

INTERNATIONAL
STANDARD

ISO/IEC
19510

First edition
2013-07-01

**Information technology — Object
Management Group Business Process
Model and Notation**

*Technologies de l'information — Modèle de procédé d'affaire et notation
de l'OMG*



Reference number
ISO/IEC 19510:2013(E)

© ISO/IEC 2013



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2013

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Table of Contents

Foreword	xxvii
Introduction	xxviii
1 Scope	1
1.1 General	1
2 Conformance	1
2.1 General	1
2.2 Process Modeling Conformance	2
2.2.1 BPMN Process Types	2
2.2.2 BPMN Process Elements	3
2.2.3 Visual Appearance	8
2.2.4 Structural Conformance	8
2.2.5 Process Semantics	9
2.2.6 Attributes and Model Associations	9
2.2.7 Extended and Optional Elements	9
2.2.8 Visual Interchange	10
2.3 Process Execution Conformance	10
2.3.1 Execution Semantics	10
2.3.2 Import of Process Diagrams	10
2.4 BPEL Process Execution Conformance	10
2.5 Choreography Modeling Conformance	10
2.5.1 BPMN Choreography Types	10
2.5.2 BPMN Choreography Elements	11
2.5.3 Visual Appearance	11
2.5.4 Choreography Semantics	11
2.5.5 Visual Interchange	11
2.6 Summary of BPMN Conformance Types	12
3 Normative References	12
3.1 General	12
3.2 Normative	13
3.3 Non-Normative	13
4 Terms and Definitions	16
5 Symbols	16
6 Additional Information	16
6.1 Conventions	16
6.1.1 Typographical and Linguistic Conventions and Style	16
6.1.2 Abbreviations	17
6.2 Structure of this Document	17

6.3 Acknowledgments	17
7 Overview	19
7.1 General	19
7.2 BPMN Scope	20
7.2.1 Uses of BPMN	21
7.3 BPMN Elements	25
7.3.1 Basic BPMN Modeling Elements	26
7.3.2 Extended BPMN Modeling Elements	29
7.4 BPMN Diagram Types	39
7.5 Use of Text, Color, Size, and Lines in a Diagram	39
7.6 Flow Object Connection Rules	40
7.6.1 Sequence Flow Connections Rules	40
7.6.2 Message Flow Connection Rules	41
7.7 BPMN Extensibility	42
7.8 BPMN Example	43
8 BPMN Core Structure	47
8.1 General	47
8.2 Infrastructure	49
8.2.1 Definitions	49
8.2.2 Import	51
8.2.3 Infrastructure Package XML Schemas.....	52
8.3 Foundation	53
8.3.1 Base Element	54
8.3.2 Documentation	54
8.3.3 Extensibility	55
8.3.4 External Relationships	59
8.3.5 Root Element	62
8.3.6 Foundation Package XML Schemas	62
8.4 Common Elements	64
8.4.1 Artifacts	64
8.4.2 Correlation	72
8.4.3 Error	79
8.4.4 Escalation	80
8.4.5 Events	81
8.4.6 Expressions	82
8.4.7 Flow Element	84
8.4.8 Flow Elements Container	86
8.4.9 Gateways	88
8.4.10 Item Definition	89
8.4.11 Message	91
8.4.12 Resources	93
8.4.13 Sequence Flow	95
8.4.14 Common Package XML Schemas	98
8.5 Services	101
8.5.1 Interface	102
8.5.2 EndPoint	103

8.5.3 Operation	103
8.5.4 Service Package XML Schemas	104
9 Collaboration	107
9.1 General	107
9.2 Basic Collaboration Concepts	110
9.2.1 Use of BPMN Common Elements	110
9.3 Pool and Participant	111
9.3.1 Participants	113
9.3.2 Lanes	119
9.4 Message Flow	119
9.4.1 Interaction Node	122
9.4.2 Message Flow Associations	122
9.5 Conversations	123
9.5.1 Conversation Node	127
9.5.2 Conversation	129
9.5.3 Sub-Conversation	129
9.5.4 Call Conversation	130
9.5.5 Global Conversation	131
9.5.6 Conversation Link	131
9.5.7 Conversation Association	134
9.5.8 Correlations	135
9.6 Process within Collaboration	136
9.7 Choreography within Collaboration	136
9.8 Collaboration Package XML Schemas	138
10 Process	143
10.1 General	143
10.2 Basic Process Concepts	147
10.2.1 Types of BPMN Processes	147
10.2.2 Use of BPMN Common Elements	148
10.3 Activities	149
10.3.1 Resource Assignment	152
10.3.2 Performer	154
10.3.3 Tasks	154
10.3.4 Human Interactions	163
10.3.5 Sub-Processes	171
10.3.6 Call Activity	182
10.3.7 Global Task	186
10.3.8 Loop Characteristics	188
10.3.9 XML Schema for Activities	194
10.4 Items and Data	202
10.4.1 Data Modeling	202
10.4.2 Execution Semantics for Data	224
10.4.3 Usage of Data in XPath Expressions	225
10.4.4 XML Schema for Data	228
10.5 Events	232
10.5.1 Concepts	233
10.5.2 Start Event	237

10.5.3 End Event	245
10.5.4 Intermediate Event	248
10.5.5 Event Definitions	259
10.5.6 Handling Events	274
10.5.7 Scopes	280
10.5.8 Events Package XML Schemas	281
10.6 Gateways	286
10.6.1 Sequence Flow Considerations	288
10.6.2 Exclusive Gateway	289
10.6.3 Inclusive Gateway	291
10.6.4 Parallel Gateway	292
10.6.5 Complex Gateway	294
10.6.6 Event-Based Gateway	296
10.6.7 Gateway Package XML Schemas	300
10.7 Compensation	301
10.7.1 Compensation Handler	302
10.7.2 Compensation Triggering	303
10.7.3 Relationship between Error Handling and Compensation	304
10.8 Lanes	304
10.9 Process Instances, Unmodeled Activities, and Public Processes	308
10.10 Auditing	310
10.11 Monitoring	310
10.12 Process Package XML Schemas	311
11 Choreography	315
11.1 General	315
11.2 Basic Choreography Concepts	317
11.3 Data	319
11.4 Use of BPMN Common Elements	319
11.4.1 Sequence Flow	320
11.4.2 Artifacts	321
11.5 Choreography Activities	321
11.5.1 Choreography Task	323
11.5.2 Sub-Choreography	328
11.5.3 Call Choreography	333
11.5.4 Global Choreography Task	335
11.5.5 Looping Activities	335
11.5.6 The Sequencing of Activities	335
11.6 Events	339
11.6.1 Start Events	339
11.6.2 Intermediate Events	340
11.6.3 End Events	343
11.7 Gateways	344
11.7.1 Exclusive Gateway	344
11.7.2 Event-Based Gateway	349
11.7.3 Inclusive Gateway	351
11.7.4 Parallel Gateway	358
11.7.5 Complex Gateway	360

11.7.6 Chaining Gateways	361
11.8 Choreography within Collaboration	361
11.8.1 Participants	361
11.8.2 Swimlanes	362
11.9 XML Schema for Choreography	363
12 BPMN Notation and Diagrams	367
12.1 BPMN Diagram Interchange (BPMN DI)	367
12.1.1 Scope	367
12.1.2 Diagram Definition and Interchange	367
12.1.3 How to Read this Clause	368
12.2 BPMN Diagram Interchange (DI) Meta-model	368
12.2.1 Overview	368
12.2.2 Abstract Syntax	368
12.2.3 Classifier Descriptions	370
12.2.4 Complete BPMN DI XML Schema	378
12.3 Notational Depiction Library and Abstract Element Resolutions	380
12.3.1 Labels	381
12.3.2 BPMNShape	381
12.3.3 BPMNEdge	410
12.4 Example(s)	412
12.4.1 Depicting Content in a Sub-Process	412
12.4.2 Multiple Lanes and Nested Lanes	417
12.4.3 Vertical Collaboration	418
12.4.4 Conversation	419
12.4.5 Choreography	421
13 BPMN Execution Semantics	425
13.1 General	425
13.2 Process Instantiation and Termination	426
13.3 Activities	426
13.3.1 Sequence Flow Considerations	427
13.3.2 Activity	428
13.3.3 Task	430
13.3.4 Sub-Process/Call Activity	430
13.3.5 Ad-Hoc Sub-Process	431
13.3.6 Loop Activity	432
13.3.7 Multiple Instances Activity	432
13.4 Gateways	434
13.4.1 Parallel Gateway (Fork and Join)	434
13.4.2 Exclusive Gateway (Exclusive Decision (data-based) and Exclusive Merge)	434
13.4.3 Inclusive Gateway (Inclusive Decision and Inclusive Merge)	435
13.4.4 Event-based Gateway (Exclusive Decision (event-based))	437
13.4.5 Complex Gateway (related to Complex Condition and Complex Merge)	437
13.5 Events	439
13.5.1 Start Events	439
13.5.2 Intermediate Events	440
13.5.3 Intermediate Boundary Events	440
13.5.4 Event Sub-Processes	440

