then becomes in which deliverable. Conceivably it could be implemented in the UBRD, BRD, EST or any combination of these. Since an overall principle in this section has been to use the EST as the repository for items for which it is not practical to define elsewhere, my feeling is that simultaneity is perhaps best specified in the EST. Having said that, the most immediate question is whether simultaneity is an issue which can be delayed until processing or implementation models are specified.

So, although this research has concluded that simultaneity can be modelled at the policy level, perhaps best in the EST, the research itself has not provided strong argumentation to conclude that it must or should be recorded at this level of abstraction.

Future research

Further consideration of later models such as processing and implementation models is required before a conclusion can be drawn.

13.8 Critique of Action Research and Intellectual Framework

Action research as an approach was reviewed critically in chapter 11. Specifically in section 11.3, action research as a methodology was reviewed because a subsequent study (ie the second action research study reported in chapter 12) was planned. So the detail of that will not be recounted here. However, in reflecting on the second action research study, one major issue emerged which requires discussion in the context of action research as a research vehicle or approach. The issue is that of control. In the second action research study, it was intended that the validation aspects of the technique (ie structured walkthrough, amendment analysis and abnormal life analysis) would be thoroughly investigated. However, that effort was deflected because of the problem that emerged regarding users drawing their own diagrams (ie the two diagram test). The problem presented a classical trade-off to me as a researcher. On the one hand it was important to explore the role of validation that had been designed into the technique. On the other hand, there was clearly a more immediate question that required examination in terms of the drawing of the UBRD. In the end the immediacy of the UBRD problem won out over the less immediate problem of validation, but it was not an easy decision to make. Indeed, in taking this decision the whole question of to what extent should a researcher try to exercise control over the direction of the research arose. In this context **control** might inhibit the natural **emergence** of new ideas and conclusions from the research and was thus something that required a great deal of consideration in terms of the overall goals of the research and the research question itself.

Below, the intellectual framework used throughout this thesis is now critically reviewed in order to provide an assessment of its utility. Overall my position is that the use of a framework was a significant benefit to this research. Having reflected upon it in depth, I consider that most of the arguments against using a framework are hollow compared to the advantages of using a framework.

There are four steps in this discussion. Firstly, the arguments **against** using a framework are proposed and the merit of each is evaluated. This is followed by a sub-section which rehearses the arguments **for** using a framework. This leads into a third sub-section which examines key issues arising out of the previous two sub-sections. The final sub-section summarises the conclusions reached.

13.8.1 Arguments against using a Framework

Figure 13.22 is a cognitive map which shows the broad arguments against not using a framework.

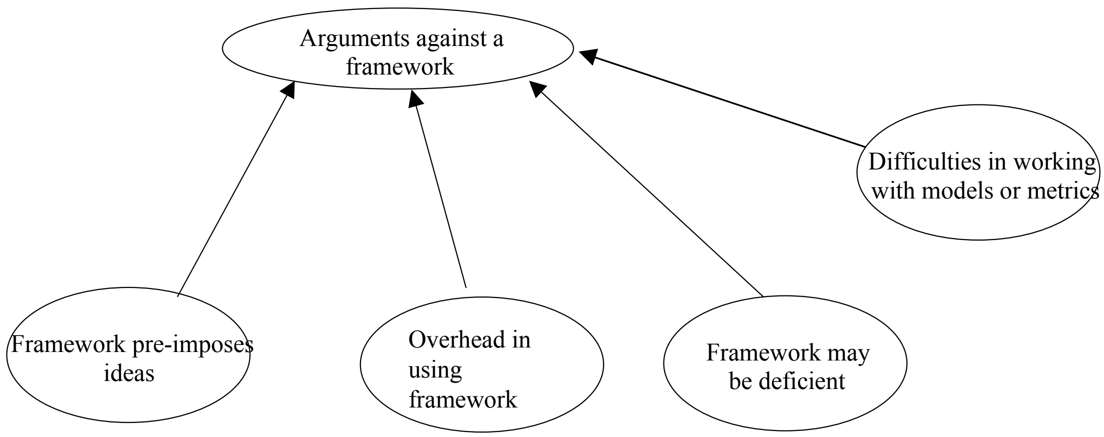


Figure 13.22 Cognitive Map showing Arguments against using a Framework

One argument for not using a framework is that it cannot be determined what conclusions would have emerged had the framework not been used. In this context, a framework can be seen as a restricting influence which perhaps guides a research down particular lines by virtue of the ideas in the framework. Unfettered by a framework, arguably any conclusions reached would emerge naturally from feedback obtained from the research itself and thus be free of any pre-imposed views or values.

A second argument recognises that there is a cognitive overhead in using a framework. Clearly, the framework has to be learnt by the researcher and also its concepts and its implications carried by the researcher while the research is being undertaken. This argument would have merit if there was no perceived value in using a framework. However, as is my position, where there are benefits arising out of using a framework, and these are perceived to outweigh any disadvantages, then this argument is in itself a poor one.

A third argument relates to the observation that there were many occasions in which the intellectual framework was deficient and therefore unable to provide appropriate guidance. For example the original framework did not provide any guidance for assessing complexity or in discriminating alternative pathways for construction. It follows that success is effectively predicated on the wisdom of the initial choice of intellectual artefact(s) incorporated into the framework. Inappropriate artefacts would inhibit research progress. However, this argument is to a large extent a hollow one, because the framework was defined within an iterative learning cycle which permitted new ideas to be incorporated in the next cycle.

The fourth argument is a pragmatic one and relates to difficulties with the metrics or models used within the framework. It may be said that a framework is only as good as the metrics and models that it contains. During the course of this research it was quite clear that the metrics and models utilised were not perfect. Two examples are cited here. Consider the metric for complexity. This was based on McCabe's cyclomatic complexity metric. This metric was developed for establishing the complexity of a computer program or module. The extent to which it can be applied as a metric to quantify how well a systems analyst or indeed a user can understand a business rules diagram is not clear. This was further complicated by the fact that when a BRD was converted to a control-flow graph there were many entry points to the graph representing many event-driven situations in an information system. Again, McCabe's metric was not really designed for this since structured programs can always be written with a single entry point (McCabe 1976, Watson and McCabe 1996). In any case, it is debatable just how precise any metric really is. In other words, surely it is the case that all metrics are approximate? For example, the measurement of time is approximate even with atomic clocks. Perhaps a more relevant question here is whether the metric is good enough. In other words, as long as it allows the researcher to make reasonable conclusions, then it may be considered fit for its purpose. With regard to McCabe's metric, though perhaps a surrogate measure, most conclusions were based on quite stark differences. For example, the full BRD in one application had a complexity factor of 79, whereas the UBRD had a factor of less than 10.

A second example illustrates a different aspect of this argument. The original Wand and Weber (W + W) framework (1993) had a single mapping between the ontological world and the design world. This was subsequently extended to show many design worlds because the multiple steps in constructing the BRD required a more sophisticated model to illustrate the mapping process. So, although there were difficulties in utilising the original model, it was possible to amend it and thus obtain benefit.

In summary, I conclude that only the first argument, ie that the framework pre-imposes ideas, carries some weight in this discussion.

13.8.2 Arguments for using a Framework

Figure 13.23 is a cognitive map summarising the arguments for using a framework.

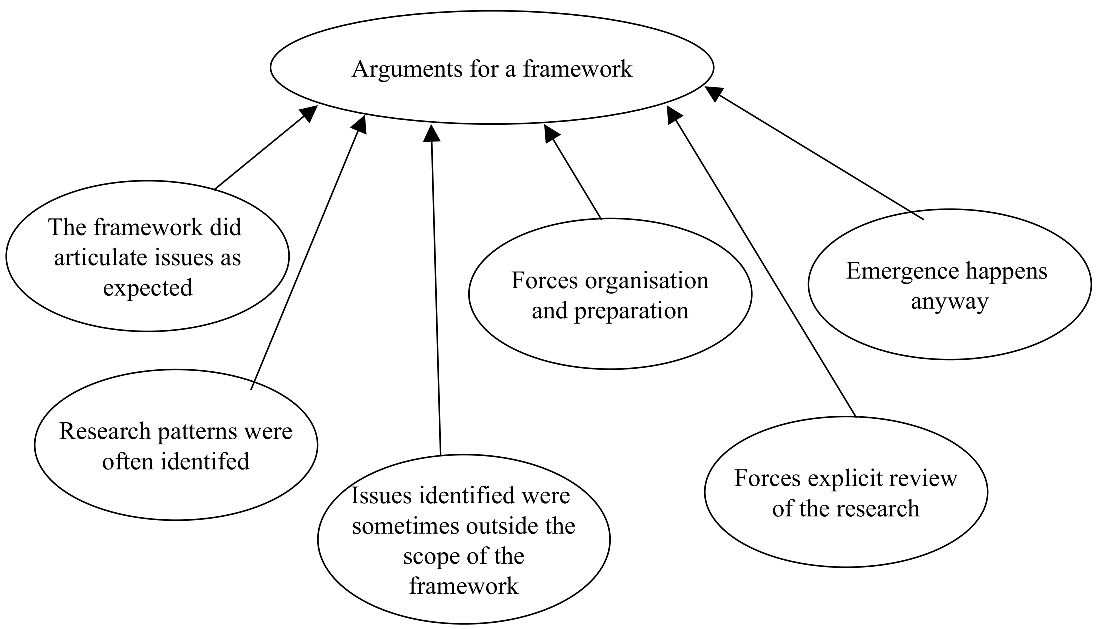


Figure 13.23 Cognitive Map showing Arguments for using a Framework

The six arguments on the cognitive map are now discussed. The first argument is that an intellectual framework permits articulation of issues or, put another way, the framework did do its job on many occasions. The W + W framework was selected because it focused on mappings between the ontological world and the design world and what might be the symptoms if such mappings were poor. Table 13.1 shows examples of articulated issues from the research. These were discussed in detail in chapter 11.

For research associated with developing new diagrams, this kind of abstraction was initially considered ideal for detecting flaws or weaknesses in the design of the diagram. The fact that using the W + W framework did permit articulation of a wide range of flaws is testimony to the general argument for using an intellectual framework.

