## APPENDIX A

### Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADL</td>
<td>Architecture Description Language</td>
</tr>
<tr>
<td>AI</td>
<td>artificial intelligence</td>
</tr>
<tr>
<td>ALC</td>
<td>algorithmic logic coupling</td>
</tr>
<tr>
<td>AMI</td>
<td>asynchronous method invocation</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ATC</td>
<td>ambiguous type coupling</td>
</tr>
<tr>
<td>BBM</td>
<td>break before make</td>
</tr>
<tr>
<td>CBD</td>
<td>component-based development</td>
</tr>
<tr>
<td>CLI</td>
<td>Common Language Infrastructure</td>
</tr>
<tr>
<td>COM</td>
<td>Component Object Model</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>CR/LF</td>
<td>carriage-return/line-feed</td>
</tr>
<tr>
<td>DCE</td>
<td>Distributed Computing Environment</td>
</tr>
<tr>
<td>DCOM</td>
<td>Distributed Component Object Model</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DS</td>
<td>Dispatching Server</td>
</tr>
<tr>
<td>DSM</td>
<td>Distributed Shared Memory</td>
</tr>
<tr>
<td>EAI</td>
<td>enterprise application integration</td>
</tr>
<tr>
<td>EBS</td>
<td>event-based system</td>
</tr>
<tr>
<td>EIS</td>
<td>Enterprise Information System</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise JavaBeans</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In, First Out</td>
</tr>
<tr>
<td>GIOP</td>
<td>General Inter-ORB Protocol</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GUI</td>
<td>graphical user interface</td>
</tr>
<tr>
<td>GUID</td>
<td>Globally Unique Identifier</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Numbers Authority</td>
</tr>
<tr>
<td>IC</td>
<td>integrated circuit</td>
</tr>
<tr>
<td>IDL</td>
<td>interface definition language</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IIOP</td>
<td>Internet Inter-ORB Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Platform, Enterprise Edition</td>
</tr>
<tr>
<td>JDK</td>
<td>Java Development Kit</td>
</tr>
<tr>
<td>JIT</td>
<td>Just In Time</td>
</tr>
<tr>
<td>JMS</td>
<td>Java Message Service</td>
</tr>
<tr>
<td>JTA</td>
<td>Java Transaction API</td>
</tr>
<tr>
<td>JTS</td>
<td>Java Transaction Service</td>
</tr>
<tr>
<td>LC</td>
<td>logic coupling</td>
</tr>
<tr>
<td>LCE</td>
<td>loosely coupled event</td>
</tr>
<tr>
<td>LLC</td>
<td>literal logic coupling</td>
</tr>
<tr>
<td>LPC</td>
<td>local procedure call</td>
</tr>
<tr>
<td>MBB</td>
<td>make before break</td>
</tr>
<tr>
<td>MDB</td>
<td>message-driven bean</td>
</tr>
<tr>
<td>MIL</td>
<td>module interconnection language</td>
</tr>
<tr>
<td>MOM</td>
<td>message-oriented middleware</td>
</tr>
<tr>
<td>MSMQ</td>
<td>Microsoft Message Queuing</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>OO</td>
<td>object-oriented</td>
</tr>
<tr>
<td>OOP</td>
<td>object-oriented programming</td>
</tr>
<tr>
<td>OS</td>
<td>operating system</td>
</tr>
<tr>
<td>P2P</td>
<td>peer-to-peer</td>
</tr>
<tr>
<td>PC</td>
<td>procedure call</td>
</tr>
<tr>
<td>PCB</td>
<td>printed circuit board</td>
</tr>
<tr>
<td>PGM</td>
<td>Pragmatic General Multicast</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>PTP</td>
<td>point-to-point</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAD</td>
<td>rapid application development</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>RMI</td>
<td>Remote Method Invocation</td>
</tr>
<tr>
<td>RPC</td>
<td>remote procedure call</td>
</tr>
<tr>
<td>SAF</td>
<td>store-and-forward</td>
</tr>
<tr>
<td>SC</td>
<td>signature coupling</td>
</tr>
<tr>
<td>SDL</td>
<td>Specification and Description Language</td>
</tr>
<tr>
<td>SM</td>
<td>shared memory</td>
</tr>
<tr>
<td>SMP</td>
<td>symmetric multiprocessor</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SR</td>
<td>shared resource</td>
</tr>
<tr>
<td>SRMP</td>
<td>SOAP Reliable Messaging Protocol</td>
</tr>
<tr>
<td>TC</td>
<td>type coupling</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol / Internet Protocol</td>
</tr>
<tr>
<td>UDT</td>
<td>user-defined type</td>
</tr>
<tr>
<td>UI</td>
<td>user interface</td>
</tr>
<tr>
<td>UMA</td>
<td>Uniform Memory Access</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>UNC</td>
<td>Universal Naming Convention</td>
</tr>
<tr>
<td>UTC</td>
<td>unambiguous type coupling</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>


Box, Don, *Essential COM* (Boston: Addison-Wesley Professional, 1997).