13.5.5 Compensation	441
13.5.6 End Events	443
14 Mapping BPMN Models to WS-BPEL	445
14.1 General	445
14.2 Basic BPMN-BPEL Mapping	446
14.2.1 Process	447
14.2.2 Activities	448
14.2.3 Events	455
14.2.4 Gateways and Sequence Flows	461
14.2.5 Handling Data	465
14.3 Extended BPMN-BPEL Mapping	469
14.3.1 End Events	469
14.3.2 Loop/Switch Combinations From a Gateway	469
14.3.3 Interleaved Loops	470
14.3.4 Infinite Loops	473
14.3.5 BPMN Elements that Span Multiple WSBPEL Sub-Elements	473
15 Exchange Formats	475
15.1 Interchanging Incomplete Models	475
15.2 Machine Readable Files	475
15.3 XSD	475
15.3.1 Document Structure	475
15.3.2 References within the BPMN XSD	476
15.4 XMI	477
15.5 XSLT Transformation between XSD and XMI	477
Annex A - Changes from v1.2.....	479
Annex B - Diagram Interchange.....	481
Annex C - Glossary.....	499
Annex D - Legal Information	505

List of Figures

Figure 7.1 – Example of a private Business Process 21
Figure 7.2 – Example of a public Process 22
Figure 7.3 – An example of a Collaborative Process 23
Figure 7.4 – An example of a Choreography 23
Figure 7.5 – An example of a Conversation diagram 24
Figure 7.6 – An example of a Collaboration diagram with black-box Pools 43
Figure 7.7 – An example of a stand-alone Choreography diagram 44
Figure 7.8 – An example of a stand-alone Process (Orchestration) diagram 45
Figure 8.1 – A representation of the BPMN Core and Layer Structure 47
Figure 8.2 – Class diagram showing the core packages 48
Figure 8.3 – Class diagram showing the organization of the core BPMN elements 49
Figure 8.4 – Definitions class diagram 50
Figure 8.5 – Classes in the Foundation package 53
Figure 8.6 – Extension class diagram 55
Figure 8.7 – External Relationship Metamodel 60
Figure 8.8 – Artifacts Metamodel 64
Figure 8.9 – An Association 65
Figure 8.10 – The Association Class Diagram 65
Figure 8.11 – A Directional Association 66
Figure 8.12 – An Association of Text Annotation 66
Figure 8.13 – A Group Artifact 67
Figure 8.14 – A Group around Activities in different Pools 67
Figure 8.15 – The Group class diagram 68
Figure 8.16 – A Text Annotation 69
Figure 8.17 – The Correlation Class Diagram 74
Figure 8.18 – Error class diagram 79
Figure 8.19 – Escalation class diagram 80
Figure 8.20 – Event class diagram 82
Figure 8.21 – Expression class diagram 83
Figure 8.22 – FlowElement class diagram 85
Figure 8.23 – FlowElementContainers class diagram 87
Figure 8.24 – Gateway class diagram 88
Figure 8.25 – ItemDefinition class diagram 90
Figure 8.26 – A Message 91
Figure 8.27 – A non-initiating Message 91
Figure 8.28 – Messages Association overlapping Message Flows 92
Figure 8.29 – Messages shown Associated with a Choreography Task 92
Figure 8.30 – The Message class diagram 93
Figure 8.31 – Resource class diagram 94
Figure 8.32 – A Sequence Flow 95
Figure 8.33 – A Conditional Sequence Flow 95
Figure 8.34 – A Default Sequence Flow 96

Figure 8.35 – SequenceFlow class diagram 96
Figure 8.36 – The Service class diagram 102
Figure 9.1 – Classes in the Collaboration package 108
Figure 9.2 – A Pool 111
Figure 9.3 – Message Flows connecting to the boundaries of two Pools 112
Figure 9.4 – Message Flows connecting to Flow Objects within two Pools 112
Figure 9.5 – Main (Internal) Pool without boundaries 113
Figure 9.6 – Pools with a Multi-Instance Participant Markers 113
Figure 9.7 – The Participant Class Diagram 114
Figure 9.8 – A Pool with a Multiple Participant 116
Figure 9.9 – The Participant Multiplicity class diagram 116
Figure 9.10 – ParticipantAssociation class diagram 118
Figure 9.11 – A Message Flow 119
Figure 9.12 – A Message Flow with an Attached Message 120
Figure 9.13 – A Message Flow passing through a Choreography Task 120
Figure 9.14 – The Message Flow Class Diagram 121
Figure 9.15 – MessageFlowAssociation class diagram 123
Figure 9.16 – A Conversation diagram 124
Figure 9.17 – A Conversation diagram where the Conversation is expanded into Message Flows 124
Figure 9.18 – Conversation diagram depicting several conversations between Participants in a related domain 125
Figure 9.19 – An example of a Sub-Conversation 126
Figure 9.20 – An example of a Sub-Conversation expanded to a Conversation and Message Flow 126
Figure 9.21 – An example of a Sub-Conversation that is fully expanded 127
Figure 9.22 – Metamodel of ConversationNode Related Elements 128
Figure 9.23 – A Communication element 129
Figure 9.24 – A compound Conversation element 130
Figure 9.25 – A Call Conversation calling a GlobalConversation 130
Figure 9.26 – A Call Conversation calling a Collaboration 130
Figure 9.27 – A Conversation Link element 131
Figure 9.28 – Conversation links to Activities and Events 132
Figure 9.29 – Metamodel of Conversation Links related elements 133
Figure 9.30 – Call Conversation Links 134
Figure 9.31 – The ConversationAssociation class diagram 135
Figure 9.32 – An example of a Choreography within a Collaboration 137
Figure 9.33 – Choreography within Collaboration class diagram 138
Figure 10.1 – An Example of a Process 143
Figure 10.2 – Process class diagram 144
Figure 10.3 – Process Details class diagram 145
Figure 10.4 – Example of a private Business Process 148
Figure 10.5 – Example of a public Process 148
Figure 10.6 – Activity class diagram 149
Figure 10.7 – The class diagram for assigning Resources 152
Figure 10.8 – A Task object 154
Figure 10.9 – Task markers 155
Figure 10.10 – The Task class diagram 155
Figure 10.11 – A Service Task Object 156
Figure 10.12 – The Service Task class diagram 157

Figure 10.13 – A Send Task Object 158
Figure 10.14 – The Send Task and Receive Task class diagram 158
Figure 10.15 – A Receive Task Object 159
Figure 10.16 – A Receive Task Object that instantiates a Process 160
Figure 10.17 – A User Task Object 161
Figure 10.18 – A Manual Task Object 161
Figure 10.19 – A Business Rule Task Object 162
Figure 10.20 – A Script Task Object 162
Figure 10.21 – Manual Task class diagram 163
Figure 10.22 – User Task class diagram 164
Figure 10.23 – HumanPerformer class diagram 165
Figure 10.24 – Procurement Process Example 168
Figure 10.25 – A Sub-Process object (collapsed) 171
Figure 10.26 – A Sub-Process object (expanded) 172
Figure 10.27 – Expanded Sub-Process used as a “Parallel Box” 172
Figure 10.28 – Collapsed Sub-Process Markers 173
Figure 10.29 – The Sub-Process class diagram 173
Figure 10.30 – An Event Sub-Process object (Collapsed) 175
Figure 10.31 – An Event Sub-Process object (expanded) 175
Figure 10.32 – An example that includes Event Sub-Processes 176
Figure 10.33 – A Transaction Sub-Process 177
Figure 10.34 – A Collapsed Transaction Sub-Process 177
Figure 10.35 – A collapsed Ad-Hoc Sub-Process 179
Figure 10.36 – An expanded Ad-Hoc Sub-Process 179
Figure 10.37 – An Ad-Hoc Sub-Process for writing a book chapter 181
Figure 10.38 – An Ad-Hoc Sub-Process with data and sequence dependencies 182
Figure 10.39 – A Call Activity object calling a Global Task 183
Figure 10.40 – A Call Activity object calling a Process (Collapsed) 183
Figure 10.41 – A Call Activity object calling a Process (Expanded) 183
Figure 10.42 – The Call Activity class diagram 184
Figure 10.43 – CallableElement class diagram 185
Figure 10.44 – Global Tasks class diagram 187
Figure 10.45 – LoopCharacteristics class diagram 188
Figure 10.46 – A Task object with a Standard Loop Marker 189
Figure 10.47 – A Sub-Process object with a Standard Loop Marker 189
Figure 10.48 – Activity Multi-Instance marker for parallel instances 190
Figure 10.49 – Activity Multi-Instance marker for sequential instances 190
Figure 10.50 – ItemAware class diagram 203
Figure 10.51 – DataObject class diagram 204
Figure 10.52 – A DataObject 206
Figure 10.53 – A DataObject that is a collection 206
Figure 10.54 – A Data Store 207
Figure 10.55 – DataStore class diagram 207
Figure 10.56 – Property class diagram 209
Figure 10.57 – InputOutputSpecification class diagram 211
Figure 10.58 – A DataInput 213
Figure 10.59 – Data Input class diagram 213