Research Code	**Description**
1A1	prerequisite flaw
1A2	union and intersection
1A3	alternative point of entry
1A6	simultaneity
2A1	simultaneity
2A2	overkill
2D2	complex objects
2D3	Harel blob
4A1	simultaneity
4D1	show conditions and triggers on diagram
5P1	business rules versus information systems rules
5P4	sub-states and parallelism
5P5	event specification guide

Table 13.1 Mapping Issues identified during Research

Secondly, the experience of this research is that various patterns emerged in the analysis of the research. For example, recall in the first action research study, certain problems with the W + W framework were classified in a group as abstraction (A), mapping (M) or notational (N). Patterns such as these immediately focus the attention of the researcher and are often the pre-cursor of progress or more significant observations. What emerged from this observation was that typically some new ontological construct was being identified, ie it was an abstraction issue, but that further, there was no notational construct for this in the design world and neither therefore would there be a mapping for it. Pattern recognition and the subsequent identification of explanations for them are the stuff of qualitative research (Miles and Huberman 1994). That patterns were identified as a result of using the intellectual framework is an argument for using a framework, because such patterns were considered helpful conceptual adjuncts by the researcher, albeit that such patterns may be predisposed by the framework itself.

The third argument relates to the fact that in this research, issues that were actually outside the scope of the original W + W framework were able to be identified and even plotted on a diagram which was an extension to the original framework (figure 11.2) thus assisting progress towards a more comprehensive framework of ideas. The issues categorised under abstraction (A), process (P) and complexity (C) are all examples of this and indeed once identified much of the subsequent research focus centred round these categories.

The fourth argument is that the use of a framework forces the researcher to organise and prepare the research. Indeed, one of the motivations behind the Checkland (1985) paper was to encourage a more rational and organised approach to undertaking research. The very act of structuring a problem in terms of a framework of ideas (F), a methodology (M) and an area of application (A) abstracts the research in a certain manner and effectively forces the researcher to organise his or her research. Moreover, the identification of a framework of ideas as part of the intellectual framework forces the researcher to search for and synthesise intellectual artefacts into a consistent framework of ideas. Such preparation at the outset of research formalises the research process, focuses it and provides a means for evaluating it later.

The fifth argument concerns the review step which is incorporated into Checkland's intellectual framework. In a similar vein to the previous argument about organisation and preparation, the argument here is that Checkland's intellectual framework has an explicit step which requires that the researcher considers what has been learnt or achieved in the research. Of great practical value was the separation of the **undertaking** of the research activity from the **reviewing** of the research activity. Although a simple point, in the heat of undertaking research, there is clearly the temptation to move quickly on to the next phase without standing back and looking reflectively at what has been achieved. Without an explicit review step within a framework, a researcher might overlook or pay insufficient attention to reflection and learning and without a framework there would be no explicit review steps.

The sixth argument is that the experience of this research demonstrates that emergence of new ideas happens when using a framework anyway. During this research a number of important issues emerged. For example, the shift to focusing on business rules was not the original goal of the research. That change emerged as the research unfolded. Another example of emergent issue was the way that complexity kept recurring as a theme in the first action research study. So it can be concluded that a framework in itself is not necessarily an inhibitor of emergence, although the argument remains that it may influence the type of phenomenon that emerges.

13.8.3 Synthesis of a Revised Framework

Here a number of key issues arising out of the earlier analysis are identified and synthesised into a model which may be used for future research.

In any one situation, there is no way to prove that using a framework or not using a framework would have yielded better or different results. This is because it is impossible to return to a time to repeat a study where all the history of a situation, experiences and predispositions of participants and so on are exactly the same. There is no doubt however, that using a framework structures the research, ie ‘a priori’ it declares a set of ideas in certain way. Such ideas and the vocabulary associated with them, predefine to some degree what the research will seek, how it will seek it and how it will be evaluated. Framework-based research can therefore be said to be limited because it does not necessarily allow ideas to emerge freely out of the data. On the other hand without a framework, ideas can be argued to emerge more naturally. This discussion has similarity with debates over content analysis in which themes from the data are identified either prior to the analysis or emerge from the analysis (Glaser and Straus 1967).

However, research is a practical discipline which is carried out by humans who, though with the best intentions, still interpret the world and make sense of it based on their own outlook and experiences. A good example here concerns the decision to incorporate conditions as a diagrammatic construct into the diagram. Whereas it can be argued as 'logical' that conditions are a missing component in expressing completeness, both the analyst involved and myself were very familiar with flowcharts and flowchart-thinking. Though the concepts of conditions and flowcharts were not explicitly part of the intellectual framework, we cannot deny that they are not part of the repertoire of mental models we use **implicitly** (as opposed to their explicit declaration in a framework) to interpret and make sense of the world. So to what extent it can be said that such issues truly emerged from the research and to what extent that this phenomenon was more a matter of us recognising a situation or pattern at the backs of our minds for which we had a known solution is not totally clear. This question of the implicit models that any human researcher carries goes right to the crux of the debate here, because anyone arguing for performing research **without** a framework would have to demonstrate even more so in my view, how 'emergent' issues were **not** the product of previous influences however subtle or unintended!

In this research, the framework of ideas used initially may be considered neutral or context-free in that its role was to **manage** problem areas rather than to make **ontological** assumptions about problem areas. For example, the complexity metrics used allowed complexity factors to be calculated and thus permitted decisions to be taken on that basis. They did not in themselves predispose solutions. This was true also for Glasson's model and the extended W + W framework. So although such models frame the analysis and to that degree limit the decision space, it is the actual decision arising out of that analysis which may be biased in terms of favouring a certain solution. As indicated earlier such choices may well be based on the researcher or indeed other participants recognising patterns which are not explicitly part of the declared framework.

Figure 13.24 shows an extension of the original framework proposed by Checkland (1985). It indicates how I would use the framework in a future study. In other words it tries to reflect what has been learnt about using this framework.

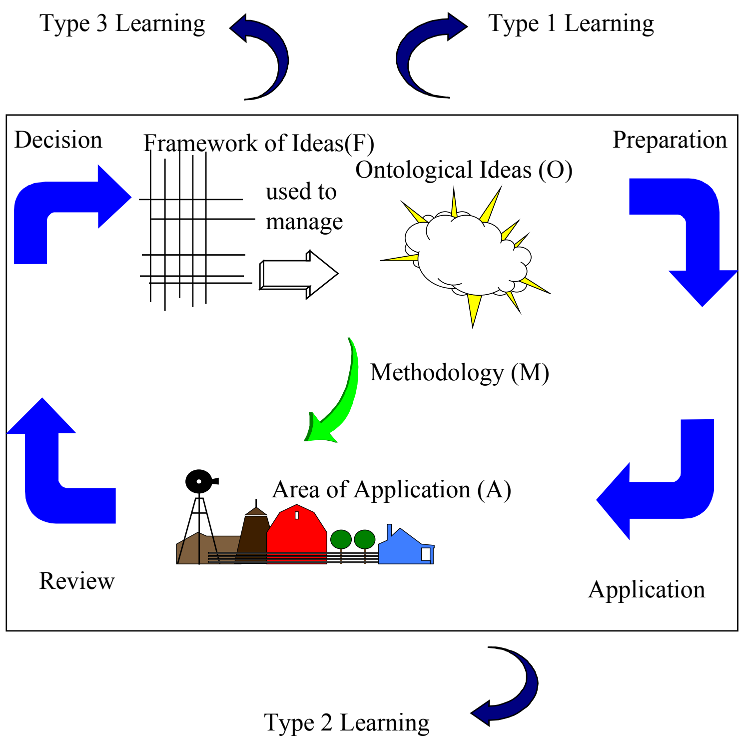


Figure 13.24 Synthesised Framework for Action Research Study

There are three aspects to figure 13.24. These are an extended intellectual framework, the learning cycle and the levels of learning. Figure 13.24 extends the original intellectual framework by distinguishing between those intellectual artefacts which manage the research ideas (F) and those ontological ideas (O) which are being tested. Separating them serves to highlight the different purposes they serve. It may be recalled that initially the ontological ideas were embedded in the methodology in the first action research study. The difficulty with this was that it was felt important to distinguish between the ontology and the methodology for testing the ontology. So, it was decided to relocate the ontology into the framework of ideas. However, here too there was a need to distinguish **ontologies** from the ideas which **manage ontological ideas,** hence the extension to Checkland's model in figure 13.24.

In both cases however, there may be a need to refer to outside theory or models. In the case of the ideas which manage the ontology, the theories and models have been those of Wand and Weber (1993), Glasson (1989), McCabe (1979) and Rossi (1997). These theories have been used to frame the research and to act as guides in decision-making. With regard to the set of ontological ideas, there also may be theories which can be utilised. This is discussed more fully in chapter 14 but for instance, the Harel blob and the complex object constructs were examples of where existing theory was able to be brought into the set of ontological ideas successfully. However, sometimes existing theory provided little or no support. For example, there were difficulties associated with modelling sub-states and the one-to-many business rule. In summary, it is suggested that for this type of research question, it is appropriate to deconstruct Checkland's model to distinguish between managing ontology, the ontology itself and the methodology for testing the ontology.

With regard to the learning cycle, the steps of preparation, application, review and decision have been used throughout this thesis and have been an integral part of the execution and reporting of this research. The preparation step was a most important step in this research because it was necessary to write down and formalise (in the form of an academic paper) an ontological world so that participants could be informed and be able to participate in the research activity. The act of writing a paper was in itself a test of some of the ontological ideas because these ideas required to be organised in a coherent and exemplified manner in order to communicate to others. In the application step, the methodology was applied to an area of application. Some of the ideas being tested came from existing theory and other were more along the lines of hypotheses being tested within the research itself. These ideas were evaluated (partly) by using templates of questions (appendix 1). However, the nature of action research is such that it is possible that further questions and issues emerge. This was quite evident in the second action research study in which a 'two diagram test' was developed to test a particular problem. In the review step answers to questions both on the original template and from the study itself were obtained, thus ensuring as full a review as possible. In addition, other sources of feedback such as the documents produced and field notes were also used in the analysis process. Finally in the decision step all decisions were formally recorded. A record of progress was important due to the sheer number of issues involved. These were also recorded in a research progress map showing a kind of 'audit trail' of the history of problems and decisions.

Three levels of learning were apparent in this study. These levels have some similarity with operational, tactical and strategic levels of activity in organisations. Level 1 type learning occurred at the end of each step (ie at the end of each preparation, application, review and decision step). The learning here was at the operational level in the sense it examined the issue directly in terms of the stated framework of ideas declared at that time. Level 2 type learning was formally documented at the end of the first action research study when the intellectual framework was reviewed and subsequently updated with additional ideas. Level 3 type learning was recorded at the end of the research project when the wisdom of using a framework was reviewed and the pros and cons for using or not using a framework was debated.

