Contents

Foreword xv

Acknowledgments xvii

Introduction xix

Part 1. Systems Theory 1

Chapter 1. Systems Theory 3

1.1. The definition of a system 3
1.2. Definition of a complex system 5
1.3. Definition of a system of systems 6
1.4. The systems approach 7
1.4.1. The reductionist approach 7
1.4.2. The holistic approach 8
1.4.3. The analytical and systemic approach 9
1.4.4. The emergence 10
1.5. The systemic method 14
1.5.1. Systemic exploration 15
1.5.2. Qualitative modeling 17
1.5.3. Quantitative modeling 20
1.5.4. The simulation 21
1.6. How to understand the complexity of a system 22
1.6.1. Theme 1: the system is in an environment 23
1.6.2. Theme 2: the system is structured 24
1.6.3. Theme 3: the system evolves over time and has a dynamic 24
1.6.4. Theme 4: the system is controlled 25
1.7. Conclusion on systems theory 31

Part 2. Systems and Requirements Engineering 33

Chapter 2. Introduction to Systems Engineering 35

2.1. The system meets needs 35
2.1.1. Definition of a need 35
2.2. Definition of a stakeholder requirement 36
2.3. How to go from need to system 36
2.4. Definition of systems engineering 36
2.5. Iterative systems engineering process 38
2.5.1. Prescription and system construction 38
2.5.2. Needs analysis 40
2.5.3. Requirements specification 41
2.5.4. Functional and logical design 41
2.5.5. Physical design 42
2.5.6. Verification and validation 43
2.5.7. Evaluation and comparison 43
2.5.8. The requirements cycle 44
2.6. System architecture 45
2.7. V-cycle 47
2.8. W-cycle 48
2.9. Conclusion on systems engineering 49

Chapter 3. Introduction to Requirements Engineering 51

3.1. Definition of requirements engineering 51
3.2. The importance and challenges of requirements engineering 53
3.3. Problem domain and solution domain 54
3.4. Formalizing stakeholder needs and system requirements 55
3.5. Validate system requirements 56
3.6. Allocate system requirements to subsystems 56
3.7. Validate the allocated requirements of the subsystems 56
3.8. Choose the solution: define the requirements of the subsystems 56
3.9. Managing requirements 57
3.9.1. Why manage requirements? 57
3.9.2. What are the risks associated with poor requirements management? 57
3.9.3. Managing requirement attributes 58
3.9.4. Managing the configuration of requirements 59
3.9.5. Managing changes 61
3.10. Conclusion on requirements engineering 62

Part 3. Definition of Requirements 65

Chapter 4. Unifying Thread Example 67

4.1. Objective of the unifying thread example 67
4.2. Presentation of the raw specifications 68

Chapter 5. Needs Analysis 71

5.1. Objectives of needs analysis 71
5.2. Stakeholder identification 72
5.2.1. Definition of stakeholder 72
5.2.2. PESTEL analysis 73
5.2.3. Analysis by type of environment 74
5.2.4. Application to the unifying thread example: stakeholder identification 76
5.3. Identification of external interactions 78
5.3.1. Objectives of identifying external interactions 78
5.3.2. Application to the unifying thread example: identification of external interactions 80
5.4. Collection of needs 83
5.4.1. Why capture needs? 83
5.4.2. Methods for collecting needs 84
5.4.3. Application to the unifying thread example: formalization of needs 85
5.5. Identification of the life cycle. 87
5.5.1. Building the system life cycle 87
5.5.2. Application to the unifying thread example: identifying the life cycle . . . 89
5.6. Identification of the system's missions 91
5.6.1. Definition of the system's missions 91
5.6.2. Application to the unifying thread example: mission definition 94
5.7. Identification of operational scenarios 96
5.7.1. Defining an operational scenario 96
5.7.2. Application to the unifying thread example: definition of operational scenarios 97
5.8. Identification of services and constraints 100
5.8.1. From missions to services 100
5.8.2. Application to the unifying thread example: identification of services and constraints 101
5.9. Formalization of stakeholder needs 102
5.9.1. The input specifications 102
5.9.2. Application to the unifying thread example: formalization of needs 107
5.10. Conclusion on needs analysis 110

Chapter 6. Requirements Specification 111

6.1. Objective of the requirements specification process 111
6.2. Identification of the system's functional modes. 113
6.2.1. Definition and purpose of the functional modes 113
6.2.2. What to do to identify functional modes 114
6.2.3. Application to the unifying thread example: identifying the functional modes 115
6.4. Identification of system functions 116
6.4.1. What is a system function?. 116
6.4.2. What to do to identify the system functions 119
6.4.3. Application to the unifying thread example: identification of system functions 122
6.5. Identification of external interactions 123
6.6. Defining system behaviors 124
6.6.1. Objective of defining the functional behavior of the system 124
6.6.2. Application to the unifying thread example: definition of a functional scenario 124
6.7. Defining the system requirements 126
6.7.1. Why define the system requirements?. 126
6.7.2. What to do to define the system requirements 128
6.7.3. How to define system requirements: application to the unifying thread example. 130
6.8. System specification. 134
6.8.1. The specification document 134
6.8.2. Application to the unifying thread example: system specification 138
6.9. Conclusion on requirements specification 140

