

Subject Index

A

Abstract class 281, 553
Active object 254
Adaptability *see* Adaptation and extension
Adaptation and extension 38, 40, 42, 423–427
 discussion 421–423
 multi-platform support 424–425
 out-of-band extensions 424
 specialized algorithms 424
 service extensions and restrictions 424
 pattern abstracts 427–428
 pattern descriptions 436–462
 pattern discussion 426–433
Anonymous communication 235
Application 554
 application integration 221
 application partitioning 170–171
Application control 37–38, 40, 42, 330–332
 discussion 327–330
 data structure decoupling 330
 explicit coordination and control of requests 329
 location decoupling 330
 security 329
 technology decoupling 329
 workflow decoupling 330
 pattern abstracts 332–333
 pattern descriptions 337–352
 pattern discussion 331–334
Architectural drift 194
Architecture *see* Software architecture
Asynchronous communication 28, 221–223, 235, 262, 268–269
Asynchronous I/O 107, 263, 360–361, 554
Asynchronous data transfer 221–222

B

Bandwidth 76, 127, 246, 249, 301–303, 495, 554
Broadcast 225, 233, 236, 495–496, 554
Broker architecture 73–75
 see also CORBA; Object Request Broker (ORB)

C

Callback 555
CCM *see* CORBA Component Model (CCM)
Change propagation infrastructure 234
Change requests 188–190
Client 555
Communication middleware 95–127
 architecture 96–100
 distributed object computing
 middleware 25, 163, 556, 558
 message-orientated middleware 28, 30, 177
 publish/subscribe middleware 25, 28, 29
 structured communication 22–24
 see also CORBA; Object Request Broker (ORB)
Completion event 264, 268, 555
Component middleware xxiv,25–28, 278
 architecture 95–127
 container 98–102, 106–107, 111, 113, 115, 118–119, 276, 278, 281, 288, 428–429, 435, 438, 461, 475, 478–479, 481–482, 484, 488–489, 492, 497, 499–501, 556
CORBA Component Model (CCM) 27, 75, 97
Enterprise JavaBeans (EJB) 27, 75, 556
Enterprise Service Bus (ESB) 30

- Component middleware (*continued*)
 - Service-Orientated Architecture (SOA) 30, 222, 272
 - web services 25, 30–32, 212, 215
 - Component partitioning 37, 40, 42, 305–327
 - discussion 303–306
 - component partitioning 304
 - component quality 305
 - component flexibility 305
 - distribution of functionality 305
 - concurrency and parallelism 306
 - pattern abstracts 308–309
 - pattern descriptions 313–327
 - pattern discussion 308–310
 - Component 555
 - Aggregate components 300–303
 - business delegates 292
 - component groups 326–327
 - component object 26, 214–215, 309, 317, 400–401, 405, 418–419, 555
 - component/implementation separation 281–283
 - concrete component 309, 555
 - concurrency and parallelism 308
 - distribution of functionality 307
 - expressiveness and simplicity 273
 - flexibility 307
 - limiting client access 284–285, 286–287, 290
 - loose coupling and stability 273
 - partitioning 306
 - quality attributes 272, 307
 - software productivity and quality 306
 - Component middleware 26–27
 - Composite pattern 11–12
 - Compound pattern 11–12
 - Concrete class 281, 409, 436, 477, 530, 555
 - Concurrency 38, 40, 42, 85–87, 353–370, 556
 - discussion 351–353
 - software diversity 353
 - multi-threading costs 353
 - portability 353
 - pattern abstracts 355–356
 - pattern descriptions 359–369
 - pattern discussion 354–356
 - Condition variable 115, 356, 374, 376, 381, 556
 - Connection 556
 - Connection establishment 109–110, 564
 - Container 568
 - CORBA (Common Object Request Broker Architecture) 25, 26, 30, 96, 215–216, 236, 238, 556
 - CORBA Component Model (CCM) 75
 - reference architecture and ORB 108
 - see also* Object Request Broker (ORB)
 - CPU 556
 - CRC card 36, 556
 - Critical section 204, 243, 247, 338, 340, 372–373, 375, 380, 384, 386–388, 390–392, 556
- D**
- Data driven applications 202
 - Database Access 39, 40, 42, 533–547
 - discussion 531–533
 - existing legacy and IT infrastructures 532
 - customer support 532
 - experience 532
 - performance 532
 - data usage scenarios 532
 - pattern abstracts 535–536
 - pattern descriptions 538–547
 - pattern discussion 534–535
 - Deadlock 35, 38, 283, 369, 373–375, 380, 384–385, 557
 - Decoupling
 - data structures 330
 - location 330
 - technology 331
 - workflow 330–3
 - Demarshaling 218, 445, 557
 - Demultiplexing *see* Event demultiplexing and dispatching
 - Design 557
 - Design Pattern 6, 429, 557
 - Device 557