Figure 10.60 – A Data Output 215
Figure 10.61 – Data Output class diagram 215
Figure 10.62 – InputSet class diagram 218
Figure 10.63 – OutputSet class diagram 219
Figure 10.64 – DataAssociation class diagram 221
Figure 10.65 – A Data Association 221
Figure 10.66 – A Data Association used for an Outputs and Inputs into an Activities 221
Figure 10.67 – A Data Object shown as an output and an inputs 223
Figure 10.68 – A Data Object associated with a Sequence Flow 224
Figure 10.69 – The Event Class Diagram 233
Figure 10.70 – Start Event 238
Figure 10.71 – End Event 245
Figure 10.72 – Intermediate Event 249
Figure 10.73 – EventDefinition Class Diagram 261
Figure 10.74 – Cancel Events 262
Figure 10.75 – Compensation Events 262
Figure 10.76 – CompensationEventDefinition Class Diagram 262
Figure 10.77 – Conditional Events 263
Figure 10.78 – ConditionalEventDefinition Class Diagram 264
Figure 10.79 – Error Events 264
Figure 10.80 – ErrorEventDefinition Class Diagram 265
Figure 10.81 – Escalation Events 265
Figure 10.82 – EscalationEventDefinition Class Diagram 266
Figure 10.83 – Link Events 266
Figure 10.84 – Link Events Used as Off-Page Connector 267
Figure 10.85 – A Process with a long Sequence Flow 268
Figure 10.86 – A Process with Link Intermediate Events used as Go To Objects 268
Figure 10.87 – Link Events Used for looping 269
Figure 10.88 – Message Events 269
Figure 10.89 – MessageEventDefinition Class Diagram 270
Figure 10.90 – Multiple Events 271
Figure 10.91 – None Events 271
Figure 10.92 – Multiple Events 272
Figure 10.93 – SignalEventDefinition Class Diagram 272
Figure 10.94 – Signal Events 272
Figure 10.95 – Terminate Event 273
Figure 10.96 – Timer Events 273
Figure 10.97 – Exclusive start of a Process 274
Figure 10.98 – A Process initiated by an Event-Based Gateway 275
Figure 10.99 – Event synchronization at Process start 275
Figure 10.100 – Example of inline Event Handling via Event Sub-Processes 277
Figure 10.101 – Example of boundary Event Handling 278
Figure 10.102 – A Gateway 286
Figure 10.103 – The Different types of Gateways 287
Figure 10.104 – Gateway class diagram 288
Figure 10.105 – An Exclusive Data-Based Decision (Gateway) Example without the Internal Indicator 289
Figure 10.106 – A Data-Based Exclusive Decision (Gateway) Example with the Internal Indicator 290

Figure 10.107 – Exclusive Gateway class diagram 290
Figure 10.108 – An example using an Inclusive Gateway 291
Figure 10.109 – Inclusive Gateway class diagram 292
Figure 10.110 – An example using an Parallel Gateway 293
Figure 10.111 – An example of a synchronizing Parallel Gateway 293
Figure 10.112 – Parallel Gateway class diagram 294
Figure 10.113 – An example using a Complex Gateway 294
Figure 10.114 – Complex Gateway class diagram 295
Figure 10.115 – Event-Based Gateway 296
Figure 10.116 – An Event-Based Gateway example using Message Intermediate Events 297
Figure 10.117 – An Event-Based Gateway example using Receive Tasks 297
Figure 10.118 – Exclusive Event-Based Gateway to start a Process 298
Figure 10.119 – Parallel Event-Based Gateway to start a Process 298
Figure 10.120 – Event-Based Gateway class diagram 299
Figure 10.121 – Compensation through a boundary Event 302
Figure 10.122 – Monitoring Class Diagram 303
Figure 10.123 – Two Lanes in a Vertical Pool 305
Figure 10.124 – Two Lanes in a horizontal Pool 305
Figure 10.125 – An Example of Nested Lanes 306
Figure 10.126 – The Lane class diagram 307
Figure 10.127 – One Process supporting to another 309
Figure 10.128 – Auditing Class Diagram 310
Figure 10.129 – Monitoring Class Diagram 311
Figure 11.1 – The Choreography metamodel 316
Figure 11.2 – An example of a Choreography 317
Figure 11.3 – A Collaboration diagram logistics example 318
Figure 11.4 – The corresponding Choreography diagram logistics example 319
Figure 11.5 – The use of Sequence Flows in a Choreography 320
Figure 11.6 – The metamodel segment for a Choreography Activity 322
Figure 11.7 – A Collaboration view of Choreography Task elements 323
Figure 11.8 – A Choreography Task 323
Figure 11.9 – A Collaboration view of a Choreography Task 324
Figure 11.10 – A two-way Choreography Task 324
Figure 11.11 – A Collaboration view of a two-way Choreography Task 325
Figure 11.12 – Choreography Task Markers 326
Figure 11.13 – The Collaboration view of a looping Choreography Task 326
Figure 11.14 – The Collaboration view of a Parallel Multi-Instance Choreography Task 327
Figure 11.15 – A Choreography Task with a multiple Participant 327
Figure 11.16 – A Collaboration view of a Choreography Task with a multiple Participant 328
Figure 11.17 – A Sub-Choreography 329
Figure 11.18 – A Collaboration view of a Sub-Choreography 329
Figure 11.19 – An expanded Sub-Choreography 330
Figure 11.20 – A Collaboration view of an expanded Sub-Choreography 330
Figure 11.21 – Sub-Choreography (Collapsed) with More than Two Participants 331
Figure 11.22 – Sub-Choreography Markers 332
Figure 11.23 – Sub-Choreography Markers with a multi-instance Participant 332
Figure 11.24 – A Call Choreography calling a Global Choreography Task 333

Figure 11.25 – A Call Choreography calling a Choreography (Collapsed) 333
Figure 11.26 – A Call Choreography calling a Choreography (expanded) 334
Figure 11.27 – The Call Choreography class diagram 334
Figure 11.28 – A valid sequence of Choreography Activities 336
Figure 11.29 – The corresponding Collaboration for a valid Choreography sequence 337
Figure 11.30 – A valid sequence of Choreography Activities with a two-way Activity 337
Figure 11.31 – The corresponding Collaboration for a valid Choreography sequence with a two-way Activity 338
Figure 11.32 – An invalid sequence of Choreography Activities 338
Figure 11.33 – The corresponding Collaboration for an invalid Choreography sequence 339
Figure 11.34 – An example of the Exclusive Gateway 345
Figure 11.35 – The relationship of Choreography Activity Participants across the sides of the Exclusive Gateway shown through a Collaboration 346
Figure 11.36 – Different Receiving Choreography Activity Participants on the output sides of the Exclusive Gateway 347
Figure 11.37 – The corresponding Collaboration view of the above Choreography Exclusive Gateway configuration 348
Figure 11.38 – An example of an Event Gateway 349
Figure 11.39 – The corresponding Collaboration view of the above Choreography Event Gateway configuration 350
Figure 11.40 – An example of a Choreography Inclusive Gateway configuration 352
Figure 11.41 – The corresponding Collaboration view of the above Choreography Inclusive Gateway configuration 353
Figure 11.42 – An example of a Choreography Inclusive Gateway configuration 354
Figure 11.43 – The corresponding Collaboration view of the above Choreography Inclusive Gateway configuration 355
Figure 11.44 – Another example of a Choreography Inclusive Gateway configuration 356
Figure 11.45 – The corresponding Collaboration view of the above Choreography Inclusive Gateway configuration 357
Figure 11.46 – The relationship of Choreography Activity Participants across the sides of the Parallel Gateway 358
Figure 11.47 – The corresponding Collaboration view of the above Choreography Parallel Gateway configuration 359
Figure 11.48 – An example of a Choreography Complex Gateway configuration 360
Figure 11.49 – The corresponding Collaboration view of the above Choreography Complex Gateway configuration 361
Figure 11.50 – An example of a Choreography Process combined with Black Box Pools 362
Figure 11.51 – An example of a Choreography Process combined with Pools that contain Processes 363
Figure 12.1 – BPMN Diagram 369
Figure 12.2 – BPMN Plane 369
Figure 12.3 – BPMN Shape 369
Figure 12.4 – BPMN Edge 370
Figure 12.5 – BPMN Label 370
Figure 12.6 – Depicting a Label for a DataObjectReference with its state 381
Figure 12.7 – Combined Compensation and Loop Characteristic Marker Example 384
Figure 12.8 – Expanded Sub-Process Example 413
Figure 12.9 – Start and End Events on the Border Example 414
Figure 12.10 – Collapsed Sub-Process 415
Figure 12.11 – Contents of Collapsed Sub-Process 416

Figure 12.12 – Nested Lanes Example 417
Figure 12.13 – Vertical Collaboration Example 418
Figure 12.14 – Conversation Example 420
Figure 12.15 – Choreography Example 422
Figure 13.1 – Behavior of multiple outgoing Sequence Flows of an Activity 427
Figure 13.2 – The Lifecycle of a BPMN Activity 428
Figure 13.3 – Merging and Branching Sequence Flows for a Parallel Gateway 434
Figure 13.4 – Merging and Branching Sequence Flows for an Exclusive Gateway 434
Figure 13.5 – Merging and Branching Sequence Flows for an Inclusive Gateway 435
Figure 13.6 – Merging and branching Sequence Flows for an Event-Based Gateway 437
Figure 13.7 – Merging and branching Sequence Flows for a Complex Gateway 437
Figure 14.1 – A BPMN orchestration process and its block hierarchy 446
Figure 14.2 – An example of distributed token recombination 469
Figure 14.3 – An example of a loop from a decision with more than two alternative paths 470
Figure 14.4 – An example of interleaved loops 471
Figure 14.5 – An example of the WSBPEL pattern for substituting for the derived Process 472
Figure 14.6 – An example of a WSBPEL pattern for the derived Process 472
Figure 14.7 – An example: An infinite loop 473
Figure 14.8 – An example: Activity that spans two paths of a WSBPEL structured element 474
Figure B.1 – Diagram Definition Architecture 483
Figure B.2 – The Primitive Types 483
Figure B.3 – Diagram Definition Architecture 484
Figure B.4 – Diagram Definition Architecture 484
Figure B.5 – Dependencies of the DI package 488
Figure B.6 – Diagram Element 488
Figure B.7 – Node 488
Figure B.8 – Edge 489
Figure B.9 – Diagram 489
Figure B.10 – Plane 489
Figure B.11 – Labeled Edge 490
Figure B.12 – Labeled Shape 490
Figure B.13 – Shape 490

(Blank page)