Chapter 7. Requirements Validation 143

7.1. General process 144
7.2. Selecting methods and defining validation procedures 145
7.3. Establishing requirements traceability 145
7.3.1. Establishing traceability from needs to requirements 145
7.3.2. Establish traceability from requirements to needs 145
7.4. Analysis of assumptions and induced requirements 146
7.4.1. Analysis of assumptions 146
7.4.2. Analysis of induced requirements 146
7.5. Rolling out the validation 146
7.5.1. Checking the requirements individually (correctness). 147
7.5.2. Globally checking the requirements (consistency). 147
7.5.3. Validating requirements against needs 148

7.6. Identifying and resolving gaps between needs and requirements 148
7.6.1. Identifying gaps and conflicts 148
7.6.2. Assessing conflicts 148
7.6.3. Study of trade-offs 149
7.6.4. Identifying trade-offs and impacts 149
7.7. Saving a database of validated requirements 149
7.8. Conclusion on requirements validation 150

Part 4. System Design 151

Chapter 8. Functional and Logical Design 153

8.1. Design and functional architecture 153
8.1.1. Static functional architecture 153
8.1.2. Dynamic functional architecture 154
8.1.3. Behavioral functional architecture 156
8.2. Identifying the sub-functions of the system and their interactions 157
8.2.1. Purpose of identifying sub-functions 157
8.2.2. Functional chain analysis approach 158
8.2.3. Implementation of the unifying thread example: identification of sub-functions 161
8.2.4. The outputs of the static architecture 164
8.2.5. Rules on the quality of a functional decomposition 167
8.3. Functional interface analysis 168
8.3.1. The coupling matrices 168
8.3.2. Characterizing functional interfaces 171
8.4. Consolidation of functional modes 171
8.4.1. Objective of consolidating functional modes 171
8.4.2. What to do to identify the functional sub-modes 172
8.4.3. Application to the unifying thread example: resumption of the functional modes 174
8.5. Resuming system functioning 176
8.5.1. Objective of the system recovery 176
8.5.2. Implementation of the unifying thread example: resumption of the system functioning 178
8.6. Propose groupings of sub-functions into logical components 180
8.6.1. The use of logical components 180
8.6.2. Proposing a schematic diagram of the solution 180
8.7. Allocate functional requirements to sub-functions 180
8.7.1. Objective of the allocation of system requirements to sub-functions 180
8.7.2. Application to the unifying thread example: allocation of system requirements to sub-functions 182
8.8. Conclusion on functional design 184

Chapter 9. Verification and Validation of the Functional Architecture . . . 185

9.1. Verification of the functional architecture . . . 185
9.2. Validation of functional architecture . . . 186
9.2.1. Selecting methods and defining validation procedures . . . 187
9.2.2. Establishing traceability . . . 187
9.2.3. Analysis of assumptions . . . 187
9.2.4. Unrolling the validation . . . 188
9.2.5. Performing the revalidation . . . 188
9.2.6. Recording the results of the functional architecture's validation . . . 188
9.3. Conclusion on the verification and validation of functional design . . . 188

Chapter 10. Physical Design . . . 191

10.1. Purpose of physical design . . . 191
10.2. Identification of physical components . . . 193
10.2.1. Identifying components and allocating technical functions . . . 194
10.2.2. Completing the identification of the components with the missions of the system . . . 195
10.2.3. Completing the identification of components with non-functional requirements . . . 195
10.2.4. Identification of variants in the physical architecture . . . 195
10.2.5. Building physical architectures with a product line approach . . . 196
10.2.6. Building physical architectures from a logical architecture . . . 197
10.2.7. Application to the unifying thread example: identification of components . . . 198
10.3. Decomposition of the components . . . 201
10.3.1. Architecture principles . . . 202
10.3.2. Coupling matrix . . . 205
10.3.3. Global coupling quality of an architecture . . . 206
10.3.4. The physical tree structure . . . 209
10.3.5. The architecture of the components . . . 211
10.3.6. Application to the unifying thread example: components and interfaces . . . 212
10.4. Characterization of interfaces . . . 215
10.5. Identification of system configurations . . . 218
10.5.1. Definition of a technical configuration . . . 218
10.5.2. Activities to identify technical configurations . . . 219
10.5.3. Application to the unifying thread example: identification of configurations . . . 220
10.6. Identifying the physical functioning of the system . . . 222
10.6.1. Definition of the physical functioning of the system . . . 222
10.6.2. Application to the unifying thread example: constructional scenario . . . 223
10.7. Allocation and definition of subsystem requirements . . . 224
10.7.1. Emergence of properties . . . 224