13.8.4 Conclusions from this Critique

The question of whether it is generally better to use a framework or not is a question which demands a much broader and deeper treatment than can be established from one research study. However, one strong recommendation from this critique is that at the beginning of any research, a researcher should at least explicitly consider whether a research framework should be used or not. The answer to that question has significant implications for the research but in practice it will depend on the research question, what previous work has been done in the problem area and the personal preference of the researcher.

Secondly, the research has demonstrated that significant benefits derive from following an organised approach which makes explicit how it will evaluate that research prior to the research. In other words, it is concluded that research which does use a framework is (or at least can be, as it is not in itself a guarantee) useful and productive.

Thirdly, I have used framework ideas which are neutral in the sense that they guide the research rather than predefine it and I have also separated managing ontological ideas being investigated from the framework of ideas. Such neutrality and separation diminish the potency of the argument which suggests that using a framework inherently biases the research. There is a learning curve involved with using a framework such as this one. It takes time to become familiar with ways of using it. The model presented is one way of working with it. It is not the only way, but it does represent the distillation of experience in working with it and reflecting upon it throughout this research.

13.9 Summary

In this chapter a review of the findings of this research has been presented. This was done in two parts. The first part of the review was partitioned into three areas. In the first area, the construction of the definition of a business rule was described, indicating how each component part of the definition was arrived at. The second area related to the policy rule constraint. This is a major constraint on the business rule in this research and it was felt that its discussion merited a full section. The third area of review looked at the justification for conclusions relating to the main deliverables of the BRD technique, namely the UBRD, the BRD and the EST. The arguments for the main technical and social conclusions with respect to each deliverable were detailed over three sections of this chapter.

In the second part of this chapter, a review of action research itself was undertaken. Action research has been a foundation of this research and in particular the use of an intellectual framework to guide this research was an integral part of the research process.

It was appropriate therefore to critically examine the use of this framework in order to fully identify and discuss this research.

In terms of specific findings, this chapter reported that:

- the definition of a business rule provided in chapter 2 represents an encapsulation of many aspects of business rules covered in this research,
- it is possible to develop a viable model of policy rules,
- there is a need for separate users diagrams,
- there is a need for a flexible approach to developing the UBRD,
- there is a need for guidance support for users,
- there is a need to validate business rules with users,
- major changes in the focus of the research were inclined to bring about a need for notational changes,
- there is justification for using the Harel blob for showing trigger and complex conditions,
- the research is inconclusive with regard to the use of sub-states,
- it is possible to construct a BRD from UBRDs,
- there is evidence to speculate about a third approach to developing information systems involving business rules as the primary medium,
- a necessary degree of complexity is inherent in modelling business rules, but it is possible to introduce tactics to manage complexity,
- there is evidence which points to the need for a repository which contains a complete specification of business rules (ie the EST),
- manyness should be shown in the EST,
- the research is inconclusive regarding if and where simultaneity should be shown,
- the experience of this research suggests it is important for a researcher to explicitly ask whether a research framework should be used or not,
- research which does use a framework can be useful and productive and

- framework ideas which are to be used to manage the research process should be separated from those ontological ideas being tested.

Overall, I consider I have developed a robust technique for modelling business rules, especially in terms of the components and context of the research question. However, as the chapter outlined there are a number of aspects which require further research. The next chapter explores this in more detail.

Chapter 14

Future Research

14.1 Overview

In this chapter future research with respect to the BRD (Business Rules Diagram) technique is considered. Throughout this research there have been two levels at which discussion has taken place. The first level has been 'direct' ontological discussion on topics such as business rules, simultaneity, overkill and so on. The second level dealt with matters related to 'research', the vocabulary of which included frameworks of ideas, methodology and the like. This chapter discusses future research at both levels.

Firstly, the model used in chapter 13 which provided headings for analysing findings is used as a basis for providing a set of headings to indicate future work on direct issues. The grey bubbles in figure 14.1 enumerate a number of topic areas for research. These are further cross-referenced in table 14.1. With regard to the pre- and post-BRD, it is suggested that there are issues or questions regarding how the BRD might be dovetailed with decisions taken before the BRD technique begins (ie the pre-BRD) and after the technique is completed (ie post-BRD). With regard to the BRD technique itself four further headings for discussion have been identified. Three of these are deliverables of the technique, namely the UBRD (User Business Rules Diagram), BRD and EST (Event Specification Table) and the fourth heading is validation of the technique.

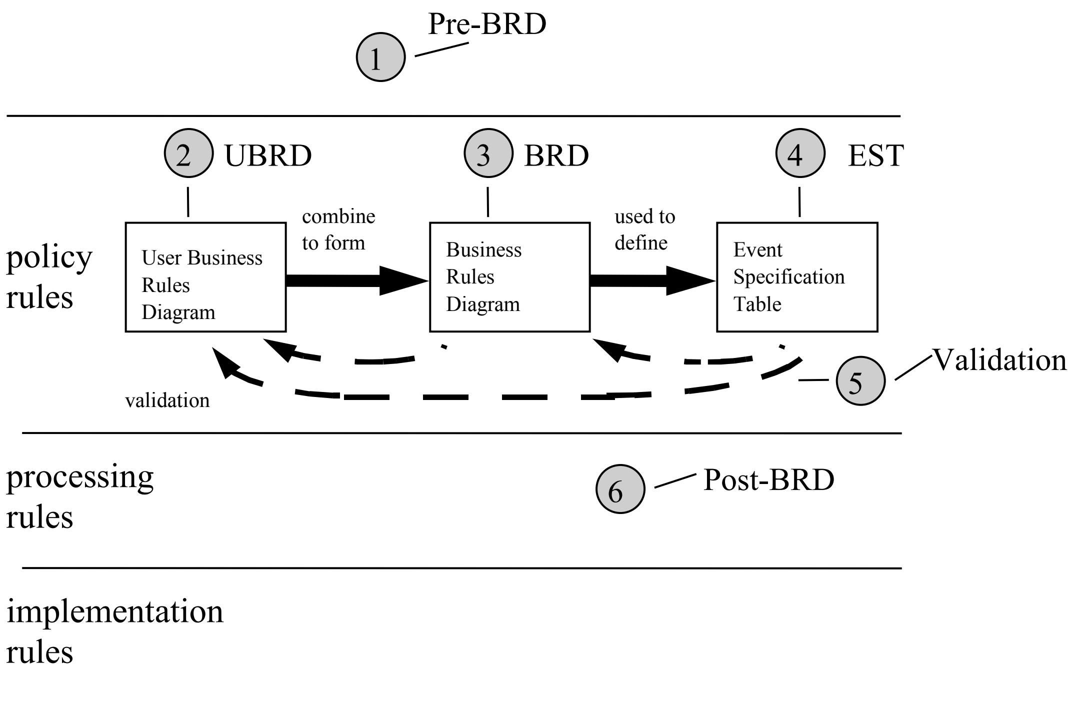


Figure 14.1 Headings for Further Research

After discussion on direct issues, there are a number of sections examining future research in the context of the intellectual framework used throughout this thesis. The model used for this is the core of the synthesised model taken from chapter 13 (figure 14.2). Again, grey bubbles are used to classify areas for future research, ie frameworks of ideas (7), ontologies (8), methodology (9) and areas of application (10). Table 14.1 is a summary of the remaining sections in this chapter.

Ref. no.	Section	Research Heading
1	14.2	Pre-BRD
2	14.3	UBRD
3	14.4	BRD
4	14.5	EST
5	14.6	Validation
6	14.7	Post-BRD
7	14.8	Framework of Ideas
8	14.9	Ontological Ideas
9	14.10	Methodology
10	14.11	Area of Application

Table 14.1 Summary of Sections in this Chapter

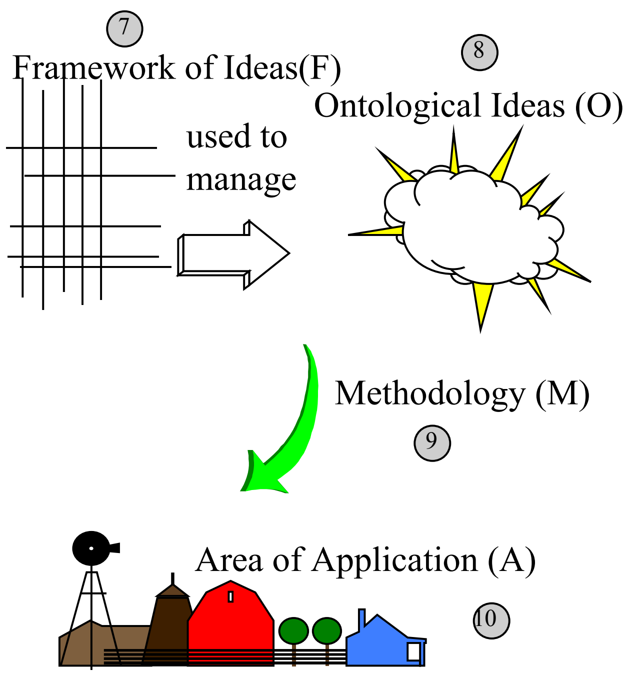


Figure 14.2 Components of Research Framework

14.2 Pre-BRD Issues

Clearly what happens before the BRD is used in an organisation has implications for how the BRD technique itself should be used. As far as this research was concerned, it was assumed that the participants had some prior knowledge of requirements and the BRD technique then took over. An interesting discovery in the second action research study was that users who have worked with a computerised system for some time tend not to distinguish between that which is policy and that which is processing or even implementation based. This is an example of the diversity that techniques such as the BRD have be able to deal with, if they are to be successful in organisations.

More generally, it may be speculated that an organisation may or may not have a formal method for identifying broad information requirements. If there is no formal method then it may be assumed that there is no formal statement of requirements and so the BRD technique must be flexible enough to generate requirements within itself or flexible enough to accommodate the incorporation of other recognised techniques for requirements elicitation. For example, the activities underpinning change analysis (Goldkuhl and Rostlinger 1993) can arguably be incorporated into the BRD. If there is a formal method for identifying requirements then it is important that a relatively smooth transition between that formal method and the BRD is enabled. Examples of formal methods or approaches for requirements elicitation include Cognitive Mapping (Montazemi and Conrath 1986), Scenarios (Mittermeir et al. 1987), SSM (Checkland and Scholes 1990), Viewpoint Modelling (Darke and Shanks 1997) and Event Flow Diagrams (Flynn and Davarpanah Jazi 1998).

