

Index

A

- Abort event, 199–200, 202
- Abstraction effect, 282–283, 326–327
- Abstraction hierarchies, 56, 131–140
 - building personalized concept, 133–141
 - implementation, 210–211
- Abstraction principle, 83–84
- Abstraction relationship, 95
- Abstractions, 58
- Abstractions and pattern macros, 172–174
- Accessible information gap, 49–50, 278
 - event abstraction hierarchies, 131, 279–285
- Ack event, 129, 171
- Acknowledge action, 189
- Acknowledge action name, 154, 226
- Acknowledge event, 189, 192, 202, 225, 226, 256
- Action declarations, 151, 152
- Action name attribute, 151
- Action name filters, 190
- Action names, 149, 153
- Action set connection, 221
- Actions, 119, 153
- Activities
 - causes, 242
 - virtual, 56
- Activity components, 68
- Actual operations, 56
- Adaptation step, 90
- Adapters, 90, 156, 204, 205, 230, 300, 336
 - causal maps, 338–339
 - hard-coded causal maps, 338
 - sniffers, 337
 - target system, 286
- Add event, 128
- After temporal operator, 169
- Agent classes, 185
- Agents, 21
 - layers, 22
 - processing events, 22
 - reusing, 211–212
- Aggregate event, 17
- Aggregating events, 192
- Aggregation, 88, 127–128
 - causal relationships, 131
 - middleware events, 129
 - predefined rules, 133
- Aggregation relationship, 95
- Aggregation rules
 - CEP events, 134
 - completed transmissions, 136–137
 - defining, 140–141
 - Degraded Performance rule, 137–138
 - drill down, 137, 138
 - triggering, 137
- Agile systems, 17
 - agents, 21–23
 - decision making, 22
 - intelligence gathering, 22
- Aircraft EPA, 227
- Airplane event processing agent, 186–187
- Alarm component, 295, 296
- Alarms, 24
- Alert action name, 154
- Alert events, 182, 225
- Alerts, 24, 335
- All Brokers interface, 316
- Allocate portions of a trade message, 107
- Alternating bit protocol, 103
- Always constraints, 195–196, 201–203
- Analysis tools, 353–355
- Analyst component, 295
- Animation EPA, 316
- Anna, 198
- Annotations, 67–68, 72
- Any relationship operator (\sim), 164
- APIs (Application Program Interfaces), 14

- App servers, 343–346
 - Application layer, 12–13
 - Application programs, 6
 - Application-level activity view, 53
 - Application-level events, 12–13
 - Application-level instrumentation, 93
 - Application-level monitoring, 24–25
 - Applications
 - availability to users, 13
 - communicating between, 91
 - identifier, 254
 - preprocessed code, 93
 - Application-to-application talk, 301
 - Architectural diagrams, 71–72
 - Architectural policies, real-time monitoring, 80
 - Architectural structure, 68
 - Architecture, 224
 - encapsulating timeliness-checking EPN, 226–227
 - simple air traffic control, 227–230
 - type-correctness tests, 76–77
 - Architecture classes
 - EPNs (event processing networks), 221–230
 - semantics, 222–224
 - Architecture diagrams, 63, 70–72
 - Architecture keyword, 222
 - Arrays, 150
 - Arrival time, 97
 - ASPs (application service providers), 28
 - AssemblePackage behavior
 - annotations, 77
 - Assignment statement, 178
 - Asymmetric relationships, 95
 - Asynchronous communication, 91
 - Asynchronous processes, 35–37
 - At temporal operator, 169
 - ATM_Operations_1 filter, 190
 - ATM_Operations_2 filter, 190
 - Atomicity, 46
 - Attributes, interpretation of, 156
 - Attributes, of events, 155
 - Authentication, 26
 - Automated parallel processing, 34
 - Automated trading Web sites, 9
 - Automating trade agreements, 19
- B**
- B2B (business-to-business) products and services, 13
 - Backward search, 247–248
 - Banking systems, tracking causally related events, 101–102
 - Basic connections, 212
 - dynamic EPNs (event processing networks), 216–217
 - semantics, 213–214
 - Basic parallel connection, 213
 - Basic event patterns, 157–158
 - matching event, 159
 - naming parameters, 161–162
 - omitting parameters, 162
 - placeholders, 160–161
 - public attributes of events, 163
 - Basic sequential connection, 213
 - Behavior, 184
 - Behavior annotations, 73, 75
 - Behavior rules
 - body, 74
 - examples, 74–78
 - trigger, 74
 - Behavior specification, 63, 72–74
 - Behavioral constraints, 80
 - Benign observation, 90
 - Bid events, 36, 159
 - Binary operators, 164
 - Bindings, 159–161, 350–351
 - Bodies, 178–179
 - Body of a rule, 176, 177
 - BookFulfillment process, 76
 - Book-Request events, 76
 - Broker component, 295, 296
 - Broker performance, 313–316
 - Broker Response Time, 317
 - Broker Surveillance Monitor, 321
 - Broker Surveillance Monitor view, 319
 - BrokerReply event, 304
 - Building personalized concept abstraction
 - hierarchies, 133–141
 - Business level, 11–13
 - Businesses
 - electronic collaboration model, 18
 - opportunistic trading, 20–21
 - policy enforcement, 20
 - regulatory monitoring, 21
 - risk mitigation, 20