10.7.2. Architecture sizing 226
10.7.3. Allocating non-functional requirements 227
10.7.4. Definition of subsystem requirements 228
10.8. Conclusion on the physical design 230

Chapter 11. Verification and Validation of the Physical Architecture 231

11.1. Verification of the physical architecture 231
11.1.1. Defining physical design verification procedures. 232
11.1.2. Performing physical design verification 232
11.1.3. Rechecking the physical design 233
11.1.4. Recording the results of the physical design verification 233
11.2. Validation of the physical architecture 233
11.3. Conclusion on the verification and validation of the physical architecture . . 235

Chapter 12. Evaluation and Comparison of Solutions 237

12.1. Evaluation of architectures 237
12.1.1. Selection of common criteria 237
12.1.2. Value criteria. 237
12.1.3. Risk and cost criteria 238
12.2. Comparison of architectures. 239
12.2.1. Selecting stakeholders. 239
12.2.2. Characterizing the weight of each criterion 239
12.2.3. Getting stakeholders to vote 240
12.2.4. Comparing architectures 240
12.2.5. Justifying architectural choices. 241
12.3. Conclusion on the evaluation and comparison of the architectures 241

Part 5. Virtual System Integration 243

Chapter 13. Integration on a W-Cycle 245

13.1. Virtual integration and real integration 245
13.2. Simulation models 247
13.3. Conclusion on integration on a W-cycle 249

Chapter 14. Creating a Simulable Design Model 251

14.1. Defining the simulation objectives 251
14.2. Simulation and the systemic approach 251
14.3. Analysis and modeling of system architecture. 252
14.4. Analysis and causal modeling. 254
14.4.1. Extracting simulation variables. 254
14.4.2. Relationships between variables 255

14.4.3. Feedback loops 255
14.4.4. Application to the unifying thread example: causal analysis 258
14.5. Formalizing the simulation model 261
14.5.1. Classification of variables 261
14.5.2. Dynamic modeling 262
14.5.3. Application to the unifying thread example: formalization of the causal model 265
14.6. Running the simulation 266
14.6.1. Identifying and configuring equations 266
14.6.2. Configuring the simulation 266
14.6.3. Application to the unifying thread example: simulation 266
14.7. Analyzing the results 268
14.7.1. Purpose of the analysis of the results 268
14.7.2. Application to the unifying thread example: analysis of the results 268
14.8. Conclusion on the realization of a simulable design model 269

Chapter 15. Making a Simulable Specification Model 271

15.1. The monitoring model 271
15.1.1. Why make a monitoring model? 271
15.1.2. How to build a monitoring model 272
15.2. Writing structured requirements 273
15.2.1. Reminder on the structure of a requirement 273
15.2.2. Application to the unifying thread example: selecting structured requirements 274
15.3. Formalizing property requirements 274
15.3.1. Properties 274
15.3.2. How to formalize a property 275
15.3.3. Application to the unifying thread example: formalization of properties 276
15.4. Building a monitoring model 278
15.4.1. Application to a property 278
15.4.2. Creating the specification model 279
15.5. Running the simulation 280
15.5.1. Connecting the specification model 280
15.5.2. Application to the unifying thread example: running the specification model 281
15.6. Analyzing the results 283
15.6.1. Result from the specification model 283
15.6.2. Application to the unifying thread example 283
15.7. Conclusion on the realization of a simulable specification model 288

Part 6. System Integration 289

Chapter 16. Using Models 291

16.1. Models in systems engineering 291
16.2. Typology of models 291
16.3. Analytical behavioral models 292
16.3.1. Using the behavioral model in V&V. 293
16.3.2. Model-based design 293
16.4. Business analytical models 296
16.4.1. Using the business model in V&V 296
16.5. Conclusion on the use of models 296

Chapter 17. System Integration, Verification and Validation 297

17.1. System integration 297
17.1.1. Receiving subsystems 298
17.1.2. Assembling subsystems 299
17.2. Checking the system 299
17.2.1. Defining the system verification strategy 300
17.2.2. Defining the system verification procedures 302
17.2.3. Performing the system check 303
17.2.4. Establishing system compliance 304
17.3. System validation 304
17.3.1. Preparing for validation 305
17.3.2. Performing validation 306
17.3.3. Analyzing the results 307
17.3.4. Saving validation results 307
17.4. Conclusion on system integration, verification and validation 308

General Conclusion 309

References 311

Index 315