- Distributed computing xxviii-xxix, 17-32
 - ad hoc network programming 22
 - asynchronous arrival of events 254
 - benefits 18-20
 - collaboration and connectivity 19
 - economics 19
 - fault tolerance 19-20
 - performance and scalability 19
 - challenges 20-1
 - accidental complexity 355, 464
 - continuous re-invention and re-discovery 21
 - inadequate methods and techniques 21
 - inherent complexity 20
 - complexity issues 20-21
 - continuous re-invention and discovery 21
 - inherent distribution 20
 - location independence 237-239
 - method and technique limitations 21
 - service-oriented architectures (SOA) 29-30
 - SOAP 29-30
 - structured communication 22-14
 - supporting technologies 21-31
 - ad hoc network programming 22-23
 - component middleware xxiv, 25-28, 278
 - distributed object computing middleware 25
 - message-orientated middleware 28, 30, 177
 - publish/subscribe middleware 25, 28-29
 - service-orientated architecture 25, 30, 222, 272-273, 569
 - structured communication 22-24
 - web services 29-31
 - see also* Communication middleware; CORBA; Component Middleware
 - Distributed object computing (DOC)
 - middleware 24-26, 27
 - Distributed system 558
 - see also* Distributed computing
 - Distribution 558
 - Distribution infrastructure 211-251
 - discussion 209-211
 - pattern abstracts 214-218
 - pattern descriptions 221-251
 - pattern discussion 214-218
 - see also* Distributed computing
 - Domain 558
 - Domain object 206-208
 - access 71-2, 82-4
 - collaboration and containment relationships 208
 - concurrency 85-87
 - configuring at runtime 88-89
 - functionality 79-82
 - Dynamically linked library (DLL) 122-123, 491, 558
- E**
- Encapsulation 73
 - Endpoint 103, 107-109, 214-215, 217, 558
 - Enterprise Java Beans (EJB) 96
 - Enterprise Service Bus (ESB) 30
 - Event 558
 - Event demultiplexing *see* Event demultiplexing and dispatching
 - Event demultiplexing and dispatching 37, 41, 45, 48, 96, 98, 100-107, 111, 119-120, 123-125, 253-269, 560
 - discussion 251-253
 - asynchronous arrival of events 252
 - hiding the complexity of event demultiplexing and dispatching 253
 - simultaneous arrival of events 252
 - multiple event types 252
 - non-deterministic arrival of events 254
 - pattern abstracts 256
 - pattern descriptions 259-269
 - pattern discussion 254-256
 - Event dispatching *see* Event demultiplexing and dispatching
 - Event handler 48, 104-107, 110, 125, 159, 250, 255, 258-3, 265, 268, 363-364, 394, 558
 - Exception safe 13, 428, 451, 558
 - Extensibility *see* Adaptation and extension

F

Factory 558
 Flow control 61–65, 70, 77, 90, 108, 111,
 113, 130, 144, 160, 361, 558
 Function 559
 Functional responsibilities, separating
 209
 Functionality, distributing 79–82
 Future 559

G

Gateway 68–9, 76, 79, 82, 124, 144, 535,
 542–545, 559
 General Inter-ORB Protocol (GIOP)
 105–106
 Group communication 235
 GUI (Graphical User Interface) 288, 330, 460,
 559

H

Handle 559
 Heuristic computation 205–207
 Hosts 146, 154, 195, 210, 214, 315, 321,
 326, 330, 353, 435, 461–462,
 559
 HTTP (HyperText Transfer Protocol) 29–30,
 190, 333, 359, 420, 559

I

Idiom 5, 8, 316, 391, 428, 431, 485, 537, 560
 Inheritance 118, 139, 267, 316, 397, 415,
 430, 433, 436, 454, 458, 526, 535,
 538, 540, 547, 555
 Inlining 456, 560
 Instance 560
 Instantiation 530, 560
 Inter-mode dependencies 464
 Interface 560