List of Tables

Table 2.1 – Descriptive Conformance Sub-Class Elements and Attributes 3
 Table 2.2 – Analytic Conformance Sub-Class Elements and Attributes 4
 Table 2.3 – Common Executable Conformance Sub-Class Elements and Attributes 6
 Table 2.4 – Common Executable Conformance Sub-Class Supporting Classes 7
 Table 2.5 – Types of BPMN Conformance 12
 Table 7.1 – Basic Modeling Elements 27
 Table 7.2 – BPMN Extended Modeling Elements 29
 Table 7.3 – Sequence Flow Connection Rules 40
 Table 7.4 – Message Flow Connection Rules 42
 Table 8.1 – Definitions attributes and model associations 50
 Table 8.2 – Import attributes 52
 Table 8.3 – Definitions XML schema 52
 Table 8.4 – Import XML schema 53
 Table 8.5 – BaseElement attributes and model associations 54
 Table 8.6 – Documentation attributes 54
 Table 8.7 – Extension attributes and model associations 56
 Table 8.8 – ExtensionDefinition attributes and model associations 57
 Table 8.9 – ExtensionAttributeDefinition attributes 57
 Table 8.10 – ExtensionAttributeValue model associations 57
 Table 8.11 – Extension XML schema 58
 Table 8.12 – Example Core XML schema 58
 Table 8.13 – Example Extension XML schema 59
 Table 8.14 – Sample XML instance 59
 Table 8.15 – Relationship attributes 61
 Table 8.16 – Reengineer XML schema 61
 Table 8.17 – BaseElement XML schema 62
 Table 8.18 – RootElement XML schema 63
 Table 8.19 – Relationship XML schema 63
 Table 8.20 – Association attributes and model associations 66
 Table 8.21 – Group model associations 68
 Table 8.22 – Category model associations 69
 Table 8.23 – CategoryValue attributes and model associations 69
 Table 8.24 – Text Annotation attributes 70
 Table 8.25 – Artifact XML schema 70
 Table 8.26 – Association XML schema 70
 Table 8.27 – Category XML schema 70
 Table 8.28 – CategoryValue XML schema 71
 Table 8.29 – Group XML schema 71
 Table 8.30 – Text Annotation XML schema 71
 Table 8.31 – CorrelationKey model associations 75
 Table 8.32 – CorrelationProperty model associations 75
 Table 8.33 – CorrelationPropertyRetrievalExpression model associations 76

Table 8.34 – CorrelationSubscription model associations 76
Table 8.35 – CorrelationPropertyBinding model associations 77
Table 8.36 – Correlation Key XML schema 77
Table 8.37 – Correlation Property XML schema 77
Table 8.38 – Correlation Property Binding XML schema 78
Table 8.39 – Correlation Property Retrieval Expression XML schema 78
Table 8.40 – Correlation Subscription XML schema 78
Table 8.41 – Error attributes and model associations 80
Table 8.42 – Escalation attributes and model associations 81
Table 8.43 – FormalExpression attributes and model associations 84
Table 8.44 – FlowElement attributes and model associations 86
Table 8.45 – FlowElementsContainer model associations 87
Table 8.46 – Gateway attributes 89
Table 8.47 – ItemDefinition attributes & model associations 90
Table 8.48 – Message attributes and model associations 93
Table 8.49 – Resource attributes and model associations 94
Table 8.50 – ResourceParameter attributes and model associations 95
Table 8.51 – SequenceFlow attributes and model associations 97
Table 8.52 – FlowNode model associations 98
Table 8.53 – Error XML schema 98
Table 8.54 – Escalation XML schema 98
Table 8.55 – Expression XML schema 98
Table 8.56 – FlowElement XML schema 99
Table 8.57 – FlowNode XML schema 99
Table 8.58 – FormalExpression XML schema 99
Table 8.59 – InputOutputBinding XML schema 99
Table 8.60 – ItemDefinition XML schema 100
Table 8.61 – Message XML schema 100
Table 8.62 – Resources XML schema 100
Table 8.63 – ResourceParameter XML schema 101
Table 8.64 – SequenceFlow XML schema 101
Table 8.65 – Interface attributes and model associations 103
Table 8.66 – Operation attributes and model associations 104
Table 8.67 – Interface XML schema 104
Table 8.68 – Operation XML schema 104
Table 8.69 – EndPoint XML schema 105
Table 9.1 – Collaboration Attributes and Model Associations 108
Table 9.2 – Participant attributes and model associations 115
Table 9.3 – PartnerEntity attributes 115
Table 9.4 – PartnerRole attributes 116
Table 9.5 – ParticipantMultiplicity attributes 117
Table 9.6 – ParticipantMultiplicity Instance attributes 117
Table 9.7 – ParticipantAssociation model associations 119
Table 9.8 – Message Flow attributes and model associations 122
Table 9.9 – MessageFlowAssociation attributes and model associations 123
Table 9.10 – ConversationNode Model Associations 129
Table 9.11 – Sub-Conversation Model Associations 130

Table 9.12 – Call Conversation Model Associations 131
 Table 9.13 – Conversation Link Attributes and Model Associations 133
 Table 9.14 – ConversationAssociation Model Associations 135
 Table 9.15 – Call Conversation XML schema 138
 Table 9.16 – Collaboration XML schema 138
 Table 9.17 – Conversation XML schema 139
 Table 9.18 – ConversationAssociation XML schema 139
 Table 9.19 – ConversationAssociation XML schema 139
 Table 9.20 – ConversationNode XML schema 140
 Table 9.21 – Conversation Node XML schema 140
 Table 9.22 – Global Conversation XML schema 140
 Table 9.23 – MessageFlow XML schema 140
 Table 9.24 – MessageFlowAssociation XML schema 141
 Table 9.25 – Participant XML schema 141
 Table 9.26 – ParticipantAssociation XML schema 141
 Table 9.27 – ParticipantMultiplicity XML schema 142
 Table 9.28 – PartnerEntity XML schema 142
 Table 9.29 – PartnerRole XML schema 142
 Table 9.30 – Sub-Conversation XML schema 142
 Table 10.1 – Process Attributes & Model Associations 145
 Table 10.2 – Process instance attributes 147
 Table 10.3 – Activity attributes and model associations 150
 Table 10.4 – Activity instance attributes 151
 Table 10.5 – Resource Role model associations 153
 Table 10.6 – ResourceAssignmentExpression model associations 153
 Table 10.7 – ResourceParameterBinding model associations 154
 Table 10.8 – Service Task model associations 157
 Table 10.9 – Send Task model associations 159
 Table 10.10 – Receive Task attributes and model associations 160
 Table 10.11 – Business Rule Task attributes and model associations 162
 Table 10.12 – Script Task attributes 163
 Table 10.13 – User Task attributes and model associations 164
 Table 10.14 – User Task instance attributes 165
 Table 10.15 – ManualTask XML schema 166
 Table 10.16 – UserTask XML schema 167
 Table 10.17 – HumanPerformer XML schema 168
 Table 10.18 – PotentialOwner XML schema 168
 Table 10.19 – XML serialization of Buyer process 169
 Table 10.20 – Sub-Process attributes 174
 Table 10.21 – Transaction Sub-Process attributes and model associations 178
 Table 10.22 – Ad-hoc Sub-Process model associations 180
 Table 10.23 – CallActivity model associations 185
 Table 10.24 – CallableElement attributes and model associations 186
 Table 10.25 – InputOutputBinding model associations 186
 Table 10.26 – Global Task model associations 187
 Table 10.27 – Loop Activity instance attributes 189
 Table 10.28 – StandardLoopCharacteristics attributes and model associations 190

Table 10.29 – MultiInstanceLoopCharacteristics attributes and model associations 191
Table 10.30 – Multi-instance Activity instance attributes 193
Table 10.31 – ComplexBehaviorDefinition attributes and model associations 194
Table 10.32 – Activity XML schema 194
Table 10.33 – AdHocSubProcess XML schema 195
Table 10.34 – BusinessRuleTask XML schema 195
Table 10.35 – CallableElement XML schema 196
Table 10.36 – CallActivity XML schema 196
Table 10.37 – GlobalBusinessRuleTask XML schema 196
Table 10.38 – GlobalScriptTask XML schema 197
Table 10.39 – GlobalTask XML schema 197
Table 10.40 – LoopCharacteristics XML schema 197
Table 10.41 – MultiInstanceLoopCharacteristics XML schema 198
Table 10.42 – ReceiveTask XML schema 199
Table 10.43 – ResourceRole XML schema 199
Table 10.44 – ScriptTask XML schema 200
Table 10.45 – SendTask XML schema 200
Table 10.46 – ServiceTask XML schema 200
Table 10.47 – StandardLoopCharacteristics XML schema 201
Table 10.48 – SubProcess XML schema 201
Table 10.49 – Task XML schema 201
Table 10.50 – Transaction XML schema 202
Table 10.51 – ItemAwareElement model associations 203
Table 10.52 – DataObject attributes 205
Table 10.53 – DataObjectReference attributes and model associations 205
Table 10.54 – DataState attributes and model associations 205
Table 10.55 – Data Store attributes 208
Table 10.56 – Data Store attributes 208
Table 10.57 – Property attributes 209
Table 10.58 – InputOutputSpecification Attributes and Model Associations 212
Table 10.59 – DataInput attributes and model associations 214
Table 10.60 – DataOutput attributes and associations 216
Table 10.61 – InputSet attributes and model associations 218
Table 10.62 – OutputSet attributes and model associations 220
Table 10.63 – DataAssociation model associations 222
Table 10.64 – Assignment attributes 223
Table 10.65 – XPath Extension Function for Data Objects 226
Table 10.66 – XPath Extension Function for Data Inputs and Data Outputs 226
Table 10.67 – XPath Extension Functions for Properties 227
Table 10.68 – XPath extension functions for instance attributes 228
Table 10.69 – Assignment XML schema 228
Table 10.70 – DataAssociation XML schema 229
Table 10.71 – DataInput XML schema 229
Table 10.72 – DataInputAssociation XML schema 229
Table 10.73 – DataObject XML schema 230
Table 10.74 – DataState XML schema 230
Table 10.75 – DataOutput XML schema 230

