A practical guide to

REAL-TIME SYSTEMS DEVELOPMENT

Sylvia

GOLDSMITH

Contents

Preface *page* xiii

CHAPTER ONE Introduction 1

1.1 Traditional problems in real-time development 1

1.1.1 Communication 1

1.1.2 System complexity 4

1.1.3 Different development techniques 5

1.2 The ideas behind structured development 6

1.2.1 The use of models 6

1.2.2 The diagrams 10

1.2.3 The rules 10

1.2.4 The development steps 11

1.3 Example of structured development 14

1.3.1 Mushroom-picking system: statement of purpose 14

1.3.2 Mushroom-picking system: essential environmental model 14

1.3.3 Mushroom-picking system: essential behavioural model 16

1.3.4 The processor environment model 19

1.3.5 The software environment model 22

1.3.6 The code organization model 24

1.4 Maintenance 25

PART ONE Development tools 27

CHAPTER TWO Data flow diagrams 29

2.1 Introduction 29

2.2 Data flow diagram components 29

2.2.1 Basic notation 29

2.2.2 Flows 31

2.2.3 Transformations 33

2.2.4 Stores 36

2.2.5 Terminators 38

2.3 Building a data flow diagram 39
2.3.1 Connectivity 39
2.3.2 The jigsaw approach 41
2.3.3 General guidelines 42
2.4 Checking a data flow diagram 43
2.4.1 Checking for syntactic correctness 43
2.4.2 Checking for coherence 44
2.5 Summary 49
2.6 Exercises 49

CHAPTER THREE State transition diagrams 51
3.1 Introduction 51
3.2 State transition diagram components 53
3.2.1 Basic notation 53
3.2.2 States 55
3.2.3 Transitions 60
3.2.4 Event stores 69
3.3 Building a state transition diagram 71
3.3.1 The DFD control process 71
3.3.2 System sequence 74
3.3.3 General guidelines 77
3.4 Checking a state transition diagram 78
3.4.1 Checking for syntactic correctness 78
3.4.2 Checking for coherence 79
3.5 Summary 88
3.6 Exercises 89

CHAPTER FOUR Entity relationship diagrams 95
4.1 Introduction 95
4.2 Entity relationship diagram components 97
4.2.1 Basic notation 97
4.2.2 Objects 100
4.2.3 The relationship 106
4.3 Building an entity relationship diagram 109
4.3.1 Building the ERD from narrative description 110
4.3.2 Building the ERD from the DFD 112
4.4 Checking an entity relationship diagram 114
4.4.1 Checking for syntactic correctness 115
4.4.2 Checking for coherence 116
4.5 Summary 117
4.6 Exercises 117

CHAPTER FIVE Supporting textual specifications 119
5.1 Introduction 119
5.2 The project dictionary 121

5.3 Data composition specification language 122
5.3.1 Data composition 123
5.3.2 Multiple data elements 124
5.3.3 Choice of data elements 124
5.3.4 Optional data elements 124
5.3.5 Key data elements 125
5.4 Textual support for data flow diagrams 125
5.4.1 Data flows and data stores 125
5.4.2 Data transformations 126
5.4.3 Terminators 127
5.4.4 Event flows and event stores 127
5.4.5 Control processes 127
5.4.6 Example project dictionary for teabag boxing system DFD 129
5.5 Textual support for state transition diagrams 134
5.5.1 States 134
5.5.2 Transitions 135
5.5.3 Conditions 135
5.5.4 Actions 135
5.6 Textual support for entity relationship diagrams 136
5.6.1 Objects 137
5.6.2 Relationships 137
5.6.3 Example project dictionary for teabag boxing system ERD 137
5.7 Summary of project dictionary requirements 138
5.8 Exercises 139

CHAPTER SIX Diagram integration 140
6.1 Introduction 140
6.2 Levelling 140
6.2.1 Levelling on control 141
6.2.2 Levelling on data 142
6.2.3 Object oriented levelling 145
6.3 Horizontal balancing 146
6.3.1 Balancing the DFD against the STD 146
6.3.2 Balancing the DFD against the ERD 150
6.4 Vertical balancing 153
6.5 Summary 155
6.6 Exercises 156

PART TWO Analysis 163

CHAPTER SEVEN The context diagram 165
7.1 Introduction 165

7.2 Context diagram components 167
7.2.1 Basic notation 167
7.2.2 Context diagram notation rules 167
7.3 Building the context diagram 169
7.3.1 Investigating the system requirements 169
7.3.2 Identification of terminators 170
7.3.3 Identification of interfaces 175
7.3.4 Starting the project dictionary 176
7.4 Checking the context diagram 176
7.4.1 Checking for syntactic correctness 177
7.4.2 Checking for coherence 177
7.4.3 Checking for completeness 181
7.5 Context diagrams for large and complex systems 182
7.6 Summary 188
7.7 Exercises 188