Eder, Johann, Gerti Kappel, and Michael Schrefl, “Coupling and Cohesion in Object-Oriented Systems” (technical report, University of Klagenfurt, Austria, 1994).


Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software* (Boston: Addison-Wesley Professional, 1995).


Hitz, Martin and Behzad Montazeri, “Measuring Coupling and Cohesion in Object-Oriented Systems” (proceedings of the International Symposium on Applied Corporate Computing, Monterrey, Mexico, October 1995).


Jin, Yuhui and Rob Strom, “Relational Subscription Middleware for Internet-Scale Publish-Subscribe” (proceedings of the Second International Workshop on Distributed Event-Based Systems, San Diego, CA, June 2003).


Kielmann, Thilo, “Object-Oriented Distributed Programming with Objective Linda” (proceedings of the First International Workshop on High Speed Networks and Open Distributed Platforms, St. Petersburg, Russia, June 1995).


Prem, Jatinder, Bernard Ciconte, Peter Go, Scott Dunbar, and Manish Devgan, BEA WebLogic Platform 7 (Indianapolis, IN: Sams, 2003).


Rajagopalan, Mohan, Saumya K. Debray, Matti A. Hiltunen, and Richard D. Schlichting, “Profile-Directed Optimization of Event-Based Programs” (proceedings of the ACM Special Interest Group on Programming Languages [SIGPLAN], Berlin, Germany, June 2002).


Rowstron, Antony and Peter Druschel, “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems” (proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, Germany, November 2001).


Wilhelm, Uwe and Andre Schiper, “A Hierarchy of Totally Ordered Multicasts” (proceedings of the 14th IEEE Symposium on Reliable Distributed Systems, Bad Neuenahr, Germany, September 1995).


Index

**Numbers and Symbols**
- 10-10-10 guideline, and worker complexity, 337
- `+=` operator, binding C# delegates with, 152
- `-=` operator, unbinding C# delegates with, 152
- `=` operator, binding C# delegates with, 152

**A**
- ACME
  - architecture description language, 75
  - connectors, 235
  - ports, 223
- active builders, 48
- Activity diagrams, 201
  - actions, 202
  - decision points, 203
  - fork nodes, 203
  - join nodes, 203
  - merge points, 203
  - pins, 202
- Actors system, use of events, 74
- adapters, with typed object calls, 269
- AddHandler statement function, binding VB .NET events with, 155
- Aesop connectors, 235
- ALC to UTC transforms, 43
- algorithmic logic coupling, 10–11
- algorithms
  - complementary, 11
  - equipotent, 15
  - inverse, 13
  - symmetric, 14
- ambiguous type coupling. See coupling, type, ambiguous

Anders Hejlsberg, 164
Apple Macintosh, as a reactive system, 75
architecture description languages, 75
architectures, routers in layered, 433
artificial intelligence and blackboards, 74
ASAP Cars
  - case study, 567
  - client component, 570
  - common types, 634
  - configuring MSMQ, 640
  - functional requirements, 567
  - Invoicing component, 629
  - OrderEntry component, 574
  - OrderEntry test fixture, 584
  - OrderProcessor component, 597
  - parts scheduling components, 611
  - project configuration, 569
  - router example, 583
  - server components, 596
  - server-side pipelined message processing, 596
  - testing, 640
  - vehicle assembly components, 625
assemblies, in the .NET Framework, 150
assembly connectors, in Component Wiring diagram, 208
asynchronous interactions. See also blind interaction patterns
  - and completion tokens, 320
asynchronous method invocation, in CORBA, 174
asynchronous procedure calls. See procedure calls, synchronous vs. asynchronous
ATC to SC transforms, 50
attribute filtering, 85
background builders, 389
base64 encoding, 133
   with notification payloads, 131
BEA WebLogic, description, 189
Berners-Lee, Tim, 507
big-endian format, 255
   and serialization, 140
binders, 334
   description of, 398
   dynamic, 428
   and event subscription, 80
   evolvable systems with dynamic binders, 428
   example of JIT binder, 418
   implementing state machines with, 406
   JIT, 417
   late, 401
   smart, 416
   in a system browser, 463
   in the subscription process, 150
binding .NET delegates, operators, 152–153
binding events
   in Object Pascal, 166
   with VB .NET AddHandler statements, 155
blackboards, used in artificial intelligence, 74
blind asynchronous interaction patterns, 319
blind interaction patterns, 317
   interruptible, 328
blind interactions
   polled feedback in, 322
   pushed feedback in, 321
blind synchronous interaction patterns, 318
Block diagrams, 211
Borland
   Delphi, 76
   VisiMessage, 190
Brad Cox, Software ICs, 217
breadth-first order and notification delivery, 126
bridges, using routers, 435
builders, 334
   active, 48
   background, 389
   description of, 386
   JIT, 394
   reactive, 48
   smart, 388
   example of, 388
   in a system browser, 460
buses, 235
   HTTP connections on Signal Wiring diagrams, 239
   TIBCO information bus, 239
   VPN connections on Signal Wiring diagrams, 239
business components, as workers, 334
by-reference payloads, marshaling, 141
by-value payloads, marshaling, 140