- Business-level applications, linking, 13
- Business-level events, 7, 12, 15–16
- Buy event, 306
- Buy_ahead events, 200
- BuyStock events, 306
- C**
- C⁴I (command, control, communications, computers, and intelligence) system, 5
- CallRecord events, 234
- CallRecord interface, 232
- CASE (Computer Aided Software Engineering), 329
- Causal ancestors, 119
- Causal event executions, 102–109
- Causal history, 101, 354–355
- Causal implication, 119
- Causal maps, 238, 242, 277, 305–311
 - adapters, 338–339
 - adding causality to events, 251
 - BuyStock events with causal attribute, 306
 - causal vector, 251
 - collaborative business agreement, 265–267
 - constructing, 246
 - defining, 244–246
 - generating output events, 251
 - inline transformation of events as they happen, 266
 - input execution as holding area, 252
 - linear causal chain, 252–253
 - local store, 251
 - managing event store, 251
 - mapping rules, 307–310
 - multithreaded system, 253–258
 - offline copy of collaboration execution
 - for analysis, 266
 - operation of, 250–251
 - orderly observation, 338
 - positioning, 268–269
 - processing input events, 251
 - role of, 246
 - single thread of control, 252–253
 - writing, 256–258
- Causal models, 97, 105, 107–109, 239, 302
 - accuracy, 258–260
 - activity between collaborating processes, 106–109
 - collaborative business agreement, 265
 - constructing formal, 245–246
 - defining, 244–246
 - defining as set of pattern pairs, 256
 - definition, 243
 - implementing, 250
 - incomplete, 258–259
 - inconsistencies, 259
 - need for, 242–243
 - resolving ambiguities, 248–250
 - rules, 243–244
 - single thread of control, 249–250
 - specifying with pattern pairs, 246–247
 - thread counter, 252
 - too strong, 259
 - too weak, 259
 - writing, 250
- Causal relationship, 244
 - aggregation, 131
 - correctness, 246
 - describing, 255
 - drill down, 243
 - dynamic, 241
 - event abstraction hierarchies, 282
 - event pattern maps, 193
 - event patterns, 115
 - between events, 109
 - explicit, 242–243
 - filters, 188
 - genericity across all executions, 246
 - input events, 212
 - output events, 212
 - prioritizing search, 243
 - reducing search space, 242–243
 - understanding system, 243
- Causal rules, 256
 - adding guards to right-side patterns, 248
 - backward search, 247–248
 - collaborative business agreement, 267–269, 270–271
 - left-side pattern, 247, 248
 - predicate guards, 249
 - right-side pattern, 247, 248
 - too strong, 259
 - too weak, 259
- Causal tracking, 109
 - example of, 46–49

