

---

# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	Networked Computing	1
1.2	Middleware	2
1.3	Event-Based Systems	3
1.4	Application Scenarios	4
1.4.1	Information Dissemination	4
1.4.2	Network Monitoring	4
1.4.3	Enterprise Application Integration	5
1.4.4	Mobile Systems	6
1.4.5	Ubiquitous systems	6
1.5	Putting Event-Based Systems Into Context	7
1.6	From Centralized to Internet-Scale Event Systems	8
1.7	Structure of the Book	8
<b>2</b>	<b>Basics</b>	11
2.1	Terminology	11
2.1.1	Events and Notifications	11
2.1.2	Producers and Consumers	12
2.1.3	Subscriptions and Filters	13
2.1.4	Event Notification Service	13
2.2	Models of Interaction	14
2.2.1	Request/Reply	15
2.2.2	Anonymous Request/Reply	15
2.2.3	Callback	16
2.2.4	Event-Based	16
2.2.5	Comparison	17
2.2.6	Interaction vs. Implementation	17
2.3	Notification Filtering Mechanisms	19
2.3.1	Channels	19
2.3.2	Subject-Based Filtering	19
2.3.3	Type-Based Filtering	19

2.3.4	Content-Based Filtering .....	20
2.4	A Model Distributed Notification Service .....	20
2.4.1	System Model .....	20
2.4.2	Architecture .....	21
2.4.3	Distributed Notification Routing .....	22
2.5	Specification of Event Systems .....	23
2.5.1	Formal Background .....	24
2.5.2	A Simple Event System .....	26
2.5.3	A Simple Event System With Ordering Requirements ..	30
2.5.4	Simple Event System With Advertisements .....	31
2.6	Further Reading .....	33
<b>3</b>	<b>Content-Based Models and Matching .....</b>	<b>35</b>
3.1	Content-Based Data and Filter Models .....	35
3.1.1	Tuples .....	35
3.1.2	Structured Records .....	36
3.1.3	Semistructured Records .....	52
3.1.4	Objects .....	56
3.2	Matching Algorithms .....	57
3.2.1	Brute Force .....	59
3.2.2	Counting Algorithm .....	59
3.2.3	Decision Trees .....	60
3.2.4	Binary Decision Diagrams .....	61
3.2.5	Efficient XML Matching .....	63
3.3	Further Reading .....	64
<b>4</b>	<b>Distributed Notification Routing .....</b>	<b>67</b>
4.1	System Model .....	67
4.2	Routing Algorithm Framework .....	69
4.2.1	Atomic Steps of the Implementation .....	69
4.2.2	Notification Forwarding and Delivery .....	72
4.2.3	Avoidance of Duplicate and Spurious Notifications .....	73
4.2.4	Routing Table Updates .....	73
4.3	Valid and Monotone Valid Routing Algorithms .....	74
4.3.1	Valid Routing Algorithms .....	74
4.3.2	Monotone Valid Routing Algorithms .....	76
4.4	Valid Framework Instantiations .....	77
4.5	Content-Based Routing Algorithms .....	80
4.5.1	Flooding .....	81
4.5.2	Simple Routing .....	82
4.5.3	Identity-Based Routing .....	85
4.5.4	Covering-Based Routing .....	91
4.5.5	Merging-Based Routing .....	98
4.5.6	Discussion .....	104
4.6	Extensions of the Basic Routing Framework .....	107

4.6.1	Routing With Advertisements .....	107
4.6.2	Hierarchical Routing Algorithms .....	112
4.6.3	Rendezvous-Based Routing .....	115
4.6.4	Topology Changes .....	117
4.6.5	Joining and Leaving Clients .....	119
4.6.6	Routing in Cyclic Topologies .....	120
4.6.7	Exploiting IP Multicast .....	122
4.6.8	Topology Maintenance .....	123
4.7	Further Reading .....	125
<b>5</b>	<b>Engineering of Event-Based Systems .....</b>	<b>129</b>
5.1	Engineering Requirements .....	129
5.1.1	Application Examples .....	130
5.1.2	Requirements .....	132
5.1.3	Existing Support .....	136
5.2	Accessing Publish/Subscribe Functionality .....	137
5.2.1	Generic APIs .....	137
5.2.2	Domain-Specific APIs .....	139
5.3	Using the API .....	140
5.3.1	Patterns and Idioms .....	141
5.3.2	Emitting Notifications .....	143
5.4	Further Reading .....	147
<b>6</b>	<b>Scoping .....</b>	<b>149</b>
6.1	Controlling Cooperation .....	150
6.1.1	Implicit Coordination and Visibility .....	150
6.1.2	Explicit Control of Visibility .....	151
6.1.3	The Role of Administrators .....	151
6.2	Event-Based Systems With Scopes .....	152
6.2.1	Visibility and Scopes .....	152
6.2.2	Specification .....	153
6.2.3	Notification Dissemination .....	156
6.2.4	Duplicate Notifications .....	158
6.2.5	Dynamic Scopes .....	159
6.2.6	Attributes and Abstract Scopes .....	161
6.2.7	A Correct Implementation .....	161
6.3	Event-Based Components .....	164
6.3.1	Component Interfaces .....	164
6.3.2	Scope Interfaces .....	164
6.3.3	Event-Based Components .....	167
6.3.4	Example .....	167
6.4	Notification Mappings .....	169
6.4.1	Specification .....	169
6.4.2	A Correct Implementation .....	173
6.4.3	Example .....	176