It is suggested that pre-BRD research has to encompass the whole IRD process. Only then can the value of the BRD be completely assessed. The BRD as a technique exists within the context of IRD. Questions, issues and difficulties with IRD generally or with the earlier stages of IRD will manifest themselves often only at a later stage, ie quite possibly during the construction of the BRD. In other words, only if the scope of the research integrates earlier acquisition techniques with modelling techniques (ie the

BRD) will it be possible to establish the complete worth of the BRD. As an example of this consider stakeholder and viewpoint modelling about which there is currently active publication (Vidgen 1997, Darke and Shanks 1997). The models used by these authors provide insight into the process by which consensus is achieved when assertions are converted into requirements. However, these models, by nature of the acquisition techniques currently in use, deploy informal representations to specify requirements. But informal requirements still leave opportunities for ambiguity, omission and misinterpretation and so cannot be considered a satisfactory endpoint to IRD. As a second example, human and social aspects more generally are important here also. For example having volunteered a business rule, an individual has a political obligation to defend it amongst peers. Leifer et al. (1994) in discussing deep structures demonstrate that IRD is a complex and highly subjective business and so questions of how the BRD might fit into an IRD context require to be researched with a sensitivity to human and social issues (Newman and Robey 1992, Flynn and Davarpanah Jazi 1998) and thus demand a broader and more integrated treatment.

One further avenue of research involves looking at techniques which assist in the recording of design decisions. Such techniques need not be in direct competition with the BRD but rather complement its work. One example of such a technique is Design Rationale (Lourdios and Loucopolous 1996).

14.3 The User Business Rules Diagram

One of the main conclusions that emerged concerning the UBRD was the need to adopt a flexible approach with respect to the specific notation with which users may be comfortable. While at one level this may be acceptable as a conclusion, it is also clearly a candidate area for further research. It would be interesting to establish the boundaries of flexibility in terms of the extent to which constructs may be considered optional in the diagram. In the second action research study we saw the introduction of the idea of processing loops in the UBRD because the users were comfortable with this concept and perhaps thought in those terms. However, there are other ways in which the UBRD may be modified. For example, it would be interesting to establish whether state is a critical construct in the UBRD or whether by guidance the analyst could still ensure that events are 'true' events on the UBRD but avoid the display of states on the UBRD. It is not suggested that the role of state here would necessarily be diminished. Modelling with states remains fundamental to the construction of the BRD. For example, there is a lot of useful information obtained when users work with the analyst to obtain object life histories and in particular the cognitive cross-referencing that takes place in confirming relationships between states, events and signals is highly valuable. Nevertheless, questions of this type with respect to the UBRD are valid since they help to establish the limits of flexibility that might be acceptable with regard to the UBRD.

The degree of readability is also an issue which merits further research. Notwithstanding that there may be a variety of UBRDs, a necessary prerequisite for all variants is that users are able to read and understand a UBRD in order to confirm or deny its accuracy and to consider its implications. One way of answering this question is to conduct laboratory experiments requiring users to interpret and process UBRDs.

There are also opportunities to improve the quality of the process of constructing the UBRD. As an illustration, there may be opportunities for a more comprehensive discussion of business rules in the early stages of UBRD construction. Currently rules are simply brainstormed and relatively little attention is paid to how accurate they are, or whether there is overlap between business rules. Such discussion may cause these rules to be restated. Further, it may be possible to use some of the steps and activities of formal requirements elicitation techniques as discussed earlier to facilitate this. In summary it is possible that introducing and testing further steps into the UBRD construction process may improve the quality of the UBRD.

14.4 The Business Rules Diagram

In this section a number of specific issues concerning the BRD which remain unresolved are discussed.

14.4.1 Parallel states

Pragmatically it has been assumed that parallel states can be avoided on the BRD but may be required on the Object Life Histories. While it has been possible to present a case for avoiding the need to show sub-states on a BRD as a means of indicating parallelism, this is clearly an area for further research. Firstly, more research from a theoretical perspective is important, ie examining parallelism in human activity systems. Secondly, looking at sister disciplines such as real-time computing may also shed light on this issue. Thirdly, whatever is obtained from these two areas of investigation needs to be robust enough to be applicable to the majority of information systems applications and therefore more business applications need to be explored to understand more from a practical standpoint about where parallelism of this type may occur in information systems.

14.4.2 Sequentiality in a business rule

A question which emerged from the second action research study concerned the need to show sequential and non-sequential conditions on the BRD. This question needs further exploration. It may be that adding a graphical construct to show sequentiality/non-sequentiality is an over-simplification. Experience with the diagram so far, for example with the union and intersection problem, suggests that graphical tactics can work satisfactorily where the degree of complexity is low or moderate, but in highly complex scenarios graphical tactics can sometimes confuse rather than help. For example, suppose for the sequentiality problem, the tactics of connecting lines between conditions (as in a flowchart) was used to indicate sequentiality and that further no connecting lines meant non-sequentiality. For a moderately complex rule in which some conditions are sequential and others not, perhaps the tactic of a Harel blob or a separating line to partition conditions into sequential or non-sequential groups could be successful. However, in a highly complex business rule, this may break down visually. If so, perhaps once more the EST's role will be to depict information not on the BRD. More applied research is required to test these assertions.

14.4.3 Harel blob

The Harel blob has been used as an important graphical construct since the first action research study. Its role as an encapsulator has developed since that time and in chapter 13 it was concluded that its use in showing trigger selection and complex conditions justified its inclusion in the BRD. However, its potential can be explored further. In particular there may be other simpler ways of using it. For example, the Harel blob could be reduced to the basic logical operators of union (as in trigger selection), intersection (as was seen in complex conditions) and negation. I have added negation (ie the 'NOT' symbol) to deal with situations such as where all orders except faxed orders are acceptable. However, such assertions do need to be tested more formally. Further, the reservation discussed previously regarding the ability of graphical approaches to work well in highly complex situations applies here also.

14.4.4 Simultaneity

Simultaneity is an issue which has been present throughout this research. The most immediate priority with simultaneity is to establish whether it is absolutely necessary to be shown at the policy level or whether it can be delegated to a lower level of abstraction. This issue will be progressed only when it can be demonstrated (or not) that simultaneity can be modelled successfully at a lower level of abstraction and so the outcome of work described in the post-BRD section is relevant here. In summary, further research is required to establish the most appropriate place to model simultaneity. As with parallelism a combination of theoretical and practical tactics is suggested.

14.5 The Event Specification Table

When a major change is instituted there is often a settling down period afterwards in which minor alterations have to be made to the initial idea. This was found to be the case with a batch of notational changes made in the Business Event Diagram and also later when the focus of the research shifted to the business rule. It follows that as the role of the EST is consolidated to that of repository then some modifications may be required.

Two specific items need further work. The first is the logic of conditions. Currently, the notation only permits intersection and negation of conditions. It was speculated in earlier sections that complex conditions may require union and also the ability to show sequentiality and non-sequentiality. Such changes to the notation require appropriate testing.

The second item requiring further research is to do with the post-condition. In chapter 13 it was recommended that post-conditions be incorporated into the EST. A detailed notation for this still needs to be worked out and subsequent testing must follow.

14.6 Validation

The validation aspects of the technique are considered vital if it is to succeed as a tool which can be used in organisations (Flynn and Warhurst 1994). However, it was only possible in this research project to take the development of the technique to the point where testing the ideas on validation could begin. While the shift to a business rules emphasis and the subsequent development of the UBRD were valuable outcomes of the research, it was regrettably not possible to have tested the validation aspects of the technique. One immediate priority for further research is to conduct action research studies which involve the validation steps previously outlined.

Research into the structured walkthrough step is important because I contend that walkthroughs are crucial reflective activities which allow participants to review and consolidate what they have asserted and in doing so they will begin to improve the completeness and correctness of the specification.

Abnormal life analysis is a technique which will contribute to identifying unusual or infrequent activities and thus will also prove to influence completeness in a specification.

Amendment analysis is a technique which may or may not be argued to be relevant at the policy level. For instance, it may be claimed that a technique which considers the effects of amendments such as how change of customer status might affect the way orders for that customer are treated is a processing rule rather than a policy rule. On the other hand, this may involve substantial amounts of money and therefore be considered important enough as a core business rule, ie a policy rule.

These validation techniques provide the opportunity for corporate staff to be able to reason and speculate about aspects of business. This idea was first mooted in the original SDD (State Dependency Diagram) paper (McDermid 1990a) but was never validated. So in terms of establishing the value of validation techniques one useful yardstick and therefore a topic for future research would be the degree to which the BRD facilitates reasoning by its users.

One further category of research comes under the heading of validation. In figure 13.1, two feedback loops were identified. The first was that in the construction of the BRD, there needs to be a feedback to the set of UBRDs which are the basis of its genesis. Arguably, the structured walkthrough can achieve this validation. The second feedback loop is where questions arise in the EST. Here the feedback may involve checking back to the BRD or even further back to the UBRD or users. One suggestion by the systems analyst in the second action research study was to walkthrough rows of the EST with users. This suggestion illustrates that there may be a number of different ways of tackling such feedback tasks and therefore further research can help in this regard.

14.7 Post-BRD

One major limitation of the research conducted was that it did not consider what happens after the BRD is developed with respect to the remainder of the systems development life cycle. There are two main implications of this observation. The first is to examine how the specification of policy rules relates to lower levels of rules. This is particularly so with processing rules since this is seen as the next adjacent level of rule. A major research project is therefore to develop a notation for processing rules which would dovetail with the specification of policy rules. Indeed at a broader level the whole question of levels of business rules and possible alternatives to what have been proposed is a likely fruitful avenue for research.

The second implication is to do with the assertion that there is sufficient evidence to speculate that a third approach to systems development (termed 'business rule' based development) is justified. However, only when at least one complete project with accompanying methodology through to post-implementation review has been conducted will it be possible to assert with some authority that this contention is a realistic one. The processing rules model and fully implemented project which demonstrates the feasibility of 'business rules' based development are two immediate and important future directions of this research.

