Contents

Foreword . . . .**xiii**

About the Authors . . . .**xvii**

Preface. . . . .**xix**

1 Introduction. **1**

1.1 What Are Web Services? 3

1.2 Benefits of Web Services 6

1.3 Challenges of Web Service Development 8

1.3.1 Evolving Technologies and Products 8

1.3.2 Security 9

1.3.3 Reliability, Availability, and Scalability. 10

1.4 Typical Web Service Scenarios 11

1.4.1 Interacting with Business Partners 14

1.4.2 Integrating with Existing Enterprise Information Systems. 15

1.4.3 Reaching Diverse Clients 15

1.4.4 Aggregation of Partner Data 16

1.5 J2EE 1.4: The Platform for Web Services 17

1.5.1 J2EE Platform Overview. 17

1.5.2 J2EE Platform Benefits 19

1.5.3 J2EE Platform Technologies. 21

1.6 Conclusion 24

2 Standards and Technologies **27**

2.1 Overview of Web Service Standards 27

2.1.1 Extensible Markup Language 29

2.1.2 Simple Object Access Protocol. 33

2.1.3 Registry Standards. 35

2.1.4 Web Services Description Language 36

2.1.5 Emerging Standards 40

2.2 J2EE: The Integrated Platform for Web Services. 40

2.2.1 Java^{TM} APIs for XML Processing 41
2.2.2 Java^{TM} API for XML-Based RPC 44
2.2.3 Java^{TM} API for XML Registries 47
2.2.4 SOAP with Attachments API for Java^{TM}. 48
2.2.5 Web Service Technologies Integrated in J2EE Platform . . 49
2.2.6 Support for WS-I Basic Profile 51
2.3 Other Java-XML Technologies 52
2.4 Conclusion 54

3 Service Endpoint Design. 57
3.1 Example Scenarios 58
3.2 Flow of a Web Service Call 59
3.3 Key Web Services Design Decisions 61
3.4 Designing a Service's Interaction Layer. 66
3.4.1 Designing the Interface 66
3.4.2 Receiving Requests 89
3.4.3 Delegating Web Service Requests to Processing Layer . . . 92
3.4.4 Formulating Responses 98
3.5 Processing Layer Design 99
3.6 Publishing a Web Service 101
3.6.1 Publishing a Service in a Registry 101
3.6.2 Understanding Registry Concepts 102
3.6.3 Registry Implementation Scenarios 103
3.7 Handling XML Documents in a Web Service 105
3.7.1 Exchanging XML Documents. 107
3.7.2 Separating Document Manipulation from
Processing Logic 111
3.7.3 Fragmenting XML Documents 111
3.7.4 Using XML 111
3.7.5 Using JAXM and SAAJ Technologies 112
3.8 Deploying and Packaging a Service Endpoint 113
3.8.1 Service Information in the Deployment Descriptors. 113
3.8.2 Package Structure 117
3.9 Conclusion 119

4 XML Processing 121
4.1 XML Overview. 122
4.1.1 Document Type and W3C XML Schema Definitions . . . 125
4.1.2 XML Horizontal and Vertical Schemas 126

4.1.3 Other Specifications Related to XML 127
4.2 Outline for Handling XML Documents 128
4.3 Designing XML-Based Applications 131
4.3.1 Designing Domain-Specific XML Schemas 131
4.3.2 Receiving and Sending XML Documents 135
4.3.3 Validating XML Documents 139
4.3.4 Mapping Schemas to the Application Data Model 143
4.3.5 Choosing Processing Models 151
4.3.6 Fragmenting Incoming XML Documents 153
4.3.7 Abstracting XML Processing from Application Logic . . . 155
4.3.8 Design Recommendation Summary 163
4.4 Implementing XML-Based Applications 164
4.4.1 Choosing an XML Processing Programming Model 164
4.4.2 Combining XML Processing Techniques 176
4.4.3 Entity Resolution 178
4.5 Performance Considerations 182
4.5.1 Limit Parsing of Incoming XML Documents 184
4.5.2 Use the Most Appropriate API 184
4.5.3 Choose Effective Parser and Style Sheet Implementations186
4.5.4 Reduce Validation Cost 188
4.5.5 Reduce the Cost of Referencing External Entities 190
4.5.6 Cache Dynamically Generated Documents 194
4.5.7 Use XML Judiciously 194
4.6 Conclusion 196