6.5	Transmission Policies .....	176
6.5.1	Publishing Policy .....	177
6.5.2	Delivery Policy .....	179
6.5.3	Traverse Policy .....	180
6.5.4	Influencing Notification Dissemination .....	181
6.6	Engineering With Scopes .....	182
6.6.1	Development Process .....	182
6.6.2	Scope Graph Handling .....	183
6.6.3	Scope Graph Language .....	187
6.7	Implementation Strategies for Scoping .....	196
6.7.1	Scope Architectures .....	197
6.7.2	Comparing Architectures .....	209
6.7.3	Implement Scopes as Event Brokers .....	210
6.7.4	Integrate Scoping and Routing .....	213
6.8	Combining Different Implementations .....	225
6.8.1	Architectures and Scope Graphs .....	226
6.8.2	Bridging Architectures .....	227
6.8.3	Integration With Other Notification Services .....	228
6.9	Further Reading .....	228
<b>7</b>	<b>Composite Events .....</b>	<b>231</b>
7.1	Application Scenarios .....	231
7.2	Requirements .....	234
7.3	Composite Events .....	234
7.4	Composite Event Detection .....	236
7.4.1	Composite Event Detectors .....	236
7.4.2	Composite Event Language .....	238
7.5	Detection Architectures .....	242
7.5.1	Centralized Detection .....	243
7.5.2	Distributed Detection .....	244
7.6	Further Reading .....	250
<b>8</b>	<b>Advanced Topics .....</b>	<b>253</b>
8.1	Security .....	253
8.1.1	Application Scenarios .....	254
8.1.2	Requirements .....	255
8.1.3	Access Control Techniques .....	256
8.1.4	Secure Publish/Subscribe Model .....	258
8.1.5	Further Reading .....	264
8.2	Fault Tolerance .....	264
8.2.1	Fault Masking .....	265
8.2.2	Self-Stabilizing Publish/Subscribe Systems .....	265
8.2.3	Self-Stabilizing Content-Based Routing .....	266
8.2.4	Generic Self-Stabilization Through Periodic Rebuild ...	273
8.2.5	Further Reading .....	276

8.3	Congestion Control .....	276
8.3.1	The Congestion Problem .....	277
8.3.2	Requirements .....	277
8.3.3	Congestion Control Algorithms.....	279
8.3.4	Further Reading .....	285
8.4	Mobility.....	287
8.4.1	Mobility Issues in Publish/Subscribe Middleware .....	289
8.4.2	Physical Mobility.....	290
8.4.3	Logical Mobility.....	295
8.4.4	Further Reading .....	302
<b>9</b>	<b>Existing Notification Services .....</b>	<b>305</b>
9.1	Standards .....	305
9.1.1	CORBA Event and Notification Service .....	305
9.1.2	Jini.....	310
9.1.3	Java Message Service (JMS) .....	311
9.1.4	Data Distribution for Real-Time Systems (DDS) .....	313
9.1.5	WS Eventing and WS Notification.....	317
9.1.6	The High-Level Architecture (HLA) .....	317
9.2	Commercial Systems .....	318
9.2.1	IBM WebSphere MQ .....	318
9.2.2	TIBCO Rendezvous .....	320
9.2.3	Oracle Streams Advanced Queuing .....	322
9.3	Research Prototypes .....	324
9.3.1	Gryphon .....	324
9.3.2	SIENA.....	326
9.3.3	JEDI .....	329
9.3.4	REBECA .....	331
9.3.5	Hermes .....	334
9.3.6	Cambridge Event Architecture (CEA).....	337
9.3.7	Elvin .....	340
9.3.8	READY.....	340
9.3.9	Narada Brokering .....	340
<b>10</b>	<b>Outlook .....</b>	<b>343</b>
	<b>References .....</b>	<b>349</b>
	<b>Index .....</b>	<b>379</b>



<http://www.springer.com/978-3-540-32651-9>

Distributed Event-Based Systems

Mühl, G.; Fiege, L.; Pietzuch, P.

2006, XX, 388 p., Hardcover

ISBN: 978-3-540-32651-9