14.8 Framework of Ideas

The concept of using a framework of ideas to guide this research has been a cornerstone of the research approach. Though the idea of an intellectual framework is well established (Baskerville and Wood-Harper 1996b), little has been published in terms of sharing the experiences of working with a framework with other researchers. For example, in this research I found that it was helpful to separate ontological ideas from ideas for managing ontological ideas. It would be interesting to know whether other researchers using the intellectual framework reached similar conclusions or whether this conclusion might be limited to this type of research, ie developing diagrams.

Another question is to do with populating a framework. Initially, the framework consisted of only one idea, albeit an appropriate and important one. This idea was taken from the work of Wand and Weber (1993). As the research progressed, the framework was populated with additional ideas. These were taken from the work of Glasson (1989), McCabe (1976) and Rossi (1997). Clearly, with hindsight, it would have been better to have started with all four. This then raises questions such as how many ideas are reasonable to begin a research effort and is there some upper limit to the number of ideas that would be manageable in a single piece of research. Of course the scope of each idea is probably more important here than actual number of ideas. One can envisage that the research community may mature its thinking to the point where there may be an 'agreed' framework of ideas as a starting point for a particular type of research question. Then again, perhaps too much consensus and conformity might stifle creativity and flair. Nevertheless, discussion of this question and sharing of experiences amongst the research community can only be of benefit.

With hindsight, some aspects of cognitive science would be an example of other types of idea that the framework could have been started with and indeed the existing framework may profit from its addition. For example, the literature on cognitive psychology with respect to how we process concepts (eg Miller 1967) may well be a source for guidance or for understanding phenomena in future studies. It is clear that a thorough appreciation of how humans relate to a diagram at a cognitive (as well as social) level is an important aspect of this problem. So it might be possible to populate the framework of ideas with tools to manage or assess cognitive overload (Rich 1983, Winograd and Flores 1987, Mingers 1995, Bergel 1997, Kimble et al. 1998).

14.9 Ontological Ideas

In the literature review, a number of contributions by authors in the broad area of business rules were compared to the BRD. Clearly such published work is potentially a source of further ideas or even perhaps confirmation of some of the conclusions reached here. So the work of Herbst (1996), Flynn and Davarpanah Jazi (1998) and others needs to be continually reviewed and monitored as a potential source of ideas. Indeed, the prospect of some form of collaboration with other business rule researchers is an attractive proposition which offers the opportunity of consolidating related research endeavours.

Another way of searching for ontological ideas is to examine (or re-examine) the philosophies behind existing models and their constructs in the modelling community at large. For example, the work by Harel (1988) has also proved crucial in the development of the BRD since the concept of the Harel blob has been argued to be central in depicting triggers and conditions. As another example, the literature on flowcharting (eg Fitter and Green 1979) may be of further relevance to the UBRD. The work of such authors should be continually revisited as new ideas and problems arise in developing the BRD since they may provide clues to their solution.

A third approach to obtaining ontological ideas is by searching for other ontologies. For example, Guha and Lenat (1990) have an alternative approach to Wand and Weber. Returning to Wand and Weber's application of Bunge's work, they have extended their work by devising a representation model which amongst constructs others involves the state, event and law. These terms can be interpreted to correspond closely to the states, events and conditions of the BRD and it would be interesting to assess how well the BRD fits their representation model (Wand and Weber 1995). So, research should be continued by identifying and evaluating other ontologies or indeed extensions to existing ontologies which may have a bearing on the nature of business rules as used in this research.

Finally, as a goal, being able to construct a taxonomy of a business rule in terms of the different types of rules that may exist, how they might relate to each other in a classification hierarchy would be a valuable theoretical contribution and at this point probably represents a tangible endpoint for this research. It has to be acknowledged however, that in this respect the process of action research is bottom-up in the sense that though new types of rule may be discovered along the way it is impossible to state that all types of rules have been identified or classified. Nevertheless, as more is discovered about business rules from business applications it may be possible to infer classification schemes which better fit all the types of rules so far identified. This I consider a valuable form of theoretical retrospection on practical research.

14.10 Methodology

Action research has been used throughout this research as the vehicle for exploring the research question. Though a powerful and appropriate one, it is not the only vehicle and indeed there are many who argue for the benefits of methodological pluralism as a means of strengthening the potency of research generally (Wood-Harper 1985). Consideration should be given to combining action research with other approaches such as case study, laboratory experiment (as outlined earlier) into a comprehensive and possibly longitudinal programme of studies. Different data analysis techniques may also be interwoven within such studies. For example, Goguen and Linde (1992) present a review of different techniques appropriate for information requirements determination which could be used to understand aspects of business rules better. For example, they refer to protocol analysis, discourse and conversation analysis. These techniques have the potential to shed light on how users and analysts process their respective models. Results from such research would be an interesting dialectic to the results from semi-structured interviews of the same participants.

14.11 Area of Application

In this section, a liberal interpretation of the area of application is taken to allow discussion of the organisational and technical context in which the area of application may be seated.

After studying relatively few business applications it has been possible to present a case for the nature and componentry of a business rule which I contend is more comprehensive that many of its competitors. It would seem reasonable to assume that study of more business applications would continue to uncover more about business rules at least in the short term. In addition, the choice of further applications is important. For example, the assertion that some types of business rules may be application dependent (such as the auditing rules issue discussed previously) needs to be explored more formally. By choosing appropriate business applications, progress on this issue is more likely to be made.

A limitation of the research conducted was that the size of systems investigated would be considered small or moderate. Given the pervasiveness of the complexity issue, it may be reasonably speculated that some practical upper limit exists in terms of systems size above which the technique becomes unmanageable. Further work is required to explore this question.

Some consideration needs to be given to how business rules would be managed in an organisation. A consequence of the proposal regarding different levels of a business rule is that there may be different repositories or at least mechanisms for separating the different types of business rule. Further, there is also a need to record the connections or mappings between rules at different levels. The term repository management describes this type of need. Clearly, further research is required into repository management.

Another aspect of repository management is the use of Computer Assisted Software Engineering or CASE tools. There is clearly potential for automating and simplifying tasks in the context of the BRD technique. Three examples of many possibilities are now given. An obvious automation would be to provide a CASE tool facility which would consolidate UBRDs into a single BRD. A second example would be a facility which permits the configuration of different styles of UBRD. Once configured UBRDs could be checked for consistency and completeness. Thirdly, the ability to check the consistency between the BRD and EST and to highlight omissions etc., would be another way in which CASE tools could simplify or automate tasks and through this contribute to improving the quality of deliverables produced in a complex environment. In addition, introducing computerised assistance changes the work environment of the analyst and thus raises social questions. In terms of a direct contribution to reducing complexity itself, it would be possible for a CASE tool to provide different views of rules. For example, it could display rules showing states, events and signals but no conditions or any other permutation required. Such views may well suit the preferred style of an analyst or allow the analyst to focus on a particular problem.

In addition to technical aspects of business rule management, there are organisational questions also. The question of how business rules might be managed organisationally is important too. What would be the procedures by which business rules are agreed and modified? What emphasis and status should business rules have in an organisation? As confirmed in the second action research study, business rules are not typically husbanded in a formal way in organisations nor indeed are they typically recorded centrally. Yet there are obvious advantages to organisations that formalise their treatment of business rules. This would include the ability of organisations to react and adapt to changing circumstances (McDermid 1997). So further work is required in establishing the kinds of organisational infrastructure required to support business rules.

The work of Bickerton and Siddiqi (1992) sheds some light on the issue of information requirements determination in 'non-unitary' organisations. The direct questions that arises out of this work in relation to the BRD is whether the applicability of techniques such as the BRD may be limited to certain types of organisations, for example ones in which there is a high degree of unity and single purpose. In less cohesive organisations perhaps where there is also a great deal of uncertainty about many matters, it may be far more difficult to achieve consensus on business rules to the extent that the technique could become unworkable. In due course further exploration of such organisational types is warranted. On a broader plane, it is further suggested that the work of Jackson and Keys (1984) with their system of systems methodologies may be a useful framework for exploring this issue.

14.12 Conclusion

It is concluded that progress has been made in a discussion of what a business rule is, what its components are and how it may be constructed as well as how to go about designing new diagrams. However, there is much more that needs to be done in the area and table 14.2 is a summary of those questions raised in this chapter. In this light, the old adage of 'there are more questions than answers' takes on an immediate and challenging connotation.

Incorporate the principles of change analysis into the BRD
Ensure smooth transition between formal requirements methods and the BRD
Consider incorporating steps in order to achieve more consensus with requirements
Identify requirements processes with a sensitivity to cognitive and social issues
Investigate opportunities to record design decisions
Explore the boundaries of flexibility with respect to constructs used in the UBRD
Perform laboratory experiments exploring readability of the UBRD
Introduce steps to confirm business rules at an early stage of the process
Examine the need for parallel states
Devise a notation for sequentiality/non-sequentiality in a business rule
Amend and test the notation for the Harel blob
Decide on the role and place of simultaneity
Design notation for logic of conditions in the EST
Design notation for post-conditions in the EST
Test the value of the structured walkthrough as a validation technique
Test the value of abnormal life analysis as a validation technique
Decide on the role and place of amendment analysis
Test the ability of the BRD and UBRD to facilitate reasoning by users
Test the EST as a tool for walkthrough with users
Investigate how policy rules relate to lower levels of rules
Develop full specification/system using 'business rules' based development
Identify other yardsticks for 'measuring' complexity
Explore limitations of BRD with respect to size
Develop CASE tools to support BRD development
Create and test models for repository management
Explore the setting up of an organisational infrastructure to support business rules
Examine the impact of organisational types on the use of business rules
Identify other research approaches which may promote understanding of the BRD
Examine more applications using of the BRD
Resolve the question of application dependence
Work towards the goal of a taxonomy of business rules
Continue to examine other ontologies
Search for more framework ideas for evaluating the BRD
Monitor the work of and collaborate with other business rules researchers

Table 14.2 List of Research Topics outlined in this Chapter

Bibliography

Ackoff, R. (1971). 'Towards a system of system concepts'. *Management Science, 17*, pp661-71.

Appleton, D. (1988). 'Second Generation Applications'. *Database Programming and Design*(Feb), pp48-54.

Atknison, M., Banchilhon, F., DeWitt, D., Dittrich, K., Maier, D. and Zdonik, S. (1989). 'The object-oriented database system manifesto', *Proceedings of the First International Conference on Deductive and Object-Oriented Databases*. Kyoto, Japan.

Avison, D. E., and Fitzgerald, G. (1995). *Information Systems Development Methodologies*. (2nd ed.) Oxford: Blackwell Scientific Publications.