Table 10.76 – DataOutputAssociation XML schema 230
 Table 10.77 – InputOutputSpecification XML schema 231
 Table 10.78 – InputSet XML schema 231
 Table 10.79 – OutputSet XML schema 232
 Table 10.80 – Property XML schema 232
 Table 10.81 – Event model associations 235
 Table 10.82 – CatchEvent attributes and model associations 235
 Table 10.83 – ThrowEvent attributes and model associations 236
 Table 10.84 – Top-Level Process Start Event Types 239
 Table 10.85 – Sub-Process Start Event Types 241
 Table 10.86 – Event Sub-Process Start Event Types 241
 Table 10.87 – Start Event attributes 244
 Table 10.88 – End Event Types 246
 Table 10.89 – Intermediate Event Types in Normal Flow 250
 Table 10.90 – Intermediate Event Types Attached to an Activity Boundary 253
 Table 10.91 – Boundary Event attributes 257
 Table 10.92 – Possible Values of the cancelActivity Attribute 257
 Table 10.93 – Types of Events and their Markers 260
 Table 10.94 – CompensationEventDefinition attributes and model associations 263
 Table 10.95 – ConditionalEventDefinition model associations 264
 Table 10.96 – ErrorEventDefinition attributes and model associations 265
 Table 10.97 – EscalationEventDefinition attributes and model associations 266
 Table 10.98 – LinkEventDefinition attributes 269
 Table 10.99 – MessageEventDefinition model associations 270
 Table 10.100 – SignalEventDefinition model associations 272
 Table 10.101 – TimerEventDefinition model associations 273
 Table 10.102 – BoundaryEvent XML schema 281
 Table 10.103 – CancelEventDefinition XML schema 281
 Table 10.104 – CatchEvent XML schema 281
 Table 10.105 – CancelEventDefinition XML schema 281
 Table 10.106 – CompensateEventDefinition XML schema 282
 Table 10.107 – ConditionalEventDefinition XML schema 282
 Table 10.108 – ErrorEventDefinition XML schema 282
 Table 10.109 – EscalationEventDefinition XML schema 282
 Table 10.110 – Event XML schema 283
 Table 10.111 – EventDefinition XML schema 283
 Table 10.112 – ImplicitThrowEvent XML schema 283
 Table 10.113 – IntermediateCatchEvent XML schema 283
 Table 10.114 – IntermediateThrowEvent XML schema 283
 Table 10.115 – LinkEventDefinition XML schema 283
 Table 10.116 – MessageEventDefinition XML schema 284
 Table 10.117 – Signal XML schema 284
 Table 10.118 – SignalEventDefinition XML schema 284
 Table 10.119 – StartEvent XML schema 285
 Table 10.120 – TerminateEventDefinition XML schema 285
 Table 10.121 – ThrowEvent XML schema 285
 Table 10.122 – TimerEventDefinition XML schema 285

Table 10.123 – ExclusiveGateway Attributes & Model Associations 291
Table 10.124 – InclusiveGateway Attributes & Model Associations 292
Table 10.125 – Complex Gateway model associations 295
Table 10.126 – Instance attributes related to the Complex Gateway 296
Table 10.127 – EventBasedGateway Attributes & Model Associations 299
Table 10.128 – ComplexGateway XML schema 300
Table 10.129 – EventBasedGateway XML schema 300
Table 10.130 – ExclusiveGateway XML schema 300
Table 10.131 – Gateway XML schema 300
Table 10.132 – InclusiveGateway XML schema 301
Table 10.133 – ParallelGateway XML schema 301
Table 10.134 – LaneSet attributes and model associations 307
Table 10.135 – Lane attributes and model associations 308
Table 10.136 – Process XML schema 311
Table 10.137 – Auditing XML schema 312
Table 10.138 – GlobalTask XML schema 312
Table 10.139 – Lane XML schema 312
Table 10.140 – LaneSet XML schema 312
Table 10.141 – Monitoring XML schema 313
Table 10.142 – Performer XML schema 313
Table 11.1 – Choreography Activity Model Associations 322
Table 11.2 – Choreography Task Model Associations 328
Table 11.3 – Sub-Choreography Model Associations 332
Table 11.4 – Call Choreography Model Associations 335
Table 11.5 – Global Choreography Task Model Associations 335
Table 11.6 – Use of Start Events in Choreography 340
Table 11.7 – Use of Intermediate Events in Choreography 340
Table 11.8 – Use of End Events in Choreography 343
Table 11.9 – Choreography XML schema 363
Table 11.10 – GlobalChoreographyTask XML schema 364
Table 11.11 – ChoreographyActivity XML schema 364
Table 11.12 – ChoreographyTask XML schema 364
Table 11.13 – CallChoreography XML schema 365
Table 11.14 – SubChoreography XML schema 365
Table 12.1 – BPMNDiagram XML schema 371
Table 12.2 – BPMNPlane XML schema 372
Table 12.3 – BPMNShape XML schema 374
Table 12.4 – BPMNEdge XML schema 376
Table 12.5 – BPMNLabel XML schema 377
Table 12.6 – BPMNLabelStyle XML schema 378
Table 12.7 – Complete BPMN DI XML schema 378
Table 12.8 – Depiction Resolution for Loop Compensation Marker 382
Table 12.9 – Depiction Resolution for Tasks 385
Table 12.10 – Depiction Resolution for Collapsed Sub-Processes 386
Table 12.11 – Depiction Resolution for Expanded Sub-Processes 386
Table 12.12 – Depiction Resolution for Collapsed Ad Hoc Sub-Processes 387
Table 12.13 – Depiction Resolution for Expanded Ad Hoc Sub-Processes 387

Table 12.14 – Depiction Resolution for Collapsed Transactions 387
 Table 12.15 – Depiction Resolution for Tasks 388
 Table 12.16 – Depiction Resolution for Collapsed Event Sub-Processes 388
 Table 12.17 – Depiction Resolution for Expanded Event Sub-Processes 391
 Table 12.18 – Depiction Resolution for Call Activities (Calling a Global Task) 391
 Table 12.19 – Depiction Resolution for Collapsed Call Activities (Calling a Process) 392
 Table 12.20 – Depiction Resolution for Expanded Call Activities (Calling a Process) 392
 Table 12.21 – Depiction Resolution for Data 393
 Table 12.22 – Depiction Resolution for Events 394
 Table 12.23 – Depiction Resolution for Gateways 400
 Table 12.24 – Depiction Resolution for Artifacts 401
 Table 12.25 – Depiction Resolution for Lanes 401
 Table 12.26 – Depiction Resolution for Pools 402
 Table 12.27 – Depiction Resolution for Choreography Tasks 403
 Table 12.28 – Depiction Resolution for Sub-Choreographies (Collapsed) 404
 Table 12.29 – Depiction Resolution for Sub-Choreographies (Expanded) 405
 Table 12.30 – Depiction Resolution for Call Choreographies (Calling a Global Choreography Task) 405
 Table 12.31 – Depiction Resolution for Collapsed Call Choreographies (Calling a Choreography) 406
 Table 12.32 – Depiction Resolution for Expanded Call Choreographies (Calling a Choreography) 407
 Table 12.33 – Depiction Resolution for Choreography Participant Bands 408
 Table 12.34 – Depiction Resolution for Conversations 410
 Table 12.35 – Depiction Resolution for Connecting Objects 411
 Table 12.36 – Expanded Sub-Process BPMN DI instance 413
 Table 12.37 – Start and End Events on the Border BPMN DI instance 414
 Table 12.38 – Collapsed Sub-Process BPMN DI instance 416
 Table 12.39 – Sub-Process Content BPMN DI instance 416
 Table 12.40 – Multiple Lanes and Nested Lanes BPMN DI instance 417
 Table 12.41 – Vertical Collaboration BPMN DI instance 418
 Table 12.42 – Conversation BPMN DI instance 420
 Table 12.43 – Choreography BPMN DI instance 422
 Table 13.1 – Parallel Gateway Execution Semantics 434
 Table 13.2 – Exclusive Gateway Execution Semantics 435
 Table 13.3 – Inclusive Gateway Execution Semantics 436
 Table 13.4 – Event-Based Gateway Execution Semantics 437
 Table 13.5 – Semantics of the Complex Gateway 438
 Table 14.1 – Common Activity Mappings to WS-BPEL 448
 Table 14.2 – Expressions mapping to WS-BPEL 468

(Blank page)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19510 was prepared by the Object Management Group (OMG), as the OMG Business Process Model and Notation (BPMN), version 2.0.1, and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Introduction

The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation.

This specification represents the amalgamation of best practices within the business modeling community to define the notation and semantics of Collaboration diagrams, Process diagrams, and Choreography diagrams. The intent of BPMN is to standardize a business process model and notation in the face of many different modeling notations and viewpoints. In doing so, BPMN will provide a simple means of communicating process information to other business users, process implementers, customers, and suppliers.

The specification provided by this International Standard is identical to OMG BPMN 2.0.1.

Information technology - Object Management Group Business Process Model and Notation

1 Scope

1.1 General

The **Object Management Group** (OMG) has developed a standard **Business Process Model and Notation (BPMN)**. The primary goal of **BPMN** is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. Thus, **BPMN** creates a standardized bridge for the gap between the business process design and process implementation.