- Causal vectors, 96, 97, 251
- Causality, 88, 100–102, 239, 240–242
 - augmenting time with, 103
 - dynamic, 241
 - explicit information, 245
 - implicit information, 245
 - interenterprise commerce, 104–106
 - network management, 104
 - tracking, 97
- Causality attribute, 156, 251
- Causality between events, 301
- Causality relationship, 95
- Causes operator ($P \rightarrow Q$), 165
- Cause-time axiom, 96, 244
- Consistent bank behavior, 263
- CD/Fulfillment process, 74, 75–76, 90
- CD-Request events, 75
- CEP (complex event processing), xv–xvii, 87–88, 91
 - broad applicability, xvi–xvii
 - flexibility, xvi–xvii
 - inferring higher-level events, 131
- CEP adapters, 316, 336–339
- CEP applications
 - events generated by, 94
 - system activity view, 210
- CEP events, 134
- CEP infrastructure, 330
- CEP infrastructure interface, 341
- CEP internal form, 93
- CEP matching, 347
- CEP system, 351
- Checker EPA, 226
- Class, 81, 184–197
- Class connections, 218
- Collaboration layer, 13
- Collaboration, between processes, 105–109, 261–275
- Collaborative business agreement, 262
 - broker requirements, 264
 - canceling trade when constraint violations happen, 272
 - causal dependency, 270
 - causal maps, 265–267
 - causal models, 265
 - causal rules, 267–269, 270–271
 - communication architecture, 264–265
 - consistent bank behavior, 263
 - constraint checking, 273, 274–275
 - constraints, 263–264, 269
 - flexible processes easily modified, 273
 - global timing constraint, 263
 - minimum requirements, 273
 - posets analysis, 269–274
 - positioning causal maps, 268–269
 - rules constraining activity, 263
 - upgrading, 273
 - violations of constraints initiating
 - rollback transaction, 273
- Collaborative events, 108
- Commit phase, 47
- Communication conformance, 83
- Communication flexibility, 7–9
- Communication scenario, 65
- Communications
 - middleware, 14
 - World Wide Web, 7–8
- Compilation of EPNs and dynamic reconfiguration facilities, 342
- Completed Data Transfer rule, 136–137
- Completed transmission view, 135
- Completed transmissions aggregation
 - rules, 136–137
- CompletedTran event, 130, 135–137
- Complex event pattern matchers, 348–351
- Complex event patterns, 163–173
- Complex event patterns, matching, 37
- Complex events, xv, xvi, 7, 95, 192
 - Add event, 128
 - aggregation, 127–129
 - creation of, 129–131
 - event pattern maps, 193
 - event pattern rules, 129–130
 - members, 127–129
 - pattern matchers, 348–352
 - small steps for processing, 212
- Complex patterns, 12
 - relational operators, 163–168
- Complex systems, tracing causal relationships, 97
- Component objects, 6
- Component-based systems abstraction
 - principle, 84
- Components
 - annotations, 67–68
 - infrastructure, 340–343
 - STS (stock-trading system), 295–296
- Composition, 208

- Computational causality, 95, 258
 - Computer viruses, 23
 - Computers, intrusions into, 23–25
 - Concept abstraction hierarchy, 280
 - Conditional statement, 178
 - Confirm an allocation, 107
 - Confirm event, 194
 - Confirm execution of a trade message, 107
 - Connect rules, 218
 - Connection generators, 218–221
 - Connection rules, 212–218, 316
 - Connections, 207, 224
 - basic, 212–214
 - between dual sets of actions, 221
 - dynamic EPNs (event processing networks), 216–217
 - guarded, 214–215
 - iterative creation, 218–219
 - multiple basic, 215–216
 - Connector behavior rule, 76
 - Connector class interfaces, 70–71, 73
 - Connector components, 68
 - Connector interfaces, 68, 72, 74
 - Connectors, 62
 - behavior annotation, 75–76
 - constraint annotation, 76
 - as distributor of events, 76
 - fan-out, 76
 - patterns of events, 72
 - reactions to various inputs, 72–74
 - types of events, 72
 - ConnectTime action name, 154
 - ConnectTime event, 225
 - Constraint checking, 124–125, 273, 274–275
 - Constraint keyword, 195, 199
 - Constraints, 74, 80, 100, 124–126, 177, 184, 195–204, 355
 - abstraction effect on, 282–283
 - checking with EPAs, 273
 - collaborative business agreement, 263–264, 269
 - DecomposeOrder process, 75
 - detecting violations, 324–326
 - event patterns, 124–126, 195–204
 - against inconsistent behavior, 124–125
 - Level of Service Check constraint, 125
 - maps for checking, 196–198
 - offline monitoring, 269
 - policies and regulations, 325–326
 - purpose of, 126
 - real-time monitoring, 269
 - reducing violations, 273
 - runtime checks, 125
 - STS (stock trading systems), 320
 - violation event, 125
 - Content filters, 128–129, 191
 - Content managers, 19
 - Content-based pattern matching, 167–168
 - Content-sensitive event patterns, 115
 - Context
 - event pattern rules, 179–180
 - tests for event patterns, 124
 - Context filters, 191–192
 - Context-based pattern matching, 168
 - ControlCenter EPA, 227
 - CORBA (Common Object Request Broker Architecture), 92
 - Corporate business process tracking
 - view, 54
 - Correlating events, 193
 - Counter, 189
 - Create action, 120, 121
 - Create parallel, 120
 - Create Priority Orders rule, 122–123
 - Creation rules, 217, 218
 - Customer component, 295, 296
 - Customer Initiated Transactions
 - view, 317
 - Customer trading activities, 312–313
 - Customers, keeping happy, 77
 - Cyber defense, 17
 - Cyber warfare, 15, 17, 23–26
- ## D
- DAGs (directed acyclic graphs), 100–101
 - DARPAnet, 26
 - Data components, 88
 - Data Transfer pattern, 117–118
 - Data types, 148
 - Debugging timing, 97–98
 - DecomposeOrder process, 75, 89
 - DegPerf event, 138
 - Degraded Performance rule, 137–138
 - Degraded performance view, 135
 - Denial-of-service attacks, 23
 - Dependence relationship, 95
 - Deposit event, 160, 164, 168, 170, 171, 190