C

C#
   delegates, 72
   events and delegates, 153
C# operators, binding .NET delegates with, 152
C++, 3, 6, 7, 21, 24, 59, 78, 151, 169, 189, 197, 265, 289
C2
   architecture description language, 75
   connectors, 235
   ports, 223
Cambridge Event Architecture, 83, 86, 146
case studies, SystemBrowser, 443
Catalysis diagrams, 214
causal order. See notification delivery order, causal
Change propagation and response graphs, 209
channels
  in CORBA, 169
  and notification delivery, 81
  out-of-band, 101
  in SDL diagrams, 211
  serialized connections, 102
Chord, peer-to-peer systems, 108
coalescing
  of notifications, 308
  payloads, 146
cohesion, 1
COM, 76
  event naming conventions, 300
.COM+
  events, 175
  notification filtering, 177
  persistent subscriptions, 176
  queued components, 177, 183
  subscription types, 176
COMet, COM-to-CORBA bridge, 76
Communication diagrams, message numbering in, 206
complementary algorithms, 11
completion tokens, in asynchronous interactions, 320
component, definition of, 2
Component Wiring diagrams, 208
components
  business, 334
  in Signal Wiring diagrams, 234
composite payloads, 146
composition, of notification payloads, 146
concurrent workers, 351
connection points, in Microsoft COM event model, 167
connectors, as signal buses, 235
consumers, in CORBA, 170
containers, EJB, 162
content filtering, 84
Control.Invoke, in .Net systems, 352
Coordinator teams, 360–361
coordinators, 333
  description of, 359
  and JEDI Dispatching Servers, 359
  leveraging multiple workers, 365
  managing the application life-cycle state with, 367
  managing the lifecycle of a system browser with, 450
  managing state across a group of workers, 367
  mediating communication between workers, 366
CORBA, 83, 174–175, 187, 235, 264
  asynchronous method invocation, 174
  buses, 235
  channels in, 169
  COM-to-CORBA bridge, 76
  consumers in, 170
  definition of event, 78
  event channels, 76
  event filtering, 172
  event service, 76, 78, 169
  event suppliers and consumers, 170
  messaging, 174
  notification service, 76, 78, 171
  origins of, 76
  pull mode, 169
  push mode, 169
  QoS, 173
coupling, 1
  and built-in types, 23
  charts, 33
  circular, 59
    avoiding with interfaces, 60
    avoiding with signature coupling, 62
classifications, 9
  in client-server architectures, 52
  common scenarios, 52
  common types in ASAP Cars case study, 634
diagrams, 35
due to interfaces, 65
due to UDTs in method parameters, 67
dynamic, 7
  mathematical properties, 8
flavors, 9
through instantiation, 19
K-coupling, 20
logic, 10
  algorithmic, 11
  literal, 16
  mathematical properties, 17
measuring, 32
in parent child relationships, 56
platform, 23
and programmer dependencies, 5
signature, 25
  mathematical properties, 30
static, 6
  mathematical properties, 6
static vs. dynamic, 9
symbol, 4
transforms, 38, 46
  ALC to UTC, 43
  ATC to SC, 50
  LLC to ATC, 41
  LLC to UTC, 39
  UTC to ATC, 46
type, 18
  ambiguous, 22
  mathematical properties, 24
  unambiguous, 19
and user-defined types, 64
vector space, 31
coupling space, 31
Cox, Brad, Software ICs, 217
critical sections, as sources of deadlocks, 275