CHAPTER EIGHT The event list 191
8.1 Introduction 191
8.2 Event list components 193
8.2.1 The event 193
8.2.2 The response 202
8.2.3 Classification 205
8.3 Building the event list 209
8.3.1 Building the event list from the context diagram 210
8.3.2 Building the event list from event scenarios 213
8.3.3 General guidelines 214
8.4 Checking the event list 215
8.4.1 Naming in the event list 215
8.4.2 Partitioning the event list 216
8.5 Summary 218
8.6 Exercises 218

CHAPTER NINE The behavioural model 221
9.1 Introduction 221
9.2 Strategy for building the behavioural model 221
9.3 Identifying the system footprint 223
9.4 Generating the preliminary models 225
9.4.1 Generating the data flow diagram 225
9.4.2 Generating the state transition diagram 231
9.4.3 Generating the entity relationship diagram 235
9.5 Levelling and balancing the model 238
9.5.1 Levelling the preliminary models 238
9.5.2 Balancing the preliminary models 239
9.6 Lower level model details 239
9.6.1 Lower level models 239

9.6.2 The project dictionary 240
9.6.3 Implementation constraints 241
9.6.4 New versus reimplemented systems 242
9.7 Summary 243
9.8 Exercises 244

PART THREE Design 247

CHAPTER TEN Processor environment modelling 253
10.1 Introduction 253
10.1.1 Customer involvement in processor environment modelling 254
10.1.2 Commitment to software development 255
10.2 Processor evaluation 255
10.2.1 Estimation of processing requirements 256
10.2.2 Evaluation of real-world processors 258
10.3 Processor allocation 259
10.3.1 Processor interfaces 259
10.3.2 Allocation of data transformations 262
10.3.3 Allocation of stored data 263
10.3.4 Allocation of control processes/STDs 265
10.3.5 Construction of complete processor environment model 268
10.4 Traceability tables 268
10.5 Summary 270
10.6 Exercises 272

CHAPTER ELEVEN Human–computer interface modelling 276
11.1 Introduction 276
11.1.1 The customer's view of the HCI 276
11.1.2 The developer's view of the HCI 277
11.2 Strategies for modelling the HCI 278
11.2.1 Top down functional decomposition 279
11.2.2 Event partitioning 280
11.2.3 Menu structures 283
11.3 Summary 286
11.4 Exercises 289

CHAPTER TWELVE Software environment modelling 292
12.1 Introduction 292
12.2 Software architecture evaluation 294
12.3 Removal of 'perfect technology' 297
12.3.1 Process activation 298
12.3.2 Continuous processing and data 299
12.3.3 Discrete processing and data 300
12.3.4 Concurrent behaviour 304
12.3.5 Data security 306

12.4 Software allocation 307
12.4.1 Execution unit interfaces 308
12.4.2 Allocation to software architecture 308
12.4.3 Removal of common processing 310
12.4.4 Allocation of processing to execution units 311
12.4.5 Allocation of data to data storage units 314
12.4.6 Construction of the SEM 316
12.5 Summary 323
12.6 Exercises 324

CHAPTER THIRTEEN Structure charts 332
13.1 Introduction 332
13.2 Structure chart components 332
13.2.1 Basic notation 333
13.2.2 Modules 334
13.2.3 Module calls 334
13.2.4 Data 336
13.2.5 Further notation 337
13.3 Supporting specifications 343
13.3.1 Data specifications 344
13.3.2 Module specifications 344
13.4 Building a structure chart 345
13.5 Checking a structure chart 347
13.5.1 Checking for syntactic correctness 347
13.5.2 Checking for clarity 348
13.5.3 Checking the implementation 353
13.6 Summary 354
13.7 Exercises 354

CHAPTER FOURTEEN Code organization modelling 358
14.1 Introduction 358
14.2 Translation of execution unit into preliminary structure chart 359
14.2.1 Translation of STD into preliminary structure chart 359
14.2.2 Translation of DFD into preliminary structure chart 368
14.2.3 Translating execution unit hierarchy into preliminary structure chart 373
14.3 Structure chart refinement 374
14.4 Summary 379
14.5 Exercises 381

PART FOUR Summary 385

CHAPTER FIFTEEN Summary 387
15.1 Introduction 387

15.2 Project management and auditing 388
15.2.1 General points 388
15.2.2 Milestones 388
15.2.3 What to watch for in reviews 389
15.2.4 Testing 389
15.3 CASE tools 390
15.3.1 Graphics 391
15.3.2 Non-graphics 391
15.3.3 Checking 392
15.3.4 Human–computer interface 392
15.3.5 Support of real-time design 392
15.3.6 Configuration management 392
15.3.7 Practical considerations 393
15.4 Use of structured development with other techniques 393
15.4.1 Object oriented development 393
15.4.2 Formal techniques 395

APPENDIXES 397

A Exercise answers 399
B Connections within a data flow diagram 471
C Step-by-step guide to analysis 475
D Step-by-step guide to processor allocation 478
E Step-by-step guide to software environment modelling 483
F Step-by-step guide to code organization modelling 489

Glossary *494*
Bibliography *499*
Index *500*