- Design constraints, 63
 - examples, 74–78
- Designing processes, 65
- DetailInterface interface, 232
- Detecting patterns, 195
- Developing situation, 230–231
- Diagnostics driven from high-level
 - views, 320
- Disjoint union operator (\sim), 166
- Distributed, event-driven systems, 10
- Distributed behavior, 44
- Distributed information systems
 - application programs, 6
 - global monitoring, 209–210
 - scope of, 4–7
- Domain type, 193
- Drill down, 58, 137, 138, 355
 - causal relationship, 243
 - fabrication process, 290–291
- Drill-down diagnostics, 17
 - STS (stock trading systems), 320–324
- Driving events, 32–33
- DTP events, 153
- During temporal operator, 169
- Dynamic EPNs (event processing
 - networks)
 - basic connections, 216–217
 - connection generators, 218–221
 - connections, 216–217
 - creation rules, 217
 - fan-in connections, 217, 219–220
 - fan-out connections, 220–221
 - multiway connections, 217
 - sequential connection ($=>$), 217
- Dynamic process architecture, 78
 - diagrams, 81
 - interprocess communication, 79
 - runtime parameters, 79
- E**
- EAI (Enterprise Application Integration)
 - products, 13
- Ease of analysis, 52–53
- EDI (Electronic Data Interchange) data
 - format transformers, 105
- Efficient event analysis, 209
- Eiffel, 198
- Electronic collaboration model, 18
- Electronic commerce trading
 - documents, 19
- Electronic complex events, xvi
- Electronic enterprises, 29–30
 - process design, 61–62
- Electronic theft and destruction, 25
- Electronic trading regulatory
 - monitoring, 21
- E-mail virus attachment, 8 eMarketplaces,
 - 9, 19–20
- Enabling layer, 13
- End time, 96
- EnRouteSector architecture, 227–230
- Enterprise systems, 6
 - application layer, 12–13
 - business level, 11–13
 - collaboration layer, 13
 - distributed, event-driven systems, 10
 - enabling layer, 13
 - layered systems, 10
 - layers, 10–17
 - middleware layer, 13–14
 - network layer, 14–15
 - policies and regulations, 20–21, 325–326
 - protocol for collaboration, 107
 - strategic planning level, 11–13
 - tracking events, 15–16
 - user interfaces, 12
 - views, 53–55
- Enum (enumeration type), 150
- EPAs (event processing agents), 155, 175,
 - 176, 184, 224
 - in actions, 179
 - adapters, 204, 205
 - adding causality attribute to events, 246
 - causal relationships between activities,
 - 185–186
 - class interface, 184–185
 - connecting, 212–216
 - constraint violations, 273
 - constraints, 177
 - definition, 184–185
 - event pattern constraints, 195–204
 - event pattern filters, 187–192
 - event pattern maps, 192
 - event pattern rules, 177–184
 - executing independently as separate
 - threads of control, 185

