

Contents

Preface *xi*

1 THE NEED FOR BETTER TOOLS 1

- 1.1 What Goes Wrong in Analysis? 2
- 1.2 How Much Can We Blame Our Tools? 4
 - 1.2.1 No “model” in DP 4
 - 1.2.2 English narrative is too vague and long-winded 4
 - 1.2.3 Flowcharts do more harm than good 4
 - 1.2.4 We have no systematic way of recording user preferences and trade-offs, especially in terms of immediate access to data 6
- 1.3 How Much Does the Functional Specification Matter? 6
- References 7

2 WHAT THE TOOLS ARE AND HOW THEY FIT TOGETHER 8

- 2.1 First, Draw a Logical Data Flow Diagram 9
 - 2.1.1 Error conditions 11
 - 2.1.2 Alternative physical implementations 12
 - 2.1.3 The general system class 15
- 2.2 Next, Put the Detail in a Data Dictionary 15
- 2.3 Define the Logic of the Processes 17

- 2.4 Define the Data Stores: Contents and Immediate Access 18
 - 2.4.1 Are the logical data stores the simplest possible? 19
 - 2.4.2 What immediate accesses will be needed? 20
- 2.5 Using the Tools to Create a Functional Specification 23
- Exercises and Discussion Points 24

3 DRAWING DATA FLOW DIAGRAMS 25

- 3.1 Symbol Conventions 25
 - 3.1.1 External entity 26
 - 3.1.2 Data flow 27
 - 3.1.3 Process 29
 - 3.1.4 Data store 30
- 3.2 Explosion Conventions 31
- 3.3 Error and Exception Handling 33
- 3.4 Guidelines for Drawing Data Flow Diagrams 34
- 3.5 Example: Distribution with Inventory 35
- 3.6 Materials Flow and Data Flow 43
- References 46
- Exercises and Discussion Points 46

4 BUILDING AND USING A DATA DICTIONARY 48

- 4.1 The Problem of Describing Data 48
- 4.2 What We Might Want to Hold in a Data Dictionary 51
 - 4.2.1 Describing a data element 53
 - 4.2.2 Describing data structures 55
 - 4.2.3 Describing data flows 58
 - 4.2.4 Describing data stores 59
 - 4.2.5 Describing processes 60
 - 4.2.6 Describing external entities 61
 - 4.2.7 Describing glossary entries 61
- 4.3 ~~Manual vs. Automated Data Dictionaries~~ 61
- 4.4 What We Might Want to Get out of a Data Dictionary 63
 - 4.4.1 Ordered listings of all entries or various classes of entry with full or partial detail 63
 - 4.4.2 Composite reports 63
 - 4.4.3 Cross-referencing ability 63
 - 4.4.4 Finding a name from a description 64
 - 4.4.5 Consistency and completeness checking 65
 - 4.4.6 Generation of machine-readable data definitions 65
 - 4.4.7 Extraction of data dictionary entries from existing programs 66
- 4.5 ~~An Example of an Automated Data Dictionary~~ 67
- 4.6 ~~Cross-Project or Organization-wide Data Dictionaries~~ 72
- 4.7 Data Dictionaries and Distributed Processing 73
- Appendix: Commercially Available Data Dictionary Software Packages 75
- References 75
- Exercises and Discussion Points 75

5 ANALYZING AND PRESENTING PROCESS LOGIC 76

- 5.1 The Problems of Expressing Logic 77
 - 5.1.1 Not only but notwithstanding, and/or unless . . . 76
 - 5.1.2 Greater than, less than 78
 - 5.1.3 And/or ambiguity 79
 - 5.1.4 Undefined adjectives 80
 - 5.1.5 Handling combinations of conditions 80
- 5.2 Decision Trees 83
- 5.3 Decision Tables 88
 - 5.3.1 Conditions, actions, and rules 88
 - 5.3.2 Building the rule matrix 90
 - 5.3.3 Indifference 90
 - 5.3.4 Extended entry; the freight rate problem 92
 - 5.3.5 Decision tables vs. decision trees 95
- 5.4 Structured English, Pseudocode, and "Tight English" 95
 - 5.4.1 The "structures" of structured programming 96
 - 5.4.2 Conventions for structured English 100
 - 5.4.3 Pseudocode 102
 - 5.4.4 Logically "tight English" 104
 - 5.4.5 Pros and cons of the four tools 105
 - 5.4.6 Who does what? 107
- References 108
- Exercises and Discussion Points 108