Another goal, but no less important, is to ensure that XML languages designed for the execution of business processes, such as **WSBPEL** (Web Services Business Process Execution Language), can be visualized with a business-oriented notation.

This International Standard represents the amalgamation of best practices within the business modeling community to define the notation and semantics of **Collaboration** diagrams, **Process** diagrams, and **Choreography** diagrams. The intent of **BPMN** is to standardize a business process model and notation in the face of many different modeling notations and viewpoints. In doing so, **BPMN** will provide a simple means of communicating process information to other business users, process implementers, customers, and suppliers.

The membership of the OMG has brought forth expertise and experience with many existing notations and has sought to consolidate the best ideas from these divergent notations into a single standard notation. Examples of other notations or methodologies that were reviewed are UML Activity Diagram, UML EDOC Business Processes, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVEM, and Event-Process Chains (EPCs).

2 Conformance

2.1 General

Software can claim compliance or conformance with **BPMN 2.0** if and only if the software fully matches the applicable compliance points as stated in the International Standard. Software developed only partially matching the applicable compliance points can claim only that the software was based on this International Standard, but cannot claim compliance or conformance with this International Standard. The International Standard defines four types of conformance namely **Process Modeling Conformance**, **Process Execution Conformance**, **BPEL Process Execution Conformance**, and **Choreography Modeling Conformance**.

The implementation claiming conformance to **Process Modeling Conformance** type is NOT REQUIRED to support **Choreography Modeling Conformance** type and vice-versa. Similarly, the implementation claiming **Process Execution Conformance** type is NOT REQUIRED to be conformant to the **Process Modeling** and **Choreography Conformance** types.

The implementation claiming conformance to the **Process Modeling Conformance type** SHALL comply with all of the requirements set forth in sub clause 2.1. The implementation claiming conformance to the **Process Execution Conformance type** SHALL comply with all of the requirements set forth in sub clause 2.2. The implementation claiming conformance to the **BPEL Process Execution Semantics Conformance type** SHALL comply with all of the requirements set forth in sub clause 2.3. The implementation claiming conformance to the **Choreography Conformance type** SHALL comply with all of the requirements set forth in sub clause 2.4. The implementation is said to have **BPMN Complete Conformance** if it complies with all of the requirements stated in sub clauses 2.1, 2.2, 2.3, and 2.4.

2.2 Process Modeling Conformance

The next eight sub clauses describe **Process Modeling Conformance**.

2.2.1 BPMN Process Types

The implementations claiming **Process Modeling Conformance** MUST support the following **BPMN** packages:

- ◆ The **BPMN** core elements, which include those defined in the *Infrastructure*, *Foundation*, *Common*, and *Service* packages (see Clause 8).
- ◆ **Process** diagrams, which include the elements defined in the **Process**, **Activities**, **Data**, and **Human Interaction** packages (see Clause 10).
- ◆ **Collaboration** diagrams, which include **Pools** and **Message Flow** (see Clause 9).
- ◆ **Conversation** diagrams, which include **Pools**, **Conversations**, and **Conversation Links** (see Clause 9).

As an alternative to full **Process Modeling Conformance**, there are three conformance sub-classes defined:

- ◆ **Descriptive**
- ◆ **Analytic**
- ◆ **Common Executable**

Descriptive is concerned with visible elements and attributes used in high-level modeling. It should be comfortable for analysts who have used BPA flowcharting tools.

Analytic contains all of **Descriptive** and in total about half of the constructs in the full **Process Modeling Conformance** Class. It is based on experience gathered in BPMN training and an analysis of user-patterns in the Department of Defense Architecture Framework and planned standardization for that framework.

Both **Descriptive** and **Analytic** focus on visible elements and a minimal subset of supporting attributes/elements.

Common Executable focuses on what is needed for executable process models.

Elements and attributes not in these sub-classes are contained in the full **Process Modeling Conformance** class.

The elements for each sub-class are defined in the next sub clause.

2.2.2 BPMN Process Elements

The **Process Modeling Conformance** type set consists of **Collaboration** and **Process** diagram elements, including all **Task** types, *embedded Sub-Processes*, **CallActivity**, all **Gateway** types, all **Event** types (**Start**, **Intermediate**, and **End**), **Lane**, **Participants**, **Data Object** (including **DataInput** and **DataOutput**), **Message**, **Group**, **Text Annotation**, **Sequence Flow** (including *conditional* and *default* flows), **Message Flow**, **Conversations** (limited to grouping **Message Flow**, and associating *correlations*), **Correlation**, and **Association** (including **Compensation Association**). The set also includes markers (**Loop**, **Multi-Instance**, **Transaction**, **Compensation**) for **Tasks** and *embedded Sub-Processes*).

NOTE: Implementations are not expected to support **Choreography** modeling elements such as **Choreography Task** and **Sub-Choreography**.

For a tool to claim support for a sub-class the following criteria MUST be satisfied:

- ◆ All the elements in the sub-class MUST be supported.
- ◆ For each element, all the listed attributes MUST be supported.
- ◆ In general, if the sub-class doesn't mention an attribute and it is NOT REQUIRED by the schema, then it is not in the subclass. Exceptions to this rule are noted.

Descriptive Conformance Sub-Class

The **Descriptive** conformance sub-class elements are shown in Table 2.1.

Table 2.1 – Descriptive Conformance Sub-Class Elements and Attributes

Element	Attributes
participant (pool)	id, name, processRef
laneSet	id, lane with name, childLaneSet, flowElementRef
sequenceFlow (unconditional)	id, name, sourceRef, targetRef
messageFlow	id, name, sourceRef, targetRef
exclusiveGateway	id, name
parallelGateway	id, name
task (None)	id, name
userTask	id, name
serviceTask	id, name
subProcess (expanded)	id, name, flowElement
subProcess (collapsed)	id, name, flowElement
CallActivity	id, name, calledElement
DataObject	id, name

Table 2.1 – Descriptive Conformance Sub-Class Elements and Attributes

TextAnnotation	id, text
association/dataAssociation ^a	id, name, sourceRef, targetRef, associationDirection ^b
dataStoreReference	id, name, dataStoreRef
startEvent (None)	id, name
endEvent (None)	id, name
messageStartEvent	id, name, messageEventDefinition
messageEndEvent	id, name, messageEventDefinition
timerStartEvent	id, name, timerEventDefinition
terminateEndEvent	id, name, terminateEventDefinition
documentation ^c	text
Group	id, categoryRef

- a. **Data Association** is ABSTRACT: **Data Input Association** and **Data Output Association** will appear in the XML serialization. These both have REQUIRED attributes [sourceRef and targetRef] which refer to itemAwareElements. To be consistent with the metamodel, this will require the following additional elements: ioSpecification, inputSet, outputSet, **Data Input**, **Data Output**. When a **BPMN** editor draws a **Data Association** to an **Activity** or **Event** it should generate this supporting invisible substructure. Otherwise, the metamodel would have to be changed to make sourceRef and targetRef optional or allow reference to non-itemAwareElements, e.g., **Activity** and **Event**.
- b. associationDirection not specified for **Data Association**
- c. Documentation is not a visible element. It is an attribute of most elements.

Analytic Conformance Sub-Class

The **Analytic** conformance sub-class contains all the elements of the **Descriptive** conformance sub-class plus the elements shown in Table 2.2.

Table 2.2 – Analytic Conformance Sub-Class Elements and Attributes

Element	Attributes
sequenceFlow (conditional)	id, name, sourceRef, targetRef, conditionExpression ^a
sequenceFlow (default)	id, name, sourceRef, targetRef, default ^b
sendTask	id, name
receiveTask	id, name
Looping Activity	standardLoopCharacteristics
MultInstance Activity	multilnstanceLoopCharacteristics

Table 2.2 – Analytic Conformance Sub-Class Elements and Attributes

exclusiveGateway	Add default attribute
inclusiveGateway	id, name, eventGatewayType
eventBasedGateway	id, name, eventGatewayType
Link catch/throw Intermediate Event	Id, name, linkEventDefinition
signalStartEvent	id, name, signalEventDefinition
signalEndEvent	id, name, signalEventDefinition
Catching message Intermediate Event	id, name, messageEventDefinition
Throwing message Intermediate Event	id, name, messageEventDefinition
Boundary message Intermediate Event	id, name, attachedToRef, messageEventDefinition
Non-interrupting Boundary message Intermediate Event	id, name, attachedToRef, cancelActivity=false, messageEventDefinition
Catching timer Intermediate Event	id, name, timerEventDefinition
Boundary timer Intermediate Event	id, name, attachedToRef, timerEventDefinition
Non-interrupting Boundary timer Intermediate Event	id, name, attachedToRef, cancelActivity=false, timerEventDefinition
Boundary error Intermediate Event	id, name, attachedToRef, errorEventDefinition
errorEndEvent	id, name, errorEventDefinition
Non-interrupting Boundary escalation Intermediate Event	id, name, attachedToRef, cancelActivity=false, escalationEventDefinition
Throwing escalation Intermediate Event	id, name, escalationEventDefinition
escalationEndEvent	id, name, escalationEventDefinition
Catching signal Intermediate Event	id, name, signalEventDefinition
Throwing signal Intermediate Event	id, name, signalEventDefinition
Boundary signal Intermediate Event	id, name, attachedToRef, signalEventDefinition
Non-interrupting Boundary signal Intermediate Event	id, name, attachedToRef, cancelActivity=false, signalEventDefinition
conditionalStartEvent	id, name, conditionalEventDefinition

Table 2.2 – Analytic Conformance Sub-Class Elements and Attributes

Catching conditional Intermediate Event	id, name, conditionalEventDefinition
Boundary conditional Intermediate Event	id, name, conditionalEventDefinition
Non-interrupting Boundary conditional Intermediate Event	id, name, cancelActivity=false, conditionalEventDefinition
message ^c	id, name, add messageRef attribute to messageFlow

- a. ConditionExpression, allowed only for **Sequence Flow** out of **Gateways**, MAY be null.
- b. Default is an attribute of a sourceRef (exclusive or inclusive) **Gateway**.
- c. Note that messageRef, an attribute of various message **Events**, is optional and not in the sub-class.