- filters, 177
- indicators, 204, 205
- input events, 186
- maps, 177
- monitoring structure, 176–177
- offline, 212
- one-at-a-time event processing, 186
- organizing, 221
- out actions, 179
- partial class interfaces, 204–205
- semantics, 185–187
- EPL (event pattern languages), 113–114, 130, 346–348
 - basic requirements, 146–147
 - notational simplicity, 146
 - pattern matcher, 146
 - power of expression, 146
 - precise semantics, 146
 - RAPIDE, 147–158
 - scalable pattern matching, 146
- EPNs (event processing networks), 176–177, 277, 316
 - architecture classes, 221–230
 - basic connections, 212–214
 - checking timely message delivery, 224–226
 - common structures of, 208–212
 - composed dynamically, 212
 - connections, 207, 212–216
 - dynamic, 216–221
 - efficient event analysis, 209
 - encapsulating, 222
 - flexibility, 211–212
 - global monitoring of distributed systems, 209–210
 - guarded connections, 214–215
 - implementation of abstraction hierarchies, 210–211
 - irrelevant events, 209
 - multiple basic connections, 215–216
 - network viewing, 230–235
 - rules engines, 208
 - STS (stock trading systems), 317
 - visual tools for constructing, 234–235
- Equipment monitoring, 132–133
- ETC (Electronic Trade Confirmation), 107, 263
 - ETC (Electronic Trade Confirmation) system, 46–49
- Event abstraction hierarchies, 131–141, 280, 355
 - abstraction effect on constraints, 282–283
 - accessible information gap, 279–285
 - action specification corresponding to each activity, 280
 - application-to-application talk, 301
 - applying to target system, 286–287
 - broker performance, 313–316
 - building, 299–316
 - causal maps, 305–311
 - causal models, 302
 - causal relationships, 282
 - causality between events, 301
 - CEP events, 134
 - customer trading activities, 312–313
 - defining, 134–136
 - defining aggregation rules, 136–138, 140–141
 - defining industry standards, 284
 - definition, 280
 - detail requirements, 283–284
 - easily modifying definition, 281–282
 - event pattern maps, 280
 - fabrication process, 287–291
 - induced causality, 282
 - inserting and removing levels, 283
 - level 1, 300–301
 - level 2, 301–311
 - level 3, 311–316
 - levels in definition arbitrarily chosen, 282
 - modifiability, 283–285
 - precisely defining, 281
 - sequence of levels of activities, 131
 - set of event aggregation rules for each level, 131
 - standard for domains, 285
 - standardizing, 284
 - stock-trading activity, 138–141
 - STS (stock-trading system), 298–316
 - unified EPN, 281
 - variability of hierarchical views, 284
 - viewing fabrication line, 132–133
 - viewing network activity, 134–138