6 DEFINING THE CONTENTS OF DATA STORES 110

- 6.1 What Comes Out Must Go In 110
- 6.2 Simplifying Data Store Contents by Inspection 112
- 6.3 Simplifying Data Store Contents by Normalization 114
 - 6.3.1 The vocabulary of normalization 115
- 6.4 Some Normalized Forms Are Simpler than Others 117
 - 6.4.1 First normal form (1NF) 117
 - 6.4.2 Second normal form (2NF) 118
 - 6.4.3 Third normal form (3NF) 118
- 6.5 Making Relations out of Relations—Projection and Join 119
 - 6.5.1 Projection 121
 - 6.5.2 Join 122
- 6.6 The Importance of Third Normal Form 123
- 6.7 A Practical Example of 3NF 124
 - 6.7.1 Normalization of the CUSTOMERS data store 124
 - 6.7.2 Normalization of the BOOKS data store 127
 - 6.7.3 Normalization of the ACCOUNTS RECEIVABLE data store 128
 - 6.7.4 Normalization of the INVENTORY data store 129
 - 6.7.5 Putting the relations together 129
- References 130
- Exercises and Discussion Points 131

7 ANALYZING RESPONSE REQUIREMENTS 132

- 7.1 Describing the Ways Data Are Used 132
- 7.2 Physical Techniques for Immediate Access 134
 - 7.2.1 Indexes 134
 - 7.2.2 Hierarchical records 136
- 7.3 General Inquiry Language Capability 139
- 7.4 Types of Query 141
 - 7.4.1 Entities and attributes 141
 - 7.4.2 Six basic query types 142
 - 7.4.3 Variations on the basic types of queries 145
- 7.5 Finding out What the User's Needs and Preferences Are 145
 - 7.5.1 Operational access vs. informational access 145
 - 7.5.2 Getting a composite wish list 146
 - 7.5.3 Refining the wish list 150
- 7.6 Security Considerations 152
- Appendix: General Inquiry Packages 152
- References 153
- Exercises and Discussion Points 153

8 USING THE TOOLS: A STRUCTURED METHODOLOGY 154

- 8.1 The Initial Study 154
- 8.2 The Detailed Study 157
 - 8.2.1 Defining in more detail who the users of a new system would be 158
 - 8.2.2 Building a logical model of the current system 159
 - 8.2.3 Refining the estimates of IRACIS 160
- 8.3 Defining a "Menu" of Alternatives 162
 - 8.3.1 Deriving objectives for the new system from the limitations of the current system 162
 - 8.3.2 Developing a logical model of the new system 164
 - 8.3.3 Producing tentative alternative physical designs 164
- 8.4 Using the "Menu" To Get Commitment from User Decision-makers 168
- 8.5 Refining the Physical Design of the New System 169
 - 8.5.1 Refining the logical model 169
 - 8.5.2 Designing the physical data base 170
 - 8.5.3 Deriving the hierarchy of modular functions that will be programmed 170
 - 8.5.4 Defining the new clerical tasks that will interface with the new system 170
 - 8.5.5 A note on estimating 171
- 8.6 Later Phases of the Project 174
- References 174
- Exercises and Discussion Points 175

9 DERIVING A STRUCTURED DESIGN FROM THE LOGICAL MODEL 176

- 9.1 The Objectives of Design 177
 - 9.1.1 Performance considerations 177
 - 9.1.2 Control considerations 181
 - 9.1.3 Changeability considerations 182
- 9.2 Structured Design for Changeability 184
 - 9.2.1 What makes for a changeable system? 184
 - 9.2.2 Deriving a changeable system from the data flow diagram 186
 - 9.2.3 Module coupling 189
 - 9.2.4 Well-formed modules: cohesiveness, cohesion, binding 191
 - 9.2.5 Scope of effect/scope of control problems 193
- 9.3 The Trade-off Between Changeability and Performance 195
- 9.4 An Example of Structured Design 197
 - 9.4.1 The boundaries of the design 198
 - 9.4.2 Physical file design considerations 199
 - 9.4.3 Locating the central transform of the data flow diagram 206
 - 9.4.4 Refining the design from the top down 207
- 9.5 Top-down Development 213
 - 9.5.1 Possible top-down versions of the CBM system 215
 - 9.5.2 Why develop top-down? 217
 - 9.5.3 The role of the analyst 218
 - 9.5.4 Summary 221
- References 221
- Exercises and Discussion Points 222

10 INTRODUCING STRUCTURED SYSTEMS ANALYSIS INTO YOUR ORGANIZATION 223

- 10.1 Steps in Implementation of Structured Systems Analysis 223
 - 10.1.1 Reviewing the ground rules for conducting projects 223
 - 10.1.2 Establishing standards and procedures for the use of the data dictionary and other software 226
 - 10.1.3 Training analysts in the use of the tools and techniques 226
 - 10.1.4 Orienting users to the new approaches 227
- 10.2 Benefits and Problems 228
 - 10.2.1 Benefits from using structured systems analysis 228
 - 10.2.2 Potential problems 230
- References 231

Glossary 233

Index 239