D

Darwin
  bindings, 235
  use of events, 74
data-bound controls, as workers, 351
DCOM, 76
deadlocks, avoiding when firing events, 275
decision points, in UML Activity diagrams, 203
declarative sentences, 312
Delegate.Combine function, binding VB .NET delegates with, 153
deleagtes
  .NET untyped object calls, 150
  binding, 152–153
  in C#, 27, 72, 153
  multicasting, 152
  in VB .NET, 154
dellegation connectors, in Component Wiring diagram, 208
delivery latency, of notifications, 131
delivery order, of notifications, 297
delivery reliability, of notifications, 132
delivery systems
direct, 149
indirect, 168
Delphi. See also Object Pascal
event naming conventions, 300
origins of, 76
depth-first order, and notification delivery, 124
design patterns. See also interaction patterns
event listener, 137
model view controller, 74
subject observer, 71
diagrams
  Catalysis, 214
  Change propagation and response graphs, 209
coupling, in ASAP Cars case study, 571
  Espresso, 212–213
  Event-driven Process Chains, 209
  Lollipop, 210
  Petri nets, 209
  pin legends, in ASAP Cars case study, 572
  Rapide, 209
  SDL, 210–211
  Signal Wiring, 217
Szyperski connection-oriented diagrams, 210
UML, 199
UML Activity diagrams, 201–203
UML Communication, 206
UML Component Wiring diagrams, 208
UML Interaction Overview diagrams, 207
UML Sequence diagrams, 204–205
UML State Machine diagrams, 199–200
wiring, in ASAP Cars case study, 572
Workflow diagrams, 209
digital signal filters, 320
direct delivery systems, 149
direct-delivery, of notification, 79
directory service, looking up a message recipient, 256
distributed systems, modeling with Signal Wiring diagrams, 239
distributed transactions, and notification delivery, 120
DMZ, on Signal Wiring diagrams, 240
DNA, Microsoft Distributed interNet Applications model, 178
dynamic binders, 428
dynamic coupling, 7–8

E
EAServer, Sybase JMS implementation, 190
EJB. See Enterprise JavaBeans
EJB containers, 162
Enterprise Controllers, in Microsoft MSMQ, 180
Enterprise JavaBeans, 162–163
entity beans, Enterprise JavaBeans, 162
envelopes
as notification payloads, 135
retrieving data with, 143
equipotent algorithms, 15
errors, while firing events, 298
Espresso, 76
lines, 223
untyped object calls in, 162
Espresso diagrams, lines and ports, 212–213
Eureka Software Factory, buses, 235
event, definition of, 71
event channels, in CORBA, 76, 169
event multicasting, in JavaBeans, 158
event profiles and optimization, 308
event service
in COM+, 175
in CORBA, 78, 169
event sinks and sources, in Microsoft COM, 167
event subscription, 78
using binders, 80
event types, 83
Event-driven Process Chains, 209
event-listener design pattern, in JavaBeans, 137
events
in C#, 153
in Catalysis diagrams, 214
checking for subscribers, 254
CORBA definition, 78
firing. See firing events
in GUI systems, 75
history of, 74
in JavaBeans, 157
JavaBeans definition, 78
and Microsoft Visual Basic, 76
multicast, 135
naming conventions, 300
.NET Framework definition, 78
Object Pascal definition, 78
structured in CORBA, 172
in VB .NET, 154
vetoable in JavaBeans, 254
events and notifications, .COM+, 175
evolvable systems, using dynamic binders, 428
exceptions, while firing events, 276
external types, 19
fanout. See also notification delivery, fanout of an output signal, 218

feedback
  and interaction patterns, 319
  polled in blind interactions, 322
  polled in transparent interactions, 326
  pushed in blind interactions, 321
  pushed in transparent interactions, 324
files, used to deliver notifications, 94
filters
  and notification groups, 87
  and notifications, 83
fire-and-forget, and notification delivery, 108, 114
firing events, 253
  avoiding deadlocks, 275
  buffering messages, 260
  concurrently, 297
  dealing with exceptions, 276
  delivery order of notifications, 297
  errors, 298
multicast, 296
optimization, 307
  protecting the subscriber list, 255
push vs. pull, 254
the Fire method, 299
to retrieve data, 271
  and returned values, 299
  with typed objects, 267
unicast, 296
  with untyped targets, 264
using asynchronous procedure calls, 306
using messages, 256, 306
using pipes, 288
using procedure calls, 264, 301
using semaphores, 291
using shared files, 282
  using shared memory, 276
  using shared resources, 276
fork nodes, in UML Activity diagrams, 203
functional roles, 333

G
General Inter-ORB Protocol (GIOP), in CORBA, 175
groups, and notification filtering, 87
Gryphon, content filtering in, 84
GUIs, and events, 75