Baskerville, R., and Wood-Harper, T. (1992). 'A Critical Perspective on Action Research as a Method for Information Systems Research' (*Technical Report MCS-92-13*). New York: State University of New York.

Baskerville, R. and Wood-Harper, T. (1996a). 'A Taxonomy of Action Research Methods'. *Working Paper PAP012-0.05,* Institute for Informatics and Management Accounting, Copenhagen Business School.

Baskerville, R. L., and Wood-Harper, A. T. (1996b). 'A critical perspective on action research as a method for information systems research.' *Journal of Information Technology* 11, pp235-246.

Bergel, H. (1997). 'Cyberspace 2000: Dealing with Information Overload.' *Communications of the ACM* 40(2).

Baum, D. (1995). 'The Right Tools for Coding Business Rules'. *Datamation* (March), pp36-38.

Bickerton, M. J., and Siddiqi, J. (1992). 'The Classification of Requirements Engineering Methods', *Proceedings of the IEEE International Symposium on Requirements Engineering* (pp182-86). San Diego, California: IEEE Computer Society Press.

Bohl, M. (1971). *Flowcharting Techniques*. Chicago: Science Research Associates.

Boland, R. J. (1985). 'Phenomenology: A Preferred Approach to Research on Information Systems'. In E. Mumford, Hirschheim, R, Fitzgerald, G and Wood-Harper, T. (eds), *Research in Information Systems* (pp193-201): Elsevier Science, North-Holland.

Boland, R. J. (1990). 'Information system use as a hermeneutic process'. In H. Klein, and Nissen, H. K. (eds), *The Information Systems Research Arena of the 90s: Challenges, Perceptions and Alternative Approaches*. Copenhagen, Denmark.

Booch, G. (1991). *Object-oriented Design with Applications*. Redwood City, CA: Benjamin/Cummings.

Booch. G. (1994). *Object-oriented Analysis and Design*. (2nd ed.). Redwood, California: Benjamin/Cummings.

Bostrom, R. P., and Heinen, J. S. (1977). 'MIS problems and failures: a socio-technical perspective'. *MIS Quarterly, 1*(3), pp17-32.

Bostrom, R. P., and Thomas, R. D. (1983). 'Achieving excellence in communications: a key to developing complete, accurate, and shared requirements'. *Communications of the ACM.* 11, pp1-13.

Bunge, M. (1977). *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*. (vol. 3). Dordecht, Holland: Reidel.

Bunge, M. (1979). *Treatise on Basic Philosophy: Ontology 11: A World of Systems.* Boston, Massachusetts: Reidel.

Burns, R. (1990). *Introduction to Research Methods in Education*. Melbourne: Longman.

Byrd, T. A., Cossick, K. l., and Zmud, R. W. (1992). 'A synthesis of research on requirements analysis and knowledge acquisition techniques.' *MIS Quarterly* 16: pp117-138.

Calloway, L. J., and Ariav, G. (1990). 'Developing and using a qualitative methodology to study relationships among designers and tools'. In H. E. Nissen, Klein, H. K. and Hirschheim, R. (Ed.), *The Information Systems Research Arena of the 90's: Challenges, Perceptions and Alternative Approaches* (pp415-33). Copenhagen, Denmark: North-Holland.

Calloway, L. J. a. Ariav, G. (1995). 'Designing with dialogue charts: a qualitative content analysis of end-user designers' experiences with an SE design tool'. *Information Systems Journal* 5, pp75-103.

Chadzoglou, P. D. a. Macaulay, L. A. (1996). 'Requirements Capture and IS Methodologies'. *Information Systems Journal* 6, pp209-25.

Checkland, P. B. (1981). *Systems thinking, systems practice*. Chichester, England: John Wiley.

Checkland, P. (1985). ‘From Optimising to Learning: A Development of Systems Thinking for the 1990s’. *Journal of the Operational Research Society, 36*(9), pp757-67.

Checkland, P. (1991). ‘From Framework through Experience to Learning: the essential nature of Action Research’. *Information Systems Research: Contemporary approaches and emergent traditions*. (pp1-8) H. E. Nissen, Klein, H. K., and Hirschheim, R. Amsterdam, Elsevier.

Checkland, P., and Scholes, J. (1990). *Soft Systems Methodology in Action*. Chichester: John Wiley and Son.

Chen, P. (1976). ‘The Entity-Relationship Model - Towards a Unified View of Data’. *ACM Transactions on Database Systems, 1*(1), pp9-36.

Chikofsky, J. (1990). ‘The Database As a Business Road Map’. *Database Programming and Design*(May), pp62-7.

Chow, P. K. O., and Yeung, D. S. (1996). ‘Behavioural modelling in object-oriented methodology’. *Information and Software Technology, 38*, pp657-666.

Coad, P., and Yourdon, E. (1991). *Object-Oriented Analysis*. (2nd ed.). Englewood Cliffs, New Jersey: Yourdon Press.

Coleman, D., Hayes, F. and Bear, S. (1992). ‘Introducing Objectcharts or How to Use Statecharts in Object-oriented Design’. *IEEE Transactions on Software Engineering, 18*(1), pp9-18.

Coleman, D., Arnold, P., Bodoff, S., Dollin, C., Gilchrist, H., Hayes, F., Jeremaes, P. (1994). *Object-oriented Development: The Fusion Method*. Englewood Cliffs, New Jersey: Prentice-Hall.

Corbato, J. D. (1991). ‘On building systems that will fail’. *Communications of the ACM, 34*(9), pp72-81.

Daniels, A., and Yeats, D. (1971). *Basic Training in Systems Analysis*. London: Pitman International.

Darke, P., and Shanks, G. (1997). ‘User viewpoint modelling: understanding and representing user viewpoints during requirements definition.’ *Information Systems Journal* 7, pp213-239.

Davis, A. M. (1988). ‘A comparison of techniques for the specification of external behavior.’ *Communications of the ACM* 31, pp1098-1115.

Davis, G. B. (1982). ‘Strategies for information requirements determination.’ *IBM Systems Journal* 21, pp4-30.

Davis, W. S. (1983). *Systems Analysis and Design: A Structured Approach*. Reading, Massachusetts: Addison-Wesley.

De Marco, T. (1978). *Structured Analysis and System Specification*. Englewood Cliffs, New Jersey: Prentice-Hall.

Dick, B. (1992). *You want to do an action research thesis?* Chapel-Hill, Brisbane: Interchange Document.

Dos Santos, B. L. and Hawk, S. R. (1988). 'Differences in analyst's attitudes towards information systems development: evidence and implications'. *Information and Management, 14*, pp31-41.

Downs, E., Clare, P., and Coe, I. (1988). *Structured systems analysis and design method*. Hemel Hempstead, England: Prentice Hall.

Drucker, P. F. (1991). 'The new productivity challenge.' *Harvard Business Review* 69(6): pp69-79.

Dumdum, U. R., and Klein, H. K. (1986). 'The need for organizational requirements analysis'. In H. E. Sol and Nissen, G. (Ed.), *Quality of Work versus Quality of Information Systems* (pp393-420). Lund, Sweden.

Eden, C. (1988). 'Cognitive Mapping'. *European Journal of Operational research, 36*, pp1-13.

Eder, J., Kappel, G., Tjoa, A. M. and Wagner, R. R. (1987). 'BIER - The Behaviour Integrated Entity Relationship Approach'. In S. Spaccapietra (Ed.), *Entity-relationship Approach* (pp147-66): Elsevier Science (North- Holland).

Elden, M., and Chisholm, R. F. (1993). 'Emerging Varieties of Action Research: Introduction to the Special Issue'. *Human Relations, 46*(2), pp121-42.

Elliott, J. (1991). *Action research for educational change*. Milton Keynes, England, Open University Press.

Embley, D. (1992). *Object-oriented Systems Analysis: A Model-driven Approach*. Englewood Cliffs, New Jersey: Prentice-Hall.

Ewusi-Mensah, K., and Przasnyski, Z. H. (1994). 'Factors contributing to the abandonment of information systems development projects'. *Journal of Information Technology, 9*, pp185-201.

Farhoomand, A. F. (1987). 'An introduction to the theory and practice of action research in work organisations'. *Human Relations, 25*(6).

Fitter, M., and Green, T. R. G. (1979). ‘When do diagrams make good computer languages?’ *International journal of Man-Machine Studies, 11*, pp235-61.

Fitzgerald, G. (1991). ‘Validating new information systems techniques: a retrospective analysis’. In H. E. Nissen, Klein, H. K. and Hirschheim, R. (Ed.), *Information Systems Research: Contemporary Approaches and Emergent Traditions* (pp657-672). Amsterdam: North-Holland.

Flynn, D. J. (1992). *Information systems requirements: determination and analysis*. London: McGraw-Hill.

Flynn, D. J., and Davarpanah Jazi, M. (1998). ‘Constructing user requirements: a social process for a social context.’ *Information Systems Journal* 8: pp53-83.

Flynn, D. J., and Warhurst, R. (1994). ‘An empirical study of the validation process within requirements determination.’ *Information Systems Journal* 4: pp185-207.

Forrester, J. W. (1994). ‘Systems Dynamics, Systems Thinking and Soft OR’. *System Dynamics Review, 10*, pp245-256.

Galliers, R. D. (1990). ‘Choosing appropriate information systems research methodologies’. In H. E. Nissen, Klein, H. K. and Hirschheim, R. (Ed.), *The Information Systems Research Arena of the 90's: Challenges, Perceptions and Alternative Approaches* (pp155-73). Copenhagen, Denmark: North-Holland.

Galliers, R. D. (1992). *Information Systems Research: Issues, Methods and Practical Guidelines*. Oxford, Blackwell.

Galliers, R. D., and Land, F. F. (1987). ‘Choosing Appropriate Information Research Methodologies.’ *Communications of the ACM* (11): pp900-902.

Gane, C. P., and Sarson, T. (1979). *Structured Systems Analysis: Tools and Techniques*. New Jersey: Prentice-Hall.

Glaser, B. G., and Strauss, A. L. (1967). *The Discovery of Grounded Theory*. Chicago: Aldine.

Glasson, B. C. (1989). ‘Model of system evolution’. *Information and software technology, 31*(7), 351-6.

Goguen, J. A., and Linde, C. (1993). ‘Techniques for Requirements Elicitation’, *Proceedings of the IEEE International Symposium on Requirements Engineering* (pp152-164): IEEE Computer Society Press.