Interface partitioning 37, 40, 41, 271–303
 discussion 269–4
 component and clients
 heterogeneity 274
 component distribution 271
 component responsibilities and
 contract specification 272
 expressiveness and simplicity 271
 heterogeneity of components and
 their clients 272
 loose coupling and stability 273
 quality attributes 272–273
 pattern abstracts 274–275
 pattern descriptions 281–303
 pattern discussion 275–278
 Internet 17, 19, 30, 212, 331, 349, 560
 Interprocess communication (IPC) 22–23, 59,
 198, 237, 240–251, 246, 249–251,
 267, 291, 444, 561
 Invariant 66, 199, 307, 316, 323, 372,
 453–454, 456, 561
 Iteration 6–7

J

Java RMI 25
 Java Virtual Machine (JVM) 104
 JavaBeans 26
 Jitter 20, 79, 220, 282, 302, 324, 400, 561

L

Latency 20, 38, 79, 113, 170, 220, 246–247,
 250, 266, 302, 324, 400, 474, 561
 Layers concept 561
 access layer 68, 69
 business objects layer 67, 69
 business process layer 67, 69, 78
 container and object adapter layer 102,
 106–107, 113, 115, 119

- database access layer 42, 88, 92–94, 146, 155, 535–540, 542, 544, 546
 - decomposing layers 68–70
 - infrastructure layer 68, 69
 - Operating system abstraction layer 186
 - ORB core layer 102, 104, 106, 110, 113, 115, 119, 125, 238
 - ORB interface layer 102
 - presentation layer 67, 69, 78
 - Load balancing 27, 275, 278, 291–293, 446, 489, 561
 - Location independence 73
 - Lock 561
 - Locking *see* Synchronization
 - Logging records 85
 - Loose coupling 224, 229–230, 234–236, 416
- M**
- Marshaling 8, 11, 96, 116–117, 123–124, 218, 445, 561
 - Message
 - definition 561–562
 - message passing 22, 26, 561
 - self-describing message 222, 568
 - Message-oriented middleware 27–29
 - Method 562
 - Middleware
 - see also* Communication middleware; Component middleware; Distributed computing/systems; Distribution infrastructure problem; Layers concept
 - Modal behavior 39, 40, 42, 463–472
 - discussion 461–462
 - inter-mode dependencies 462
 - minimizing conditionals 462
 - mode visibility 462
 - mode independence 466
 - mode visibility 464–466
 - pattern abstracts 465
 - pattern descriptions 467–472
 - pattern discussion 463–464
 - Module 4, 470, 562
 - Monitor 562
 - Moore's Law 38, 354, 562
 - Mud to structure behavior 36–7, 167–210
 - discussion 165–167
 - application processing 170
 - interaction with environment 170
 - life expectancy 171
 - variability management 125
 - pattern abstracts 171–172
 - pattern descriptions 182–210
 - pattern discussion 170–179
 - Multi-platform support 424–425
 - Multicast 236, 496, 563
 - Mutex 114, 356, 372, 374, 377, 381, 388, 563
- N**
- Naming patterns 8–9
 - Network 563
 - network bridging 72–75
 - network interface 103, 563
 - Network management and control 202
 - Notification mechanisms 234–236
- O**
- Object 563
 - Object interaction 38, 40, 42, 399–421
 - discussion 397–399
 - decoupling 400
 - coherent coordination 400
 - communication overhead 400
 - pattern abstracts 401–402
 - pattern descriptions 405–421
 - pattern discussion 400–402
 - Object Request Broker (ORB) 98–100
 - abstract factories 120–122
 - client-side ORB core 110
 - changeability 101–102
 - concurrent event processing 112
 - connection management 108–111
 - dynamic configuration 121–124
 - dynamic link libraries (DLLs) 122
 - event demultiplexing 104–107
 - event queue 114–116
 - General Inter-ORB Protocol (GIOP) 105–106
 - interchangeable mechanisms 116–117

Object Request Broker (ORB) (*continued*)
 internal design 100–102
 layer concept 101–102
 portability 101–102
 scalability 111–114
 server-side ORB core 110
 stability 101–102
see also CORBA Component Model (CCM)
 Object-oriented/relational divide 87–9
 On-the-wire protocol 73, 403, 420, 572
 Operating system 564
 operating system kernel 114, 564
 Out-of-band 144–145, 155, 316, 424, 431, 444–446, 564