- Event adapters, 300, 335–336
- Event aggregation, 16–17, 354
- Event causality, 10, 95
- Event cloud, 33
- Event communication servers, 342
- Event creation, 179
- Event driven, 4, 29
 - views, 54–55
- Event type, 151
- Event inferences, 110
- Event logs, 45, 46
- Event monitoring, 44–45, 335–336
- Event pattern constraints, 124, 195–204
 - always constraints, 201–203
 - definition, 195
 - never constraints, 199–201
 - semantics, 195–198
 - state constraints, 203–204
- Event pattern filters, 187–192
 - definition, 187–188
 - semantics, 188–190
- Event pattern languages, 346–347
- Event pattern maps, 192–193
 - event abstraction hierarchies, 280
 - semantics, 193–195
- Event pattern matching, 347–351
- Event pattern rules, 175
 - aggregating network node alerts, 182–183
 - bodies, 178–179
 - body, 176, 177
 - changing state of parent, 181
 - complex events, 129–130
 - context, 179–180
 - creating new events, 181
 - definition, 178
 - examples, 182–184
 - generate statements, 181
 - pass statement, 181
 - semantics, 180–181
 - trigger, 176, 177, 179
 - visibility laws, 179–180
- Event patterns, 99–100, 114–117, 157–172, 355
 - basic patterns, 157–158
 - causal relationships, 115
 - completed message delivery on messaging middleware, 118–119
 - constraints, 124–126
 - content-sensitive, 115
 - context tests, 124
 - Create Priority Orders rule, 122–123
 - describing, 113–114
 - event types, 100
 - filters, 115
 - GUI (graphical user interface), 114
 - instance, 157
 - Late Data Transfer Warning rule, 121–122
 - management policy requiring action to be taken, 120
 - matching events, 157–158
 - multiple relationships, 115–116
 - placeholders, 158–163
 - predefined attribute, 160
 - reactive rules, 119
 - recognizing corresponding pairs of Publish/Receive events, 119
 - relational operators, 100
 - replacing variables, 117
 - rule table, 120–121
 - rules, 119–123
 - specifying causal model from, 246–247
 - successful network data transfer, 117–118
 - types of searching, 113–114
 - variables, 100
- Event processing, 90
- Event store data structures, 257–258
- Event storm, 45
- Event templates, 116–117
- Event timing, 354
- Event types, 100, 148, 151–153, 161
- Event-based simulation, 65–66
- Event-driven graphical meters, 230
- Event-driven information systems, xv
- Event-driven viewers, 286
- Events, xv, xvi, 4, 6, 88, 90
 - action declarations, 151, 152
 - action name attribute, 151
 - action names, 149
 - adaptation step, 90
 - adapting, 92–93
 - aggregating, 192
 - aggregation, 127–129

- aggregation relationship, 95
 - application-level, 12–13
 - aspects, 88
 - attributes, 354
 - business-level, 7, 12, 15
 - causal ancestors, 119, 266
 - causal attributes, 261–262
 - causal history, 101, 354–355
 - causal relationship, 95, 244
 - causal tracking, 9
 - causal tracking example, 46–49
 - causal vector parameter, 242
 - causal vectors, 97
 - causality, 100–102, 240–242
 - causality attribute, 157, 239, 242, 251
 - causality relationship, 95, 244
 - CEP (complex event processing), 91
 - cloud of, 28–29
 - complex patterns of, 12
 - components, 355
 - consumers of, 341
 - correlating, 193
 - created as instrumentation, 93–94
 - creation of, 90–94
 - data components, 88
 - data in, 33
 - definition of, 88
 - determining causality between, 247–248
 - difficulty in tracking, 9
 - dynamic causes, 97
 - ease of analysis, 52–53
 - efficient analysis, 209
 - financial trades level, 58
 - form, 88–90
 - generated by CEP applications, 94
 - genetic parameters, 96
 - grouping at lower levels, 16
 - happening on global scale, 28
 - horizontal causality, 10
 - identifying for monitoring, 245
 - instrumentation, 91
 - irrelevant, 209
 - IT layer (information technology layer), 91
 - message, 98
 - monitoring, 7
 - monitoring between participants, 47
 - observation step, 90
 - orderly observation, 109–110
 - parameters, 354
 - picking out sets of related, 45
 - posets, 100–102
 - predefined attributes, 155–157
 - private attributes, 155–156
 - public attributes, 155, 163
 - real-time recognition of complex patterns, 7
 - relationships between, 94–96
 - relativities, 52
 - relativity, 88–90
 - relevance, 52
 - significance, 88–90
 - sources of, 90–91
 - specifying for monitoring, 255
 - strategic-level, 7
 - subtypes, 151, 152–153
 - suppliers of, 340
 - thread attribute, 249
 - time relationship, 94–95
 - timestamps, 94, 96–97, 98
 - timing of, 96–97
 - timing relationship between, 97–100
 - tracking causal relationships, 21
 - tracking in enterprise system, 15–16
 - transaction Id, 98
 - transaction identifier, 97
 - transaction protocol level, 58
 - triggers, 32
 - understandability, 52
 - unique identifier field, 89
 - vertical causality, 10
 - violation, 12
 - virtual, 12
 - Exceptional situation, 33, 272
 - Exceptions, 39–41
 - Execution keyword, 153
 - Execution types, 153–154
 - Explicit causal relationship, 242–243
- F**
- FAA events, 153
 - Fabline operation, 133
 - Fabrication line
 - equipment monitoring, 132–133
 - fabline operation, 133
 - factory management, 133