H
handshaking, as an interaction pattern, 331
Harel Statecharts, 199
Hejlsberg, Anders, 164
Hermes, 83
history, of events, 74
HostEmulator test fixture, and HttpService, 549
HTML documents, and HTTP traffic, 507
HTTP, 133
  Content-Length header, 509
  definition of, 507
  headers, 508
  persistent connections, 508
  pipelined requests, 508
  response parsing in HttpService, 530
  using a custom Sequence-Number header, 509
HTTP connections, depicting on Signal Wiring diagram buses, 239
HttpService
  case study, 508
  connection pool management, 514
  HostEmulator test fixture, 549
  managing the connection, 518
  parsing HTTP responses, 530
  semaphores, 518
  testing, 543
  threading, 537
IBM WebSphere MQ, 77
   description, 189
iBus, Java, 235
ICQ, 107
IDL. See interface definition language
IETF, 507
impedance, of a signal, 218
imperative sentences, 312
indirect delivery systems, 168
indirect-delivery, of notifications, 79
infinite impulse response filters, 320
input pins, 224
interaction dynamics, 311
Interaction Overview diagrams, 207
interaction patterns
   blind interactions, 317–319
   feedback, 319
   handshaking, 331
   interruptible interactions, 327–329
   and natural languages, 312
   polled feedback in blind interactions, 322
   polled feedback in transparent interactions, 324, 326
   pull interactions, 315
   push interactions, 313
   pushed feedback in blind interactions, 321
   push-pull models, 313
   round-robin polling, 316
   transparent interactions, 323
interaction types, and natural language sentence types, 313
interactions
   blind, 351
   interruptible, 327–329
   pull, 315
   push, 313
   transparent, 323
interface definition language, 76, 169
   and marshaling, 141
interfaces, and statically bound signals on Signal Wiring diagrams, 232
Internet Engineering Task Force, 507
Internet-scale systems, 111
Interoperable Routing Protocol (IRP), in CORBA, 175
interrogative sentences, 312
interrogator, in pull interactions, 315
interruptible blind interactions, 328
interruptible interactions, 327
interruptible transparent interactions, 329
inverse algorithms, 13
Iona, COMet bridge, 76
IP multicasting, 116
IRP. See Interoperable Routing Protocol (IRP)
Ivy bus, 235

J
J2EE, 76
   JavaBeans, 162
Java, 3, 6, 7, 9, 19, 23–24, 27, 51, 59, 64, 71, 99, 105, 134, 150, 157, 159, 161, 163, 185–191, 197, 226, 260, 264, 266, 272, 297, 300, 305, 386–387, 403
Java iBus, 235
Java Message Service. See JMS (Java Message Service)
Java RMI signals, on Signal Wiring diagrams, 240
java.reflect.Method, untyped object calls with, 160
JavaBeans, 74, 76, 81, 83, 104, 116, 119, 134, 139, 160, 163, 264, 493
definition of event, 78
Enterprise, 162
event-listener design pattern in, 137
event listeners, 157
event multicasting, 158
event naming conventions, 300
untyped object calls, 159
vetoable events, 254
JBoss JMS, implementation, 190
JEDI, 112
  content filtering in, 84
  Dispatching Servers as coordinators, 359
JIT binders, 417
JIT builders, 394
JMS (Java Message Service), 76
  delivery models, 188
  events and notifications, 185
  implementations, 189–190
  interfaces, 186
  interoperability, 186
  and message-driven beans, 163
  message structure, 186
  multicasting, 188
  notification filtering, 185, 188
  QoS, 187
join nodes, 203

K
  Kazaa, peer-to-peer systems, 107
K-coupling, 20
    unambiguous type, 19
Kulik, content filtering in, 84

L
  late binders, 401
  lazy loading, and JIT builders, 394
LifecycleCoordinators
    example, 368
    example with separate sub-state coordinators, 371
    states, 372
    in a system browser, 450
linguistics, as the basis of interaction patterns, 312
listeners
    in JavaBeans, 157
    in push interactions, 313
literal logic coupling, 16
little-endian format, 255
  and serialization, 140
LLC to ACT transforms, 41
LLC to UTC transforms, 39
logic coupling, 10–11
Lollipop diagrams, 210
  lollipops, in component wiring diagrams, 208
  loosely coupled events, in COM+, 175

M
  many-to-one connections
    on Signal Wiring diagrams, 230
    with buses on Signal Wiring diagrams, 238
marshaling
  with by-reference payloads, 141
  with by-value payloads, 140
  payloads, 140
  and serialized connections, 102
MDBs (message-driven beans), Enterprise JavaBeans, 163
menus, example of implementation in a system browser, 466
merge points, in UML Activity diagrams, 203
message-based-systems, 77
message-drivne beans, Enterprise JavaBeans, 163
message-oriented-middleware, 77
  for indirect notification delivery, 108
message numbering, in Communication diagrams, 206
messages
  as notification payloads, 77
  with event notifications, 256
messaging, in CORBA, 174
messaging services, origins of, 77
messaging systems
  and centralized notification delivery, 109
  for indirect notification delivery, 108
method references
  using Java reflection, 27
  using .NET delegates, 27
methods, universal, 266
Microsoft COM, event sinks and event sources in, 167
Microsoft COM event model, connection points in, 167
Microsoft Distributed interNet Applications (DNA) model, 178
Microsoft Foundation Classes, event naming conventions, 300
Microsoft MSMQ, 77
delivery models, 182
events and notifications, 178
interoperability, 181
message routing, 180
multicasting, 183
notification filtering, 183
triggers, 183
Microsoft Transaction Server, and DNA, 178
Microsoft Visual Basic, and events, 76
Microsoft Windows, as a reactive system, 75
middleware systems. See also messaging systems for indirect notification delivery, 108
MIL75, module interconnection language, 75
mobile subscribers, Rendezvous-Notify, 108
model view controller design pattern, 74
module interconnection languages, 75
Morpheus, peer-to-peer systems, 107
MSMQ, 77, 111
and ASAP Cars case study, 568
and COM+ queued components, 177
configuring in ASAP Cars case study, 640
events and notifications, 178
interfaces, 181
message structure, 181
QoS, 182
multicasting
in JavaBeans, 158
in JMS, 188
in Microsoft MSMQ, 183
in TIBCO Rendezvous, 196
in TIBCO SmartSockets, 191
using PGM, 191
with .NET delegates, 152
multi-threading, with workers and coordinators, 351