P

Packet 350, 564
 Parallelism 27, 84, 133, 235, 308, 323, 383, 502, 564
 Parameter 564
 Partitioning *see* Component partitioning; Layer
 Passive object 399, 564
 Pattern
 Composite pattern 11–12
 Compound pattern 11–12
 definition 565
 Pattern compound 11, 565
 Pattern form 9
 Pattern language xviii–xxxi, 33–52, 549–551
 Pattern sequence 565
 Pattern story 157–161
 Patterns, basics 3–15
 about patterns 4–6
 pattern complements 10–11
 pattern compounds 11–12
 pattern connections 15
 pattern form 9
 pattern languages 13–15
 pattern stories 12
 pattern sequences 12–13, 14
 Peer event handlers 265, 265–267
 Peer-to-peer 20, 216, 565

Platform 565
 Polymorphism 71, 117–118, 138, 408, 450, 456, 458, 535, 565
 Port 565
 Process 566
 Product line 66, 399–400, 566
 Protocol 566
 protocol stack 187, 212, 566
 Proxy 566
 Publish/subscribe middleware 27–29

Q

Quality of service 39, 160, 183, 185, 188, 226, 232, 246, 249–250, 257, 259, 307, 313–314, 335, 350, 355, 365, 367, 462, 494, 511, 513, 566

R

Race condition 38, 114, 296, 299, 362, 364, 375, 386, 394, 524, 567
 Recursive mutex 374, 567
 Refactoring 41, 43, 49–50, 218, 289, 550, 567
 Reification 299, 344, 413, 458, 567
 Relationship, definition 567
 Remote method invocation (RMI) 25, 30, 177, 214, 240, 518, 567
 Remote procedure call (RPC) 23, 25, 27, 29, 30, 239, 567
 Request event 48, 107, 112, 256, 259–6, 567
 Resource management 39, 40, 42, 473–532
 discussion 471–473
 performance 472
 scalability 472
 reliability 472–473
 flexibility 473
 updates 473
 transparent lifecycle control 473
 pattern abstracts 475–477
 pattern descriptions 488–532
 pattern discussion 476–485
 Responsibility 568
 Role 568
 Routing issues 231–233

S

Scheduler 366–367, 568
Semaphore 372, 374, 377, 388, 568
Serialization 260, 338, 340, 342, 357, 361,
369, 377, 568
Servant 367, 568
Server 568
Service 569
Shared memory 22, 108, 109, 119, 569
SOAP 29–30, 215
Socket 22, 103–109, 114, 171, 255, 569
Software architecture 569
Starvation 112, 522, 569
States, objects/methods/collections *see*
Modal behavior
Structured communication 21–24
Subclass 120, 138, 140, 142, 319, 397, 427,
453, 454, 569
Subsystem 570
Superclass 453–454, 569
Synchronization 38, 40, 42, 371–397, 569
discussion 369–373
pattern abstracts 375–376
pattern descriptions 380–397
pattern discussion 374–377
Synchronous I/O 107, 263, 360–361,
570
System 570

T

Telecommunication Management Network
(TMN) system 202
Template 570
Thread 570
Thread pool 86, 127, 261, 364, 570
Thread safety *see* Synchronization problem
Transmission control protocol (TCP) 96,
107–109, 111, 119–120, 187, 571
Transport endpoint 109, 571

Trial-and-error techniques 205
Two-way method invocation 354, 571
Type-safety 284, 571

U

Ucall 98, 106–107, 113, 571
User datagram protocol (UDP) 187, 571
User interfaces 75–78

V

Variation support 197
View 571
Virtual data objects *see* Future
Virtual machine 19, 21, 104, 213, 238, 571
Virtual memory 269, 354, 519, 571

W

Warehouse management process control
system 18, 58–59
automation pyramid 59
entity level 58–59
operational level 58
process control level 59
base-line architecture 65–94
communication middleware
architecture 95–127
distribution 62
domain model 41, 49–51, 60 438,
537–538, 540–541
material flow control 61–62
order management 60–61
receiving 61
shipping 61
stock management 60
storage organization criteria
(SOC) 135
topology management 62

- Warehouse management process control system (*continued*)
 - non-functional properties 557, 563–564, 569
 - availability 63
 - component distribution 273–274, 276, 308–309
 - component integration 64
 - dynamic configuration 63–64
 - human-computer interaction 64
 - performance 62
 - persistence 63
 - portability 63
 - scalability 63
 - warehouse topology
 - architecture 129–155
- Web service 25, 30–32, 212, 215
- World Wide Web (WWW) 29–31