Goldkuhl, G., and Rostlinger, A. (1993). ‘Joint elicitation of problems: important aspects of change analysis’. In D. Avison, Kendall, J. E. and Degross, J. I. (Ed.), *Human,* *Organizational and Social Dimensions of Information systems Development* (pp107-25): Elsevier Science.

Graham, I. (1994). *Object-oriented Methods*: Addison-Wesley.

Greenspan, S., Mylopoulos, J., and Borgida, A., (1994). 'On Formal Requirements Modeling Languages: RML Revisited. *Proceedings of the 16th International Conference on Software Engineering.* (pp135-147) Los Alamitos, IEEE Press.

Greenwood, D. J., Whyte, W. F. and Harkavy, I. (1993). 'Participatory Action Research as a Process and as a Goal'. *Human Relations, 46*(2), pp175-92.

Gutierrez, O. (1987). 'Some aspects of information requirements analysis using a requirements grid technique'. *Information Analysis: Selected Readings* (pp347-62) R. D. Galliers (ed). Reading, Addison-Wesley**:**.

Guyu, R. V., and Lenat, D. B. (1990). 'Cyc: A midterm report.' *AI Magazine* 11, pp32-59.

Hamilton, S., and Ives, B. (1982). 'MIS Research Strategies'. *Information and Management* 5, 339-47.

Harel, D. (1987). 'Statecharts: A Visual Formalism for Complex Systems', *Science of Computer Programming* 8, pp231-274.

Harel, D. (1988). 'On Visual Formalisms'. *Communications of the ACM, 31*(5), pp514-30.

Henderson-Sellers, B., and Edwards, J. (1995). *Book Two of Object-oriented Knowledge*. Englewood Cliffs, New Jersey: Prentice-Hall.

Herbst, H. (1996). 'Business Rules in Systems Analysis: A Meta-model and Repository System'. *Information Systems, 21*(2), pp147-66.

Herbst, H., Knolmayer, G., Myrach, T., and Schlesinger, M. (1994). 'The specification of business rules: a comparison of selected methodologies. *Methods and Associated Tools for the Information Systems Life Cycle*'. (pp29-46) A. A. Verrijn-Stuart, and Olle, T. W. (eds), Elsevier Science, North Holland.

Hindle, T., Checkland, P., Mumford, M. and Worthington, D. (1995). 'Developing a Methodology for Multidisciplinary Action Research: A Case Study'. *Journal of the Operational Research Society, 46*, 453-64.

Hirschheim, R. A. (1986). 'Participative Systems Design: User Experiences, Evaluation and conclusions'. *The Australian Computer Journal, 18*(4), pp166-173.

Hult, M., and Lennung, S. A. (1980). 'Towards a definition of action research: a note and bibliography'. *Journal of Management Studies,* 17, pp241-50.

IEEE (1984*). Std-830*. New York, IEEE.

Israel, B. A., Schurman, S. J. and Hugentobler, M. K. (1992). 'Conducting Action Research: Relationships Between Organization Members and Researcher's. *Journal of Applied Behavioral Science, 28*(1), pp74-101.

Jackson, M. C., and Keys, P. (1984). 'Towards a System of System Methodologies'. *Journal of the Operational Research Society, 35*(6), pp473-86.

Jacobson, I., Christerson, M., Jonsson, P., and Overgaard, G. (1992). *Object-oriented Software Engineering: A Use Case Driven Approach*. Wokingham, England: Addison-Wesley.

Jones, B. (1991). Letter to editor. *Database Programming and Design*, Nov. p9.

Joosten, S. (1994). 'Trigger Modelling for Workflow Analysis', *Proceedings of the 9th Austria-Hungarian Informatics Conference* (pp234-47). Austria.

Kaiser, K. M. (1985). 'The relationship of cognitive style to the derivation of information requirements'. *Computer Personnel, 10*, pp2-12.

Kappel, G., and Schrefl, M. (1989). 'A Behavior Integrated Entity-Relationship Approach for the Design of Object-Oriented Databases'. In C. Batini (ed.), *Entity-Relationship approach* (pp311-28): Elsevier Science (North-Holland).

Ketterer, R. F., Price, H. P. and Polister, P. E. (1980). 'The Action Research Paradigm', *Evaluation and Action in the Social Environment* (pp1-15): Academic Press Inc.

Kim, J., and Courtney, J. F. (1988). 'A Survey of Knowledge Acquisition Techniques and their Relevance to Managerial Problem Domain.' *Decision Support Systems* 4(3): pp269-284.

Kimble, C., Grimshaw, D., and Hildreth, P. (1998). 'The Role of Contextual Clues in the Creation of Information Overload'. *Proceedings of the 3rd UKAIS Conference*, Lincoln, England, McGraw-Hill.

King, J. (1995). 'Sketchy plans, politics stall software development'. *Computerworld, 29*(25), p81.

Klein, H. K., and Hirschheim, R. (1991). 'Rationality Concepts in Information System Development Methodologies'. *Accounting, Management and Information Technology, 1*(2), pp157-87.

Klein, H. K., and Lyytinen, K. (1985). 'The Poverty of Scientism in Information Systems'. In E. Mumford, Hirschheim, R., Fitzgerald, G. and Wood-Harper, T. (eds.) *Research Methods in Information Systems.* (pp163-161) Elsevier Science (North-Holland).

Kolb, D. A., Rubin, I. M., and McIntyre, J. M. (1979). *Organizational Psychology: An Experiential Approach*. Englewood Cliffs: Prentice-Hall.

Kung, C. H., and Solvberg, A. (1986). 'Structural and Behavioural Modeling'. In T. Steel, and Meersman, R. (eds) *Database Semantics* (pp205-21). Elsevier Science (North-Holland).

Land, F. F. (1982). 'Adapting to Changing User Requirements'. *Information and Management, 5*.

Lau, F. (1997). 'A Review of the Use of Action Research in Information Systems Studies' in A. S. Lee, Liebenau, J. and DeGross, J. I. (eds*) Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research*, (pp31-68). London: Chapman and Hall.

Lausen, G. (1988). 'Modeling and Analysis of the Behaviour of Information Systems'. *IEEE Transactions on Software Engineering, 14*(11) pp1610-1620.

Leifer, R., Lee, S. and Durgee, J. (1994). 'Deep structures: real information requirements determination'. *Information and Management, 27*, pp275-85.

Loosley, C. (1992). 'Separation and Integration in the Zachman Framework'. *Data Base Newsletter, 20*(1).

Louridas, P., and Loucopoulos, P. (1996). 'A Framework for Evaluating Design Rationale Methods'. In K. Siau, and Wand, Y. (eds.), *Proceedings of the Workshop on Evaluation of Modeling Methods in Systems Analysis and Design* (pp1-16). Crete, Greece.

LSDM. (1986). *Structured Systems Analysis (part 1) and Structured Systems Design (part 2)*. Oxford St., London: Learmonth and Burchett Management Systems Ltd.

Lucas, M. (1993). 'Getting down to business.' *Informatic*s (September): pp41-45.

Lundeberg, M. (1982). 'The ISAC Approach to Specification of Information Systems and its Application to the Organisation of an IFIP Working Conference'. In T. W. Olle, Sol, H. G. and Virrijn-Stewart, A. A. (eds), *Information Systems Design Methodologies: A Comparative Review* (pp173-234): North-Holland.

Lyytinen, K., and Hirschheim, R. (1987). 'Information systems failures - a survey and classification of the empirical literature.' *Oxford Surveys in Information Technology*, pp257-309.

Macaulay, L. (1993). 'Requirements Capture as a Cooperative Activity', *Proceedings of the IEEE International Symposium on Requirements Engineering* (pp174-81) San Diego, California: Computer Society Press.

Mansell, G. (1991). 'Action research in information systems development'. *Journal of Information Systems, 1*, pp29-40.

Martin, J., and Finkelstein, C. (1981). *Information Engineering vols 1 and 2*. New Jersey: Prentice Hall.

Martin, J., and Odell, J. (1995). *Object-oriented Methods: A Foundation*. Englewood Cliffs, New Jersey: Prentice-Hall.

Mason, R., and Mitroff, I. (1981). *Challenging Strategic Planning Assumptions*. New York, Wiley.

McCabe, T. J. (1976). 'A Complexity Measure'. *IEEE Transactions in Software Engineering, 2*(4), pp308-20.

McDermid, D. C. (1990a). 'The State Dependency Diagram'. *The Software Engineering Journal*(May), pp165-73.

McDermid, D. C. (1990b). *Software Engineering for Information Systems*. Oxford, England: Blackwell Scientific Publications.

McDermid, D. C. (1997). 'The Need for Business Rules: A Contemporary Issue in Business'. *Journal of Contemporary Issues in Business and Government*. 3(1), pp63-74.

McKay, J., Newby, M., and Marshall, P. (1995). 'The relationship between information requirements determination and software maintenance' (vol. 2, pp567-78). *Proceedings of the 6th Australasian Conference on Information Systems*: Perth, Australia.

McKernan, J. (1991). *Curriculum Action Research: A Handbook of Methods and Resources for the Reflective Practitioner*. London: Kogan Page Ltd.

Miles, M. B., and Huberman, A. M. (1994). *Qualitative Data Analysis*. Thousand Oaks, California: Sage.

Miller, G. A. (1967). *The Psychology of Communication*. New York: Basic Books.

Miller, J. C. (1964). 'Conceptual Models for Determining Information Requirements.' *Proceedings of the Spring Joint Computer Conference* 25, pp609-20.

Mingers, J. (1995). 'Information and meaning: foundation for an intersubjective account.' *Information Systems Journal* 5, pp285-306.

Mittermeir, R., T., Hsia, P., and Yeh, R. T. (1987). 'Alternatives to overcome the communication problem of formal requirements analysis'. *Information Analysis: Selected Readings.* (pp153-65). R. D. Galliers (ed). Reading, Addison-Wesley.

Montazemi, A. R., and Conrath, D. W. (1986). 'The Use of Cognitive Mapping for Information Requirements Analysis'. *MIS Quarterly, 10*(1), pp45-56.

Moody, D. L., and Shanks, G. G. (1998). 'What makes a good data model? A framework for evaluating and improving the quality of entity relationship models'. *Australian Computer Journal*, 30(3), pp97-110.

Moriarty, T. (1993a). 'Business Rules Analysis'. *Database Programming and Design* (April), pp66-69.