multicasting
in JavaBeans, 158
in JMS, 188
in Microsoft MSMQ, 183
in TIBCO Rendezvous, 196
in TIBCO SmartSockets, 191
using PGM, 191
with .NET delegates, 152

named pipes, to deliver notifications, 290

notification
definition of, 71
indirect-delivery model, 79

notification delivery
asynchronous, 114
breadth-first order, 126
centralized, 108
channels, 81
depth-first order, 124
direct, 106
distributed, 111
distributed transactions, 120
fanout, 115
fire and forget and, 108, 114
indirect, 108
order, 119, 121
using out-of-band channels, 101
peer-to-peer, 107
point-to-point, 106
priority, 118
using procedure calls, 103, 134
local, 105
remote, 105
protocols, 91
race conditions, 126
reliability, 117
using semaphores, 99
using serialized channels, 102
using shared files, 94
using shared memory, 95
using shared resources, 93, 132
pushing and pulling, 93
synchronous, 114
thoughput, 118
timing, 118
transactions, 119
notification delivery order
causal, 121
partial, 123
total, 128
notification filtering
in COM+, 177
in JMS, 188
in TIBCO Rendezvous, 196
in TIBCO SmartSockets, 193
with JMS, 185
with Microsoft MSMQ, 183
notification forwarding XE, 139
notification payload, 77
little-endian vs. big-endian formats, 255
packaging, 255
notification payloads, 131
envelopes, 135
object-based, 137
record-based, 136
string-based, 135
and user-defined types, 137
and XML documents, 136
notification relaying, 139
notification routing, and workers, 337
notification service, in CORBA, 78, 171
notification systems, and centralized notification
delivery, 109
notifications
coalescing, 308
combining to retrieve data, 274
composing together, 146
content filtering, 84
delivery frequency, 132
delivery latency, 131
delivery order, 297
delivery reliability, 132
direct-delivery model, 79
and filters, 83
forwarding using routers, 434
multicast, 296
payload size, 132
as signals, 73
unicast, 296
used to retrieve data, 271
numbering pins, on Signal Wiring diagrams, 228

0

object, and notification payloads, 137
Object Pascal, 6, 73–74, 76, 81, 83, 104, 116, 134,
164–165, 167, 226, 265, 289–290,
293–294, 304
binding events, 166
definition of event, 78
event naming conventions, 300
objects, in Signal Wiring diagrams, 218
observer, in subject observer design pattern, 71
Obvents, 139
Odell, Event diagrams, 201
off-page signal references, on Signal Wiring diagrams, 247
OMG IDL, 169. See also interface definition language
OMG RPC, 264
one-to-many connections
on Signal Wiring diagrams, 230
with buses on Signal Wiring diagrams, 237
optimizing, notification delivery, 307
Oracle Advanced Queueing, 190
order, of notification delivery. See notification delivery, order
out-of-band channels, 101
output pins, 226
overlay networks and peer-to-peer systems, 107