Common Executable Conformance Sub-Class

This conformance sub-class is intended for modeling tools that can emit executable models.

- ◆ Data type definition language MUST be XML Schema.
- ◆ Service Interface definition language MUST be WSDL.
- ◆ Data access language MUST be XPath.

The **Common Executable** conformance sub-class elements are shown in Table 2.3 and its supporting classes in Table 2.4.

Table 2.3 – Common Executable Conformance Sub-Class Elements and Attributes

Element	Attributes
sequenceFlow (unconditional)	id, (name), sourceRef ^a , targetRef ^b
sequenceFlow (conditional)	id, name, sourceRef, targetRef, conditionExpression ^c
sequenceFlow (default)	id, name, sourceRef, targetRef, default ^d
subProcess (expanded)	id, name, flowElement, loopCharacteristics, boundaryEventRefs
exclusiveGateway	id, name, gatewayDirection (only converging and diverging), default
parallelGateway	id, name, gatewayDirection (only converging and diverging)
startEvent (None)	id, name
endEvent (None)	id, name
eventBasedGateway	id, name, gatewayDirection, eventGatewayType
userTask	id, name, renderings, implementation, resources, ioSpecification, dataInputAssociations, dataOutputAssociations, loopCharacteristics, boundaryEventRefs

Table 2.3 – Common Executable Conformance Sub-Class Elements and Attributes

serviceTask	id, name, implementation, operationRef, ioSpecification, dataInputAssociations, dataOutputAssociations, loopCharacteristics, boundaryEventRefs
callActivity	id, name, calledElement, ioSpecification, dataInputAssociations, dataOutputAssociations, loopCharacteristics, boundaryEventRefs
dataObject	id, name, isCollection, itemSubjectRef
textAnnotation	id, text
dataAssociation	id, name, sourceRef, targetRef, assignment
messageStartEvent	id, name, messageEventDefinition (either ref or contained), dataOutput, dataOutputAssociations
messageEndEvent	id, name, messageEventDefinition, (either ref or contained), dataInput, dataInputAssociations
terminateEndEvent	(Terminating trigger in combination with one of the other end events)
Catching message Intermediate Event	id, name, messageEventDefinition (either ref or contained), dataOutput, dataOutputAssociations
Throwing message Intermediate Event	id, name, messageEventDefinition (either ref or contained), dataInput, dataInputAssociations
Catching timer Intermediate Event	id, name, timerEventDefinition (contained)
Boundary error Intermediate Event	id, name, attachedToRef, errorEventDefinition, (contained or referenced), dataOutput, dataOutputAssociations

- a. Multiple outgoing connections are only allowed for converging **Gateways**.
- b. Multiple outgoing connections are only allowed for diverging **Gateways**.
- c. ConditionExpression, allowed only for **Sequence Flow** out of **Gateways**, MAY be null.
- d. Default is an attribute of a sourceRef (exclusive or inclusive) **Gateway**.

Table 2.4 – Common Executable Conformance Sub-Class Supporting Classes

Element	Attributes
StandardLoopCharacteristics	id, loopCondition
MultiInstanceLoopCharacteristics	id, isSequential, loopDataInput, inputDataItem
Rendering	
Resource	id, name
ResourceRole	id, resourceRef, resourceAssignmentExpression
InputOutputSpecification	id, dataInputs, dataOutputs
DataInput	id, name, isCollection, itemSubjectRef

Table 2.4 – Common Executable Conformance Sub-Class Supporting Classes

DataOutput	id, name, isCollection, itemSubjectRef
ItemDefinition	id, structure or import ^a
Operation	id, name, inMessageRef, outMessageRef, errorRefs
Message	id, name, structureRef
Error	id, structureRef
Assignment	id, from, to ^b
MessageEventDefinition	id, messageRef, operationRef
TerminateEventDefinition	id
TimerEventDefinition	id, timeDate

a. Structure MUST be defined by an XSD Complex Type

b. Structure MUST be defined by an XSD Complex Type

2.2.3 Visual Appearance

A key element of **BPMN** is the choice of shapes and icons used for the graphical elements identified in this International Standard. The intent is to create a standard visual language that all process modelers will recognize and understand. An implementation that creates and displays **BPMN Process** Diagrams SHALL use the graphical elements, shapes, and markers illustrated in this International Standard.

NOTE: There is flexibility in the size, color, line style, and text positions of the defined graphical elements, except where otherwise specified (see page 41).

The following extensions to a **BPMN** Diagram are permitted:

- ◆ New markers or indicators MAY be added to the specified graphical elements. These markers or indicators could be used to highlight a specific attribute of a **BPMN** element or to represent a new subtype of the corresponding concept.
- ◆ A new shape representing a kind of **Artifact** MAY be added to a Diagram, but the new **Artifact** shape SHALL NOT conflict with the shape specified for any other **BPMN** element or marker.
- ◆ Graphical elements MAY be colored, and the coloring MAY have specified semantics that extend the information conveyed by the element as specified in this International Standard.
- ◆ The line style of a graphical element MAY be changed, but that change SHALL NOT conflict with any other line style REQUIRED by this International Standard.
- ◆ An extension SHALL NOT change the specified shape of a defined graphical element or marker (e.g., changing a square into a triangle, or changing rounded corners into squared corners, etc.).

2.2.4 Structural Conformance

An implementation that creates and displays **BPMN** diagrams SHALL conform to the specifications and restrictions with respect to the connections and other diagrammatic relationships between graphical elements. Where permitted or requested connections are specified as conditional and based on attributes of the corresponding concepts, the implementation SHALL ensure the correspondence between the connections and the values of those attributes.

NOTE: In general, these connections and relationships have specified semantic interpretations, which specify interactions among the process concepts represented by the graphical elements. Conditional relationships based on attributes represent specific variations in behavior. Structural conformance therefore guarantees the correct interpretation of the diagram as a specification of process, in terms of flows of control and information. Throughout the document, structural specifications will appear in paragraphs using a special shaped bullet: Example: ♦ A **TASK** MAY be a target for **Sequence Flow**; it can have multiple *incoming* Flows. An *incoming* Flow MAY be from an alternative path and/or parallel paths.

2.2.5 Process Semantics

This International Standard defines many semantic concepts used in defining **Processes**, and associates them with graphical elements, markers, and connections. To the extent that an implementation provides an interpretation of the **BPMN** diagram as a semantic specification of **Process**, the interpretation SHALL be consistent with the semantic interpretation herein specified. In other words, the implementation claiming **BPMN Process Modeling Conformance** has to support the semantics surrounding the diagram elements expressed in Clause 10.

NOTE: The implementations claiming **Process Modeling Conformance** are not expected to support the **BPMN** execution semantics described in Clause 13.

2.2.6 Attributes and Model Associations

This International Standard defines a number of attributes and properties of the semantic elements represented by the graphical elements, markers, and connections. Some of these attributes are purely representational and are so marked, and some have mandated representations. Some attributes are specified as mandatory, but have no representation or only optional representation. And some attributes are specified as optional. For every attribute or property that is specified as mandatory, a conforming implementation SHALL provide some mechanism by which values of that attribute or property can be created and displayed. This mechanism SHALL permit the user to create or view these values for each **BPMN** element specified to have that attribute or property. Where a graphical representation for that attribute or property is specified as REQUIRED, that graphical representation SHALL be used. Where a graphical representation for that attribute or property is specified as optional, the implementation MAY use either a graphical representation or some other mechanism. If a graphical representation is used, it SHALL be the representation specified. Where no graphical representation for that attribute or property is specified, the implementation MAY use either a graphical representation or some other mechanism. If a graphical representation is used, it SHALL NOT conflict with the specified graphical representation of any other **BPMN** element.

2.2.7 Extended and Optional Elements

A conforming implementation is NOT REQUIRED to support any element or attribute that is specified herein to be non-normative or informative. In each instance in which this International Standard defines a feature to be “optional,” it specifies whether the option is in:

- how the feature will be displayed,
- whether the feature will be displayed,
- whether the feature will be supported.

A conforming implementation is NOT REQUIRED to support any feature whose support is specified to be optional. If an implementation supports an optional feature, it SHALL support it as specified. A conforming implementation SHALL support any “optional” feature for which the option is only in whether or how it SHALL be displayed.

2.2.8 Visual Interchange

One of the main goals of this International Standard is to provide an interchange format that can be used to exchange **BPMN** definitions (both domain model and diagram layout) between different tools. The implementation should support the metamodel for **Process** types specified in sub clause 13.1 to enable portability of process diagrams so that users can take business process definitions created in one vendor's environment and use them in another vendor's environment.

2.3 Process Execution Conformance

The next two sub clauses describe **Process Execution Conformance**.

2.3.1 Execution Semantics

The **BPMN** execution semantics have been fully formalized in this version of the International Standard. The tool claiming **BPMN Execution Conformance** type MUST fully support and interpret the operational semantics and **Activity** life-cycle specified in sub clause 14.2.2. Non-operational elements listed in Clause 14 MAY be ignored by implementations claiming **BPMN Execution Conformance** type. Conformant implementations MUST fully support and interpret the underlying metamodel.

