
Contents

Part I Conventional Middleware

1	Distributed Information Systems	3
1.1	Design of an Information System	4
1.1.1	Layers of an Information System	4
1.1.2	Top-down Design of an Information System.....	6
1.1.3	Bottom-up Design of an Information System	7
1.2	Architecture of an Information System	9
1.2.1	One-tier Architectures	10
1.2.2	Two-tier Architectures	12
1.2.3	Three-tier Architectures	16
1.2.4	N-tier Architectures	19
1.2.5	Distributing Layers and Tiers	21
1.3	Communication in an Information System	22
1.3.1	Blocking and Non Blocking Interactions	22
1.3.2	Synchronous or Blocking Calls	23
1.3.3	Asynchronous or Non Blocking Calls.....	24
1.4	Summary.....	26
2	Middleware	29
2.1	Understanding Middleware	30
2.1.1	Middleware as a Programming Abstraction	30
2.1.2	Middleware as Infrastructure.....	32
2.1.3	Types of Middleware.....	33
2.1.4	Middleware Convergence	34
2.2	RPC and Related Middleware.....	35
2.2.1	Historical Background	35
2.2.2	How RPC Works	36
2.2.3	Binding in RPC	39
2.2.4	RPC and Heterogeneity	41
2.2.5	Extensions to RPC	42

2.2.6	RPC Middleware Infrastructure: DCE	43
2.3	TP Monitors	45
2.3.1	Historical Background	45
2.3.2	Transactional RPC and TP Monitors	46
2.3.3	Functionality of a TP Monitor	50
2.3.4	Architecture of a TP Monitor	51
2.4	Object Brokers	53
2.4.1	Historical Background	53
2.4.2	CORBA: System Architecture	54
2.4.3	How CORBA Works	54
2.4.4	CORBA: Dynamic Service Selection and Invocation ...	55
2.4.5	CORBA: Encapsulation	57
2.4.6	TP Monitors+Object Brokers = Object Monitors	58
2.5	Message-Oriented Middleware	59
2.5.1	Historical Background	59
2.5.2	Message-Based Interoperability	60
2.5.3	Message Queues	62
2.5.4	Interacting with a Message Queuing System	63
2.5.5	Transactional Queues	64
2.6	Summary	65
3	Enterprise Application Integration	67
3.1	From Middleware to Application Integration	68
3.1.1	From a Mainframe to a Set of Servers	68
3.1.2	From a Set of Servers to a Multitude of Services	68
3.1.3	An Example of Application Integration	69
3.2	EAI Middleware: Message Brokers	71
3.2.1	Historical Background	71
3.2.2	The Need for Message Brokers	72
3.2.3	Extending Basic MOM	73
3.2.4	The Publish/Subscribe Interaction Model	75
3.2.5	Distributed Administration of a Message Broker	77
3.2.6	EAI with a Message Broker	77
3.2.7	A Critical View of Message Brokers as EAI Platforms ..	81
3.3	Workflow Management Systems	82
3.3.1	Historical Background	82
3.3.2	Workflow Definition	84
3.3.3	Workflow Execution	86
3.3.4	Workflows as Programming in the Large	87
3.3.5	Integration of WfMSs with Other Middleware Technologies	89
3.3.6	Benefits and Limitations of WfMS	90
3.4	Summary	91

4	Web Technologies	93
4.1	Exchanging Information over the Internet	94
4.1.1	Before the Web	94
4.1.2	The Web	94
4.1.3	Limitations of HTTP	96
4.2	Web Technologies for Supporting Remote Clients	97
4.2.1	Need for Supporting Remote Clients	97
4.2.2	Applets	99
4.2.3	Common Gateway Interface	100
4.2.4	Servlets	102
4.3	Application Servers	102
4.3.1	Middleware for Web Applications	103
4.3.2	J2EE as the Core of an Application Server	103
4.3.3	Application Server Support for the Application Layer ..	105
4.3.4	Application Server Support for the Presentation Layer ..	108
4.4	Web Technologies for Application Integration	111
4.4.1	Architectures for Wide Area Integration	111
4.4.2	Middleware Extensions	112
4.4.3	Firewalls and Tunneling through HTTP	114
4.4.4	Common Data Representation: EDIFACT	115
4.4.5	XML	118
4.5	Summary	119

Part II Web Services

5	Web Services	123
5.1	Web Services and their Approach to Distributed Computing ..	124
5.1.1	Defining Web Services	124
5.1.2	Motivating the Need for B2B Integration	125
5.1.3	Limitations of Conventional Middleware in B2B Integration	127
5.1.4	B2B Integration before Web Services	130
5.1.5	B2B Integration with Web Services	131
5.1.6	Web services and EAI	134
5.2	Web Services Technologies	136
5.2.1	Service Description	136
5.2.2	Service Discovery	138
5.2.3	Service Interactions	139
5.2.4	Combining Web Services: Composition	141
5.3	Web Services Architecture	141
5.3.1	The Two Facets of Web Services Architectures	141
5.3.2	Internal Architecture of a Web Service	143
5.3.3	External Architecture of a Web Service	144
5.4	Summary	148