5 Client Design 199

5.1 Choosing a Communication Technology 200
5.2 Scenarios for Web Services-Based Client Applications 201
5.2.1 Designing J2EE Clients 205
5.2.2 Designing J2SE Clients 207
5.2.3 J2ME Clients 208
5.3 Developing Client Applications to Use a Web Service 210
5.3.1 Communication Modes for Accessing a Service 212
5.3.2 Locating and Accessing a Service 219
5.3.3 Stubs and Call Configuration 223
5.3.4 WSDL-to-Java Type Mapping 225
5.3.5 Processing Return Values 227
5.3.6 Handling Exceptions 230
5.4 General Considerations 235

5.4.1 Managing Conversational State 235
5.4.2 Enhancing User Experience 237
5.4.3 Server-Side Design Considerations for Clients. 239
5.4.4 Packaging 240
5.5 Conclusion 245

6 Enterprise Application Integration 247

6.1 Integration Requirements and Scenarios 248
6.1.1 Typical Integration Scenarios 248
6.1.2 Example Integration Scenarios 249
6.2 J2EE Integration Technologies. 252
6.2.1 Relational Database Integration Technologies 253
6.2.2 Messaging Technologies. 255
6.2.3 EIS Access Technologies 259
6.2.4 Web Service and XML Technologies 262
6.3 Integration Design Approaches 263
6.3.1 Web Services Approach 266
6.3.2 Enterprise Beans and JMS Approach 271
6.3.3 The Connector Approach 271
6.3.4 Combining Approaches into an Integration Architecture . 272
6.4 Data Integration Guidelines 274
6.4.1 Data Mapping in EAI Applications. 274
6.4.2 Data Transformation 275
6.4.3 Data Filtering 282
6.5 Guidelines for Integration. 283
6.5.1 Integrating Custom Legacy Systems. 283
6.5.2 Using Screen Scraping for Integration 284
6.5.3 Metadata 285
6.5.4 Using Registries for Integration 286
6.5.5 Versioning and Evolution 286
6.5.6 Writing Applications for Integration. 288
6.6 Conclusion 288

7 Security. 291

7.1 Security Scenarios. 292
7.1.1 General Security Requirements. 293
7.1.2 Security Implications of the Operational Environment. . . 297
7.2 J2EE Platform Security Model 297
7.2.1 Authentication. 299
7.2.2 Authorization. 304

7.2.3 Confidentiality and Integrity 307
7.3 Security for Web Service Interactions 308
7.3.1 Endpoint Programming Model 308
7.3.2 Client Programming Model. 314
7.3.3 Propagating Component Identity 316
7.3.4 Handling Authorization. 320
7.3.5 JAX-RPC Security Guidelines 324
7.4 Message-Level Web Service Security 326
7.4.1 Understanding Message-Level Security 326
7.4.2 Comparing Security Mechanisms 328
7.4.3 Emerging Message-Level Security Standards. 331
7.5 Conclusion 337

8 Application Architecture and Design 339

8.1 Overview of Adventure Builder 339
8.1.1 Order Processing Center Sub-Modules and Interactions . 342
8.2 Order Processing Center Architecture and Design 344
8.2.1 Web Service Interaction and Message Exchange 345
8.2.2 Communication Architecture 349
8.3 Endpoint Design Issues. 351
8.3.1 Web Service Interface Development Approach 351
8.3.2 Endpoint Type Considerations 352
8.3.3 Granularity of Service 353
8.3.4 Passing Parameters as Documents or Java Objects. 353
8.3.5 Layering the Service 355
8.3.6 Delegating to Business Logic 357
8.3.7 Client Considerations 357
8.3.8 Publishing Web Service Details 358
8.4 Web Service Communication Patterns 358
8.4.1 Correlating Messages 359
8.4.2 Splitting and Joining Messages. 361
8.4.3 Refactoring Synchronous to Asynchronous Interactions . 363
8.5 Managing Complex Web Service Interactions 366
8.5.1 Passing Context Information on Web Service Calls 366
8.5.2 Handling Multiple Document Types. 371
8.5.3 Consolidating Web Service Interactions. 373
8.5.4 Canonical Data Model. 375
8.6 Building More Robust Web Services 376
8.6.1 Use Idempotent Endpoints 377

8.6.2 Use Client Retries with Idempotent Endpoints 378
8.6.3 Handling Asynchronous Interaction Error Conditions . . . 379
8.7 Conclusion 381
Glossary 383
Index 403