NOTE: The tool claiming **Process Execution Conformance type** is not expected to support and interpret **Choreography** models. The tool claiming **Process Execution Conformance type** is not expected to support **Process Modeling Conformance**. More precisely, the tool is not required to support graphical syntax and semantics defined in this International Standard. It MAY use different graphical elements, shapes and markers, than those defined in this International Standard.

2.3.2 Import of Process Diagrams

The tool claiming **Process Execution Conformance type** MUST support import of **BPMN Process** diagram types including its definitional **Collaboration** (see Table 10.1).

2.4 BPEL Process Execution Conformance

Special type of Process Execution Conformance that supports the **BPMN** mapping to WS-BPEL as specified in sub clause 15.1 can claim **BPEL Process Execution Conformance**.

NOTE: The tool claiming **BPEL Process Execution Conformance** MUST fully support **Process Execution Conformance**. The tool claiming **BPEL Process Execution Conformance** is not expected to support and interpret **Choreography** models. The tool claiming **BPEL Process Execution Conformance** is not expected to support **Process Modeling Conformance**.

2.5 Choreography Modeling Conformance

The next five sub clauses describe **Choreography Conformance**.

2.5.1 BPMN Choreography Types

The implementations claiming **Choreography Conformance** type MUST support the following **BPMN** packages:

- ◆ The **BPMN** core elements, which include those defined in the Infrastructure, Foundation, Common, and Service packages (see Clause 8).

- ◆ **Choreography** diagrams, which includes the elements defined in the **Choreography**, and **Choreography** packages (see Clause 11).
- ◆ **Collaboration** diagrams, which include **Pools** and **Message Flow** (see Clause 9).

2.5.2 BPMN Choreography Elements

The **Choreography Conformance** set includes **Message**, **Choreography Task**, **Global Choreography Task**, **Sub-Choreography** (expanded and collapsed), certain types of **Start Events** (e.g., **None**, **Timer**, **Conditional**, **Signal**, and **Multiple**), certain types of **Intermediate Events** (**None**, **Message**) attached to **Activity** boundary, **Timer** – normal as well as attached to **Activity** boundary, **Timer** used in **Event Gateways**, **Cancel** attached to an **Activity** boundary, **Conditional**, **Signal**, **Multiple**, **Link**, etc.) and certain types of **End Events** (**None** and **Terminate**), and **Gateways**. In addition, to enable **Choreography** within **Collaboration** it should support **Pools** and **Message Flow**.

2.5.3 Visual Appearance

An implementation that creates and displays **BPMN Choreography** Diagrams SHALL use the graphical elements, shapes, and markers as specified in the **BPMN** International Standard. The use of text, color, size and lines for **Choreography** diagram types are listed in sub clause 7.4.

2.5.4 Choreography Semantics

The tool claiming **Choreography Conformance** should fully support and interpret the graphical and execution semantics surrounding **Choreography** diagram elements and **Choreography** diagram types.

2.5.5 Visual Interchange

The implementation should support import/export of **Choreography** diagram types and **Collaboration** diagram types that depict **Choreography** within **collaboration** as specified in sub clause 9.4 to enable portability of **Choreography** definitions, so that users can take **BPMN** definitions created in one vendor's environment and use them in another vendor's environment.

2.6 Summary of BPMN Conformance Types

Table 2.5 summarizes the requirements for **BPMN** Conformance.

Table 2.5 – Types of BPMN Conformance

Category	Process Modeling Conformance	Process Execution Conformance	BPEL Process Execution Conformance	Choreography Conformance
Visual representation of BPMN Diagram Types	Process diagram types and Collaboration diagram types depicting collaborations among Process diagram types.	N/A	N/A	Choreography diagram types and Collaboration diagram types depicting collaboration among Choreography diagram types.
BPMN Diagram Elements that need to be supported.	All Task types, embedded Sub-Process, Call Activity, all Event types, all Gateway types, Pool, Lane, Data Object (including DataInput and DataOutput), Message, Group, Artifacts, markers for Tasks and Sub-Processes, Sequence Flow, Associations, and Message Flow.	N/A	N/A	Message, Choreography Task, Global Choreography Task, Sub-Choreography (expanded and collapsed), certain types of Start, Intermediate, and End Events, Gateways, Pools and Message Flow.
Import/Export of diagram types	Yes for Process and Collaboration diagrams that depict Process within Collaboration.	Yes for Process diagrams	Yes for Process diagrams	Yes for Choreography and Collaboration diagrams depicting choreography within Collaboration.
Support for Graphical syntax and semantics	Process and Collaboration diagrams that depict Process within Collaboration.	N/A	N/A	Choreography and Collaboration diagrams depicting Choreography within Collaboration.
Support for Execution Semantics	N/A	Yes for Process diagrams	Yes for Process diagrams	Choreography execution semantics

3 Normative References

3.1 General

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

3.2 Normative

OMG UML

- OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2 -
<http://www.omg.org/spec/UML/2.1.2/Superstructure>

OMG MOF

- Object Management Group - Meta Object Facility (MOF) Core Specification, V2.0
<http://www.omg.org/spec/MOF/2.0>

RFC-2119

- Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, IETF RFC 2119, March 1997
<http://www.ietf.org/rfc/rfc2119.txt>

3.3 Non-Normative

Activity Service

- Additional Structuring Mechanism for the OTS Specification, OMG, June 1999
<http://www.omg.org>
- J2EE Activity Service for Extended Transactions (JSR 95), JCP
<http://www.jcp.org/jsr/detail/95.jsp>

BPEL4People

- WS-BPEL Extension for People (BPEL4People) Specification Version 1.1, Committee Specification, 17 August 2010
<http://docs.oasis-open.org/bpel4people/bpel4people-1.1-spec-cs-01.html>

Business Process Definition Metamodel

- OMG, May 2008,
<http://www.omg.org/docs/dtc/08-05-07.pdf>

Business Process Modeling

- Jean-Jacques Dubray, “A Novel Approach for Modeling Business Process Definitions,” 2002
<http://www.ebpml.org/ebpml2.2.doc>

Business Transaction Protocol

- OASIS BTP Technical Committee, June, 2002
http://www.oasis-open.org/committees/download.php/1184/2002-06-03.BTP_ctee_spec_1.0.pdf

Dublin Core Meta Data

- Dublin Core Metadata Element Set, Dublin Core Metadata Initiative
<http://dublincore.org/documents/dces/>

ebXML BPSS

- Jean-Jacques Dubray, “A new model for ebXML BPSS Multi-party Collaborations and Web Services Choreography,” 2002
<http://www.ebpml.org/ebpml.doc>

Open Nested Transactions

- Concepts and Applications of Multilevel Transactions and Open Nested Transactions, Gerhard Weikum, Hans-J. Schek, 1992
<http://citeseer.nj.nec.com/weikum92concepts.html>

RDF

- RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft
<http://www.w3.org/TR/rdf-schema/>

SOAP 1.2

- SOAP Version 1.2 Part 1: Messaging Framework, W3C Working Draft
<http://www.w3.org/TR/soap12-part1/>
- SOAP Version 1.2 Part21: Adjuncts, W3C Working Draft
<http://www.w3.org/TR/soap12-part2/>

UDDI

- Universal Description, Discovery and Integration, Ariba, IBM and Microsoft, UDDI.org.
<http://www.uddi.org>

URI

- Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, IETF RFC 2396, August 1998
<http://www.ietf.org/rfc/rfc2396.txt>

WfMC Glossary

- Workflow Management Coalition Terminology and Glossary
<http://www.wfmc.org/wfmc-standards-framework.html>

Web Services Transaction

- (WS-Transaction) 1.1, OASIS, 12 July 2007,
<http://www.oasis-open.org/committees/ws-tx/>

Workflow Patterns

- Russell, N., ter Hofstede, A.H.M., van der Aalst W.M.P., & Mulyar, N. (2006). Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-22, BPMcentre.org
<http://www.workflowpatterns.com/>

WSBPEL

- Web Services Business Process Execution Language (WSBPEL) 2.0, OASIS Standard, April 2007
<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

WS-Coordination

- Web Services Coordination (WS-Coordination) 1.1, OASIS Standard, July 2007
<http://www.oasis-open.org/committees/ws-tx/>

WSDL

- Web Services Description Language (WSDL) 2.0, W3C Proposed Recommendation, June 2007
<http://www.w3.org/TR/wsdl20/>

WS-HumanTask

- Web Services Human Task (WS-HumanTask) 1.1, August 2010
<http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html>

XML 1.0 (Second Edition)

- Extensible Markup Language (XML) 1.0, Second Edition, Tim Bray et al., eds., W3C, 6 October 2000
<http://www.w3.org/TR/REC-xml>

XML-Namespace

- Namespaces in XML, Tim Bray et al., eds., W3C, 14 January 1999
<http://www.w3.org/TR/REC-xml-names>

XML-Schema

- XML Schema Part 1: Structures, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, W3C, 2 May 2001
<http://www.w3.org/TR/xmlschema-1/>
- XML Schema Part 2: Datatypes, Paul V. Biron and Ashok Malhotra, eds., W3C, 2 May 2001
<http://www.w3.org/TR/xmlschema-2/>

XPath

- XML Path Language (XPath) 1.0, James Clark and Steve DeRose, eds., W3C, 16 November 1999
<http://www.w3.org/TR/xpath>

XPDL

- Workflow Management Coalition XML Process Definition Language, version 2.0.
<http://www.wfmc.org/wfmc-standards-framework.html>