6	Basic Web Services Technology	151
6.1	A Minimalist Infrastructure for Web Services	152
6.2	SOAP: Simple Object Access Protocol	155
6.2.1	Goals of SOAP	155
6.2.2	Structure and Contents of a SOAP Message	157
6.2.3	Processing a SOAP Message	160
6.2.4	Binding SOAP to a Transport Protocol	161
6.2.5	A Simple Implementation of SOAP	163
6.2.6	Asynchronous SOAP	164
6.2.7	Binary Data and SOAP	165
6.3	WSDL: Web Services Description Language	165
6.3.1	Goals of WSDL	166
6.3.2	Structure of a WSDL Interface	167
6.3.3	Implications of the WSDL Model	170
6.3.4	Using WSDL	172
6.3.5	WSDL and Other Standards	173
6.4	UDDI: Universal Description Discovery and Integration	174
6.4.1	Goals of UDDI	174
6.4.2	Information in a UDDI Registry	175
6.4.3	UDDI Data Structures	175
6.4.4	Understanding tModels	176
6.4.5	UDDI Registry API	179
6.4.6	Searching the UDDI Registry	181
6.4.7	Storing WSDL Interfaces in a UDDI Registry	182
6.4.8	Public and Private Registries	184
6.5	Web Services at Work	185
6.6	Interactions Between the Specifications	187
6.6.1	Proliferation of Standards	187
6.6.2	Advanced SOAP: Effects on Service Description and Discovery	188
6.6.3	UDDI and Dynamic Binding	189
6.7	Related Standards	191
6.7.1	WS-Addressing	191
6.7.2	WS-Routing	192
6.7.3	WS-Security	192
6.7.4	WS-Policy	193
6.7.5	Web Services Invocation Framework (WSIF)	194
6.8	Summary	195
7	Service coordination protocols	197
7.1	An Introduction to Coordination Protocols	198
7.1.1	The Need for Coordination	198
7.1.2	Modeling Conversations between a Client and a Web Service	199
7.1.3	Modeling Conversations among Multiple Web Services	200

7.1.4	Service Interfaces and Coordination Protocols	205
7.1.5	Classifying Web Services Protocols	207
7.2	Infrastructure for Coordination Protocols	209
7.2.1	Conversation Controllers	209
7.2.2	Generic Protocol Handlers	212
7.2.3	Standardization Requirements for Coordination Protocols	214
7.3	WS-coordination	215
7.3.1	Goals of WS-Coordination	215
7.3.2	Components of WS-Coordination	216
7.3.3	Central Coordination	219
7.3.4	Distributed Coordination	222
7.3.5	Summary of WS-Coordination	224
7.4	WS-Transaction	225
7.4.1	Transactions in Web Services	226
7.4.2	Relationship with WS-Coordination	227
7.4.3	Atomic Transactions	228
7.4.4	Business Activities	230
7.5	RosettaNet	234
7.5.1	Goals and Scope of RosettaNet	234
7.5.2	Partner Interface Process (PIP) Specifications	235
7.5.3	RosettaNet Implementation Framework (RNIF)	238
7.6	Other Standards Related to Coordination Protocols	239
7.6.1	XML Common Business Library (xCBL)	239
7.6.2	Electronic Business Using eXtensible Markup Language (ebXML)	240
7.6.3	Web Service Choreography Interface (WSCI)	242
7.7	Summary	243
8	Service Composition	245
8.1	Basics of Service Composition	246
8.1.1	Composition as a Way to Master Complexity	246
8.1.2	The Need for Service Composition Middleware	248
8.1.3	Main Elements of Web Services Composition Middleware	249
8.1.4	Composition Versus Coordination Middleware	250
8.2	A New Chance of Success for Composition?	252
8.2.1	Limitations of Conventional Composition Middleware	253
8.2.2	Opportunities for Web Service Composition Middleware	254
8.3	Service Composition Models	256
8.3.1	Dimensions of a Web Service Composition Model	256
8.3.2	Component Model	256
8.3.3	Orchestration Model	257
8.3.4	Data and Data Transfer Model	264
8.3.5	Service Selection	267
8.3.6	Transactions	270

8.3.7	Exception Handling	273
8.4	Dependencies between Coordination and Composition	276
8.4.1	Coordination Protocols and Composition Schemas	276
8.4.2	Conversation Controllers and Composition Engines	282
8.5	BPEL: Business Process Execution Language for Web Services	283
8.5.1	Overview	284
8.5.2	Component Model	285
8.5.3	Orchestration Model	286
8.5.4	Data Types and Data Transfer	287
8.5.5	Service Selection	289
8.5.6	Exceptions and Transactions	290
8.5.7	Instance Routing	292
8.6	Summary	293
9	Outlook	295
9.1	State of the Art in Web Services	296
9.1.1	Available Technology	296
9.1.2	Current Architecture of Web Services	296
9.1.3	EAI as a Natural Fit for Today's Web Services	299
9.1.4	Emerging Trends	300
9.2	Applicability of Web Services	300
9.2.1	The Holy Grail	300
9.2.2	The Complexity of B2B Interactions	301
9.2.3	Bypassing Complexity in Closed Communities	303
9.2.4	Toward Open Communities	304
9.2.5	The Semantic Web	306
9.2.6	How Far Are We from the Holy Grail?	308
9.3	Web Services as a Problem and a Solution: an Example	308
9.3.1	Management in Conventional Middleware	309
9.3.2	Management in Web Services	309
9.3.3	Cross-Enterprise Management	312
9.3.4	Management through Web Services	316
9.3.5	Web Services Management Standards	317
9.4	Summary	320
	Bibliography	321
	Index	333