Payloads
coalescing, 146
composite, 146
marshaling, 140
messages, 77
and notification frequency, 132
and notifications, 131
PC signals
as inputs on Signal Wiring diagrams, 224
as outputs on Signal Wiring diagrams, 226
in Signal Wiring diagrams, 220
used to fetch data, 221
peer-to-peer systems, 116. See also notification delivery, peer-to-peer
Chord, 108
Herald, 116
ICQ, 107
Kazaa, 107
Morpheus, 107
Napster, 107
NICE, 116
Pastry, 108, 116
Scribe, 108
Tapestry, 108, 116
persistence, example in a system browser, 472
persistent subscriptions, in COM+, 176
Petri nets, 209
PGM. See Pragmatic General Multicast
pin legend tables, on Signal Wiring diagrams, 229
pins
hardware fanout, 218
hardware tri-state mode, 218
input, 224
naming on Signal Wiring diagrams, 223
numbering on Signal Wiring diagrams, 228
output, 226
in Signal Wiring diagrams, 223
statically bound outputs on Signal Wiring diagrams, 231
in UML Activity diagrams, 202
wiring together, 228
pipelining, with HTTP, 508
pipes, used to send notifications, 288
point-to-point, as a style, 149
point-to-point interactions, 256
point-to-point systems, with messaging systems, 110. See also notification delivery, point-to-point
poles, and switches on Signal Wiring diagrams, 240. See also switches
polling, round-robin, 316
Polylith
  buses, 235
  module interconnection language, 75
ports
  in ACME, 223
  in ADLs, 223
  in Catalysis diagrams, 214
  in Espresso diagrams, 212
  in UML Component Wiring diagrams, 208
  in Wright, 223
poset, partially ordered set, 75
Pragmatic General Multicast, 191
priority, and notification, 118–119
procedure calls
  with event notifications, 264
  local and remote, 114
  and notification delivery, 134
  synchronous vs. asynchronous, 103
process chains. See Event-driven Process Chains
Process diagrams, in SDL diagrams, 211
properties, Catalysis diagrams, 214
protocols, in CORBA, 175
PTP. See point-to-point
publish-subscribe, as a style, 149
publish-subscribe systems, 77
pull interactions, interrogator and respondent, 315
  pull mode, in CORBA, 169
  push interactions, talker and listener, 313
  push mode, in CORBA, 169
  push-pull models, and interaction patterns, 313
Q
QoS, 77, 117
  in JMS, 187
  Microsoft MSMQ, 182
  in TIBCO Rendezvous, 196
  in TIBCO SmartSockets, 192
queued components, in COM+, 177, 183
race conditions, and notification delivery, 126
Rapide, architecture description language, 75, 209
reactive builders, 48
reactive systems, 75
records, and notification payloads, 136
Reflection, and JavaBeans untyped object calls, 160
reflection, calling untyped targets, 266
reliability, and notification delivery, 117
remote procedure calls, to deliver notifications, 264
Remoting framework, and pipes in .NET systems, 289
Rendezvous, events and notifications, 193
Rendezvous-Notify, delivering notifications to mobile subscribers, 108
respondent, in pull interactions, 315
RFC 1945 and HTTP 1.0, 507
RMI, 264
  Java RMI signals on Signal Wiring diagrams, 240
RMTP, and notification multicasting, 116
roles, functional, 333
round-robin polling, as an interaction pattern, 316
routers
  as bridges, 435
  connecting coordinators with, 435
description of, 433
example in ASAP Cars case study, 583
  in layered architectures, 433
  as notification forwarders, 434
RPC
  and ASAP Cars case study, 568
  OMG, 264
  with PC signals, 220
S

