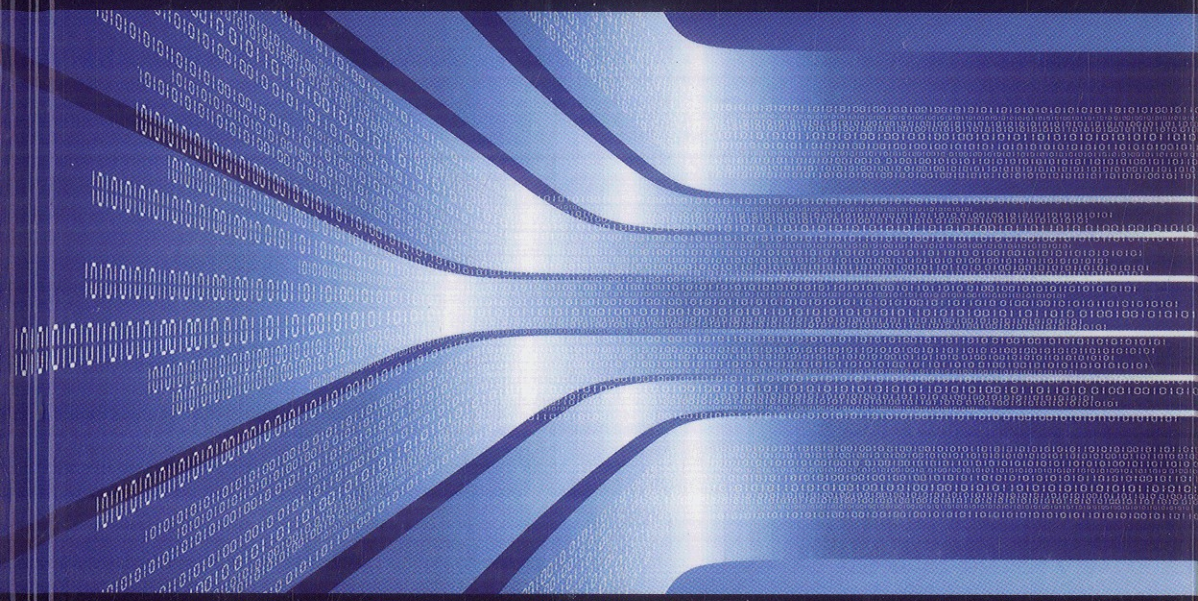

Sustainable IT Architecture

The Progressive Way of Overhauling Information Systems with SOA

Pierre Bonnet
Jean-Michel Detavernier
Dominique Vauquier
Jérôme Boyer and Erik Steinholtz

iSTE
WILEY

Table of Contents

Acknowledgements xiii

Foreword xv

Preface xxi

Guide for the Reader xxvii

Introduction to the SOA Project at SMABTP xxxi

Chapter 1. Initial Perspectives 1

1.1. 50 years of computing – an overview 1
1.2. What remains today? 5

Part I. Why a Sustainable Information System? 7

Chapter 2. Company-oriented Services 9

2.1. Consequences of the Internet revolution 9
2.2. What do the leading market players say? 12
2.3. What do the chief information officers think? 14
2.4. The issues faced at general management level 14
2.5. Levels of maturity 16

Chapter 3. SOA Maturity Levels 21

3.1. Towards the creation of a more agile information system 21
3.2. Cosmetic SOA 23
3.3. Extended SOA 24
3.4. Overhaul SOA 26
3.5. The matrices of SOA maturity 28

3.5.1. The matrix showing the definitions of SOA 28
3.5.2. The matrix showing the quality criteria of SOA 29
3.5.3. The matrix showing the strengths and weaknesses of SOA 29

Chapter 4. Economic and Social Aspects 31

4.1. Removal of obstacles that may slow down the progressive overhaul of an information system 32
4.2. The future of IT specialists 33
4.3. Off-shoring 33
4.4. The generation mix 34
4.5. The role of software infrastructure editors 35