Moriarty, T. (1993b). 'Losing the Business'. *Database Programming and Design* (June), pp66-69.

Moriarty, T. (1993c). 'The Next Paradigm'. *Database Programming and Design* (Feb), pp66-68.

Moriarty, T. (1993d). 'Where's the Business?' *Database Programming and Design* (May), pp68-69.

Mumford, E. (1985). 'Defining systems requirements to meet business needs: a case study example'. *Computer Journal*(28), pp97-104.

Myers, M. D. (1994). 'Quality in Qualitative Research in Information Systems', *Proceedings of the Fifth Australasian Conference on Information Systems*. Melbourne, Australia.

Newman, M., and Robey, D. (1992). 'A Social Process Model of User-Analyst Relationships.' *MIS Quarterly*: (June) pp249-265.

Olle, T. W., Hagelstein, J., Macdonald, I. G., Rolland, C., Van Asche, F. J. M., Verrijn-Stuart, A. A. (1991). *Information Systems Methodologies: A Framework for Understanding*. Wokingham, England: Addison-Wesley.

Orlikowski, W. J., and Baroudi, J. J. (1991). 'Studying Information Technology in Organizations: Research Approaches and Assumptions'. *Information Systems Research, 2*(1), pp1-28.

Parnas, D. (1972). 'On the criteria to be used in decomposing systems into modules.' *Communications of the ACM* 15(2): pp1053-58.

Parsons, J., and Wand, Y. (1991). 'The objects paradigm - two for the price of one'. *Proceedings of the Workshop on Information Technologies and Systems*, Boston, Massachusetts.

Peterson, J. L. (1981). *Petri Net Theory and the Modelling of Systems*. Englewood Cliffs, New Jersey: Prentice-Hall.

Peterson, J. L. (1977). ‘Petri Nets’. *Computing Surveys, 9*(3), pp223-252.

Pressman, R. (1987). *Software Engineering: A Practitioner's Guide*: McGraw-Hill, New York.

Pettigrew, A. M. (1985). ‘Contextualist Research: A Natural Way to Link Theory and Practice’. In E. E. Lawler, Mohrman, A. M., Mohrman, S., Ledford, S. E. and Cummings, T. G. (eds) *Doing Research that is Useful for Theory and Practice*. (pp222-274). San Franscisco: Jossey-Boss.

Pohl, K. (1994). ‘The Three Dimensions of Requirements Engineering: A Framework and Its Applications.’ *Information Systems* 19, pp243-258.

Quang, P. T. (1991). *MERISE in Practice*. Houndsmills, England: Macmillan.

Rich, E. (1983). *Artificial Intelligence*. Singapore, McGraw-Hill.

Rockart, J. F. (1979). ‘Chief Executives define their own data needs*.’ Harvard Business Review* 57, pp81-91.

Rosenquist, C. (1982). ‘Entity Life Cycle Models and their Applicability to Information Systems Development Life Cycles’. *Computer Journal, 25*(3) pp307-315.

Rossi, M. (1997). ‘Evolution of OO Methods: the Unified Case’. *Australian Journal of Information Systems, 4*(2), pp55-62.

Rossi, M., and Brinkkemper, J. (1995). ‘Metrics in Methods Engineering’. *Proceedings of the 7th Advanced Information Systems Engineering Conference*, Oulu, Finland, Springer-Verlag, Berlin.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorenson, W. (1991). *Object-oriented Modeling and Design*. Englewood Cliffs, New Jersey: Prentice-Hall.

Ryan, K. (1995). ‘Let's have more experimentation in requirements engineering’. *Proceedings of the Second IEEE International Symposium on Requirements* Engineering, York, UK, IEEE.

Sandifer, A. (1991). Access Path. *Database Programming and Design* (Nov), pp9-10.

Sandifer, A., and Von Halle, B. (1991a). ‘Designing by the Rules’. *Database Programming and Design* (Jan), pp11-4.

Sandifer, A., and Von Halle, B. (1991b).'A Rule by Any Other Name'. *Database Programming and Design* (Feb), pp11-13.

Sandy, G. (1996). 'The Nature and Structuring of Organisational Rules'. Proceedings of the 7th Australasian Conference on Information Systems, Hobart, Australia.

Sandy, G. A. (1994). 'Nice Models But Where's the Business'. *Proceedings of the 5th Australian Conference on Information Systems*, Melbourne, Australia.

Schein, E. (1969). *Process Consultation: Its Role in organisational Development*. Reading, Mass.: Addison-Wesley.

Shah, H. U., Dingley, S. and Golder, P. A. (1994). 'Bridging the culture gap between users and developers'. *Journal of Systems Management, 45*(July), pp18-21.

Shand, R. M. (1994). 'User manuals as project management tools: part I - theoretical background'. *IEEE Transactions on Professional Communication, 37*(2), pp75-80.

Shanks, G., Rouse, A., and Arnott, D. (1993). 'A Review of Approaches to Research and Scholarship in Information Systems', *Proceedings of the 4th Australian Conference in Information Systems* (pp29-44). Brisbane, Australia.

Shlaer, S., and Mellor, S. J. (1992). *Object Lifecycles: Modeling the World in States*. Englewood Cliffs, New Jersey: Yourdon Press.

Shneiderman, B., Mayer, R., McKay, D. and Heller, P. (1977). 'Experimental investigations of the utility of detail flowcharts in programming'. *Communications of the ACM, 20*, pp373-387.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading, Massachusetts.: Addison-Wesley.

Susman, G. I. (1983). 'Action Research: A Sociotechnical Perspective'. *Beyond Method: Strategies for Social Research*, Beverly Hills.

Susman, G. I. and Evered, R. (1978). 'An assessment of the scientific merits of action research.' *Administrative Science Quarterly* 25, pp582-603.

Swepson, P., and Dick, B. (1993). 'Action research: too risky for theses?*' Social Psychologists Meeting*, Newcastle, Australia.

Tan, M. (1993). 'Using communication theory for systems design: a model for eliciting information requirements'. In D. Avison, Kendall, J. E., and DeGross, J. I. (Ed.), *Human, Organizational and Social Dimensions of Information systems Development* (pp241-262): Elsevier Science.

Trist, E., and Bamforth, K. (1951). 'Some special and psychological consequences of the Longman method of goal getting.' *Human Relations* 4, pp3-38.

Tsalgatidou, A., and Loucopoulos, P. (1991a). 'Rule-base behaviour modelling: specification and validation of information systems dynamics'. *Information and Software Technology, 33*(6), pp425-31.

Tsalgatidou, A., and Loucopoulos, P. (1991b). 'An Object-oriented Rule-based Approach to the Dynamic Modelling of Information Systems'. In H. G. Sol and Van Hee, K. M. (eds), *Dynamic Modelling of Information Systems* (pp165-88): Elsevier Science (North Holland).

Tsaligatidou, A., Karakostas, V., and Loucopoulos, P. (1990). 'Rule-Based Requirements Specification and Validation'. In B. Steinholz, and Solvberg, A. (eds), *Proceedings of the 2nd Nordic CAISE Conference*. Berlin: Springer.

UML. (1997). *Unified Modeling Language*: www.rational.com/uml. Internet publication.

Valusek, J. R., and Fryback, D. G. (1987). 'Information requirements determination: obstacles within, among and between participants'. *Information Analysis: Selected Readings* (pp139-51). R. D. Galliers (ed). Reading, Addison-Wesley**:**

Van Horn, R. L. (1973). 'Empirical Study of Management Information Systems'. *Database, 5*, 172-82.

Vessey, I., and Conger, S. (1993). 'Learning to Specify Information Requirements: The Relationship between Application and Methodology'. *Journal of Management Information Systems, 10*(2), pp177-201.

Vidgen, R. (1997). 'Stakeholders, soft systems and technology: separation and mediation in the analysis of information systems requirements.' *Information Systems Journal* 7, pp21-46.

Vitalari, N. P. (1992). 'Structuring the Requirements Analysis Process for Information systems: A Proposition Viewpoint'. In W. W. Cotterman, and Senn J. A. (eds), *Challenges and Strategies for Research in Systems Development* (pp163-79): John Wiley, Chichester, England.

Vogel, D. R., and Wetherbe, J. G. (1984). 'MIS Research: A Profile of Leading Journals and Universities'. *Database, 16*(3), 3-14.

Von Halle, B. (1994). 'Making business rules real'. *Informatics* (Jan), pp48-9.

Walsham, G. (1995). 'The Emergence of Interpretivism in IS Research'. *Information Systems Research, 6*(4), pp376-394.

Wand, Y., and Weber, R. (1993). 'On the Ontological Expressiveness of Information Systems Analysis and Design Grammars'. *Journal of Information Systems, 3*, pp217-37.

Wand, Y., and Weber, R. (1995). 'On the deep structure of information systems.' *Information Systems Journal* 5, pp203-223.

Watson, A. H., and McCabe, T. J. (1996). *Structured Testing: A Testing Methodology using the Cyclomatic Complexity Metric*. Gaithersburg, USA: National Institute of Standards and Technology.

Weber, R. a. Zhang, Y. (1996). 'An analytical evaluation of NIAM's grammar for conceptual schema diagrams.' *Information Systems Journal* 6, pp147-70.

Whetton, D. A. (1989). 'What Constitutes a Theoretical Contribution?' *Academy of Management Review, 14*(4), pp490-5.

Whyte, W. F. (1991). *Participatory action research*. Newbury Park: Sage.

Winograd, T., and Flores, F. (1987). *Understanding Computers and Cognition*. Reading, Massachusetts, Addison-Wesley.

Wood-Harper, A. T., Antill, L. and Avison, D. E. (1985). *Information Systems Definition : The Multiview Approach*. Oxford: Blackwell Scientific Publications.

Wood-Harper, T. (1985). 'Research Methods in Information Systems: Using Action Research'. In E. Mumford, Hirschheim, R., Fitzgerald, G. and Wood-Harper, T. (eds) *Research Methods in Information Systems* (pp169-191. Elsevier Science (North Holland).

Yourdon, E. (1994). *Object-oriented Systems Design: An Integrated Approach*. Englewood Cliffs, New Jersey: Prentice-Hall.

Zmud, R. W., Anthony, W. P., and Stair, R. (1992). The use of mental imagery as a requirements technique*. Challenges and Strategies for Research in Systems Development.* (pp337-50) W. W. Cotterman, and Senn, J. A. Chichester, England, John Wiley and Son.