Scribe, peer-to-peer systems, 108
SDL diagrams, 210–211
semaphores, 99
  in HttpService, 518
    used to send notifications, 291
sentence types, and interaction patterns, 312
Sequence diagrams, 204-205
  of a LifecycleCoordinator example, 384
sequence filtering, 86
serialization
  big-endian and little-endian format, 140
  of notification payloads, 255
serialized connections, 102
session beans, Enterprise JavaBeans, 162
shared files
  and notification delivery, 94
  used to send notifications, 282
shared memory, 95
  used to send notifications, 276
shared resources
  and notification delivery, 93, 132
  with event notifications, 276
Siena, content filtering in, 84
signal references, with Signal Wiring diagrams, 247
signal timing, showing in Signal Wiring diagrams, 251
signal types, in Catalysis diagrams, 214
Signal Wiring diagrams, 217
  buses, 235
  components, 234
  depicting distributed systems, 239
  depicting many-to-one connections, 230
  depicting many-to-one connections with buses, 238
  depicting one-to-many connections, 230
  depicting one-to-many connections with buses, 237
Java RMI signals, 240
  labeling items in multipage diagrams, 251
  labeling pages, 246
  multipage signals, 247
  objects, 218
  off-page signal references, 247
  pages, 246
  PC signals, 220
  pin legend tables, 229
  pins, 223
  showing DMZs, 240
  showing signal timing in, 251
  showing switch information, 244
  signals, 220
  signal references with, 247
  SR signals, 221
  switch controllers, 241
  switches, 240
  using multiple levels of details, 249
signals
  hardware impedance, 218
  in Signal Wiring diagrams, 220
  multipage in Signal Wiring diagrams, 247
  naming conventions, 299
  PC, 220
  SR, 221
  statically bound on Signal Wiring diagrams, 231
signal routes, in SDL diagrams, 211
signature coupling. See coupling, signature
signatures
  as types, 25, 29
  compatibility, 27
  method, 25
  of universal methods, 266
sinks, in Microsoft COM event model, 167
Smalltalk, 74
smart binders, 416
smart builders, 388
SmartSockets, events and notifications, 191
SOAP
   depicting on Signal Wiring diagram buses, 239
   and notification delivery, 132
   with PC signals, 220
sockets, handling incoming traffic in 
   HttpService, 550
Software buses, 75
Software ICs, 217
Sonic Software SonicMQ, JMS 
   implementation, 190
splash screens, managing using a 
   LifecycleCoordinator, 372
SR signals
   as inputs on Signal Wiring diagrams, 225, 227
   in Signal Wiring diagrams, 221
   showing synchronicity in diagrams, 222
SRM, and notification multicasting, 116
State Machine diagrams, 199–200
state machines, with late binders, 406
static coupling, 6
store-and-forward systems, 77
strings, and notification payloads, 135
structured events, in CORBA, 172
subject observer design pattern, 71
subject-based filtering, 85
subscription process, using binders, 150
subscriptions 
   in COM+, 176
   and events, 78
   languages, 84
   models, 80
   the process, 79
   policies, 89
Sun Microsystems, Sun Java System Message 
   Queue, 190
suppliers, in CORBA, 170
SWBus, 235
switch controller, in Signal Wiring diagrams, 241
switch tables, on Signal Wiring diagrams, 244
switches
   break-before-make strategy, 241
   controlling connections at the interface 
      level, 245
   in Signal Wiring diagrams, 240
   make-before-break strategy, 242
   poles and throws, 240
   showing switching information on Signal 
      Wiring diagrams, 244
Sybase EAServer, 190
symmetric algorithms, 14
synchronous procedure calls. See procedure 
   calls, synchronous vs. asynchronous
SystemBrowser, case study 1, 443
Szyperski, connection-oriented diagrams, 210
T
talker, in push interactions, 313
Tapestry. See peer-to-peer systems, Tapestry
TCP, 133
TCP/IP
   and HTTP, 507
   connection establishment, 507
teams, of coordinators and workers, 360
testing, of a system browser, 495
threading
   handling with a coordinator team, 361
   in ASAP Cars case study, 612
   in HttpService, 537
   with workers and coordinators, 351
throws, and switches on Signal Wiring 
   diagrams, 240. See also switches
TIBCO
   Enterprise for JMS, 189
   information bus, 239
   SmartSockets, 190
TIBCO Rendezvous, 77
events and notifications, 193
interfaces, 197
interoperability, 197
message routing, 195
message structure, 195
multicasting, 196
notification filtering, 196
QoS, 196
TIBCO SmartSockets, 190
events and notifications, 191
message structure, 192
notification filtering, 193
QoS, 192
Tim Berners-Lee, 507
toolbars, example of implementation in a system browser, 466
ToolBus, 75, 235
ToolTalk, 95
topic-based filtering, 85
transactions, and workers, 337
transient subscriptions, in COM+, 176
translation filtering, 86
transparent interaction patterns, 323
transparent interactions
interruptible, 329
polled feedback in, 326
pushed feedback in, 324
triggers
and State Machine diagrams, 199
in Microsoft MSMQ, 183
type coupling, minimizing using a builder, 386
typed events, in CORBA, 172
typed object calls, drawback of, 269
typed objects, as event subscribers, 267
types, external, 19
types of events, 83

U
UDP, 133
UDTs, coupling in ASAP Cars case study, 634
UI
updating on background threads, 352
updating via Control.Invoke, 352
UI controls, implementing with workers, 338
UML
Activity diagrams, 201-203
Communication diagrams, 206
Component Wiring diagrams, 208
diagrams, 199
Interaction Overview diagrams, 207
Sequence diagrams, 204–205
State Machine diagrams, 199–200
UML stereotypes, used to denote functional roles, 336
unambiguous type coupling, 19
unbinding, with C# -= operator, 152
unicast notifications, 115, 296
UniCon, connectors, 235
universal methods, 266
untyped object calls
with .NET delegates, 150
with JavaBeans, 159
untyped targets
and universal methods, 266
as event subscribers, 264
use cases, of a system browser, 493
user settings, persisting in a system browser, 472
user-defined types, and notification payloads, 137
V
VB .NET, events and delegates, 154
Vienna Component Framework, 76
VisiMessage, Borland JMS implementation, 190
Visual Basic, and event naming, 299
VPN connections, depicting on Signal Wiring diagram buses, 239
W
Web browsers, and HTTP traffic, 507
WebLogic, 111
description, 189
WebSphere, 95, 111
WebSphere MQ, 77
description, 189
wiring
  events with binders, 398
  of pins, 228
wiring diagrams, LifecycleCoordinator
  example, 384
Wiring diagrams. See Signal Wiring diagrams
  workers, 333
    complexity guidelines, 337
    composite, 335
    concurrent, 351
    as data-bound controls, 351
    description of, 334
    examples of, 338
    synchronizing with a coordinator, 361
    as UI controls, 338
    what they should do, 334
    what they shouldn’t do, 336

Workflow diagrams, 209
Wright
  architecture description language, 75
  connectors, 235
  ports, 223

X
X Window System, as a reactive system, 75
XML document, and notification payloads, 136