Part II. The Principles of SOA 37

Chapter 5. The Properties of SOA 39

5.1. The definition of service for users 41
5.1.1. The user of the service 42
5.1.2. A business ambiguity 42
5.1.3. An example of a business service 43
5.2. The definition of service for IT specialists 44
5.2.1. The granularity of service 44
5.2.2. The separation of concerns 46
5.2.3. The service categories 47
5.2.4. Batch services 49
5.3. The properties of basic SOA 50
5.3.1. Loose coupling 50
5.3.2. Communication by messages 51
5.3.3. Design by contract 52
5.3.4. The limits of the basic properties 56
5.4. The properties of agility 56
5.4.1. The difference between the version and the variant of a service . . 58
5.4.2. Agility of the data 60
5.4.3. Agility of the rules 65
5.4.4. Agility of the processes 66
5.4.5. Agility of the human–computer interface 67

Chapter 6. Orchestration (BPM and SOA) 69

6.1. Multiple requirements in orchestration 71
6.1.1. Orchestration and SOA maturity levels 71
6.1.2. Functional requirements 73
6.1.3. Technical requirements 75
6.1.4. Enterprise architecture requirements 77

6.2. The levels of orchestration 78
6.2.1. Orchestration at the process level 79
6.2.2. Orchestration at screen level 80
6.2.3. Orchestration at the micro-process level (use cases) 81
6.2.4. Orchestration at the business service level 82
6.2.5. Orchestration between domains through the use of ESB 83
6.2.6. The orchestration of batches 83
6.3. The techniques of orchestration 85
6.3.1. The BPM engine 85
6.3.2. The business rules engine 86
6.3.3. Specific programming 86
6.4. Towards the homogenization of orchestration 87
6.4.1. Unified modeling 87
6.4.2. Unified standard 89
6.5. The benefits of orchestration 91
6.5.1. Advantages 91
6.5.2. Disadvantages 91

Part III. The Need for an Enterprise Method 93

Chapter 7. The Discovery of Services (Reference Framework and Urbanization) 95

7.1. New needs for the information system 96
7.1.1. Expansiveness and progressiveness 97
7.1.2. Mobilizing the many different competences 98
7.2. Why are different methods seldom used within companies? 98
7.3. Reference frameworks 101
7.3.1. Zachman's framework 101
7.3.2. TOGAF 102
7.3.3. Peter Herzum's framework 103
7.3.4. Important information to be taken from the reference frameworks . 104
7.4. Essential tools 105
7.4.1. UML (Unified Modeling Language) 105
7.4.2. MDA (Model Driven Architecture) 106
7.4.3. Urbanization of the information system 107

Chapter 8. The Praxeme Enterprise Method 111

8.1. Praxeme: the initiative behind a public method 112
8.2. The Praxeme method 112
8.2.1. Product 113
8.2.2. Process 113
8.2.3. Procedures 114

8.2.4. Combining the three dimensions 114
8.3. Enterprise system topology according to the Praxeme method 115
8.3.1. Upstream models 115
8.3.2. Logical (SOA), technical and software architecture models. 117
8.3.3. Hardware and physical architecture models 117
8.3.4. Enterprise system topology 118
8.3.5. Pre-modeling 119
8.4. What the Praxeme method means for SOA 120
8.4.1. How can we find the correct services? 120
8.4.2. The link between urbanization, the object-oriented approach and SOA 121
8.5. Advantages of the Praxeme method 124
8.5.1. A method that unites different approaches and integrates SOA 124
8.5.2. Risks associated with the Praxeme method. 126

Chapter 9. Modeling with Praxeme 129

9.1. The modeling of requirements. 130
9.2. Semantic modeling 130
9.2.1. The basic principles 130
9.2.2. How to obtain a semantic model 133
9.2.3. How to validate a semantic model 134
9.2.4. Semantic models and property rights – who owns a semantic model? 134
9.2.5. The structure of a semantic model 135
9.3. Pragmatic modeling 137
9.3.1. The basic principles 137
9.3.2. A new procedure for designing processes 139
9.3.3. Usage view. 140
9.4. Pre-modeling 142
9.5. Logical modeling 143
9.5.1. SOA's style of logical architecture 143
9.5.2. Service-oriented architecture as logical architecture 144
9.5.3. Types of logical components 145
9.5.4. The strata of logical architecture 151
9.5.5. Pivot language. 153
9.5.6. Service algorithm specification. 154
9.5.7. Specification of the services' pre- and post-conditions. 154
9.5.8. Logical architecture of data 156
9.5.9. Logical architecture of data repositories 157
9.5.10. Logical architecture and user interface 158
9.5.11. Designing a logic for tests 159

9.5.12. Considering ERP 160
9.5.13. Considering existent assets 160
9.5.14. Federation of systems. 160
9.5.15. Roles of logical modeling 161
9.6. Logical modeling of batch computing 162
9.7. Technical modeling 163
9.7.1. Required competences 163
9.7.2. Technical/logical negotiation 164
9.8. Software modeling 166
9.8.1. General principles 166
9.8.2. Towards the industrialization of programming 169
9.9. Benefits of the methodology. 169
9.9.1. Opportunities 169
9.9.2. Obstacles 171

Part IV. Mastering Existing Techniques 173

Chapter 10. Tools for Industrializing the Method 175

10.1. Requirements in the industrialization of procedures 176
10.2. Frameworks and design patterns. 178
10.2.1. From services framework to virtual machines 179
10.2.2. Frameworks and human–machine interfaces 182
10.2.3. Design patterns 186
10.3. Tools for increased agility 189
10.3.1. Rules engine 189
10.3.2. Reference data management system 196
10.4. Representation tools 203
10.4.1. Modeling CASE tool 203
10.4.2. Formal language (pseudo-language) 207
10.4.3. MDA 209
10.5. Tools for tests and management 212
10.5.1. Non-regression tests 212
10.5.2. Designing tests and test data. 213
10.5.3. Different levels of tests. 214
10.6. Tools for the management of different versions and the configuration of programs 216
10.6.1. The level of versions and variants 216
10.6.2. The level of delivery packages 218
10.7. Benefits of using tools in the method 219
10.7.1. Opportunities. 219
10.7.2. Risks. 220

Chapter 11. Systems Integration and Common Information Language . . 223

11.1. New requirements in communication 225
11.1.1. Increase of data flow 225
11.1.2. Considering the business 225
11.1.3. Take the bus! 227
11.2. ESB's functions 227
11.2.1. Use perimeter 227
11.2.2. ESB's components 230
11.3. Integrating ESB into SI 235
11.3.1. Towards a common language 235
11.4. ESB's benefits 239
11.4.1. Opportunities 239
11.4.2. Limitations 240

Chapter 12. SOA Platform 243

12.1. Requirements for the global vision of technical architecture 244
12.2. New technical components 245
12.2.1. The transformation of data 245
12.2.2. From a directory to a registry of services 248
12.2.3. Security 250
12.2.4. Traceability of services in production 252
12.2.5. BAM and CEP 254
12.2.6. Business Intelligence 255
12.2.7. Editing 257
12.3. Managing performance 259
12.3.1. A new order of things? 259
12.3.2. Best practice 260
12.3.3. Testing performance 261
12.4. Managing exploitation 263
12.5. Managing maintenance 269
12.6. Benefits of SOA platforms 265
12.6.1. Opportunities 265
12.6.2. Limitations 266

Chapter 13. Rules Management at the Scale of the Whole Enterprise . . . 267
Jérôme BOYER, ILOG Software

13.1. Overview 267
13.2. Deep view 268
13.3. When to use a rule engine 271
13.4. Logical architecture view 273
13.5. BRMS and SOA 283

Chapter 14. Semantic Integration 287
Erik STEINHOLTZ, Progress Software

14.1. Enabling the adaptive enterprise 288
14.2. Inhibitors for change 289
14.3. Definition of semantic integration 290
14.4. Parallel track information modeling 291
14.5. Change inhibitors addressed with semantic integration 294
14.6. Putting it to work 295
14.6.1. Canonicalizing the BIM 295
14.6.2. The quick win: a pilot project 296
14.6.3. Using the CIM for integration 296
14.6.4. Tools used 297
14.6.5. Managing change and keeping the models alive 298

Conclusion 299

Weblinks 303

Bibliography 305

Special Technical Note 307

Index 309