- Fabrication process
 - drill down diagnostic, 290–291
 - event abstraction hierarchies, 287–291
 - fabline workflow, 289
 - implementation of hierarchy, 290
 - middleware communication, 289
 - middleware layer, 287
 - personal views, 288–289
 - point-to-point communication, 289
 - product disposition, 289
 - publish/subscribe messaging technology, 287
 - Factory management, 133
 - Failed-Two-Phase-Commit event, 58
 - Fan-in connections, 219–220
 - Fan-out connections, 220–221
 - Fan-out connectors, 76
 - Fed component, 295
 - Filter classes, 187
 - Filter keyword, 187
 - Filters, 177, 184, 230, 232–233
 - action name, 190
 - causal relationship, 188
 - content, 191
 - context, 191–192
 - definition, 187–188
 - efficiency, 188
 - event patterns, 115, 187–192
 - generate statements, 188
 - pass rule, 188, 189
 - rules, 189
 - semantics, 188–190
 - state change rule, 188
 - timing relationship, 188
 - Finalize a trade message, 107
 - Financial trades level, 57
 - For loops, 179, 218, 219
 - Form, 88–90
 - Function call, 179
- G**
- Generate pattern, 181
 - Generate rules, 181
 - Generate statements, 178–179, 181, 183, 188, 193, 218–219
 - Genetic parameters, 96
 - GetQuote event, 248
 - GIDO (general intrusion detection object), 334
 - Global communication, 7–10
 - Global communication spaghetti, 9
 - Global content management, 19
 - Global eCommerce Web, 9, 18
 - Global electronic trade, 17, 18–21
 - Global event cloud, 28–29
 - operating in, 30–33
 - unnoticed, 80
 - Global monitoring of distributed systems, 209–210
 - Global timing constraint, 263
 - Global trading hub of all hubs, 19–20
 - Global type context, 148, 149
 - Government information systems, 5
 - Graphical EPA, 316
 - Guard, 214
 - Guarded connections, 214–215
 - Guarded patterns
 - content-based pattern matching, 167–168
 - context-based pattern matching, 168
 - temporal operators, 169
 - GUI (graphical user interface) event patterns, 114
- H**
- HandOff action, 227
 - HandOn action, 227
 - Header interface, 232
 - Hierarchical structure, 56, 57, 58
 - Hierarchical viewing
 - example of, 57–59
 - hierarchical structuring, 56, 57
 - operational description, 56
 - Successful-Two-Phase-Commit event, 57
 - Higher-level instrumentation, 93
 - High-level events, 12, 16
 - High-level system performance view, 54
 - Horizontal causality, 10
 - Hub of all hubs, 19–20
- I**
- I Love You virus, 8
 - IETF (Internet Engineering Task Force), 24
 - Immediate cause operator (\triangleright), 174

- In actions, 179, 180, 184–185, 214, 219, 222
 - In events, 214
 - in_Action in action, 213
 - Inconsistent event report, 271
 - Independence operator ($P \parallel Q$), 165
 - Indicators, 204, 205
 - Induced causality relationships, 282
 - Information buses, 13
 - Information gaps, 49, 278–280
 - bridging, 285–286
 - examples of, 50–51
 - horizontal dimension, 50
 - STS (stock-trading system), 296–298
 - vertical dimension, 50
 - Information processing, 3
 - Information processing systems, 3–4
 - Infrastructure, for CEP
 - components, 340–343
 - interfaces, 340–343
 - interfacing to target system, 335–336
 - observed events, 330–336
 - off-the-shelf, 343–346
 - runtime, 339–340
 - uncertainty principle, 330
 - Initiate a trade message, 107
 - Input events
 - causal relationships, 212
 - Count parameter, 253
 - EPAs (event processing agents), 186
 - InputEvent class, 89
 - Instrumentation, 91, 93–94
 - Intelligence gathering, 22
 - Interenterprise commerce causality, 104–106
 - Interface annotations, 73
 - Interface communication architecture simulation, 77
 - Interface communication architectures, 68–70
 - Interface conformance, 83
 - Interface
 - of a connector, 68–72
 - of a constraint, 195
 - of an EPA, 184–187
 - of a filter, 187
 - of a map, 192
 - of a process, 68–72
 - Interfaces, 68, 184
 - communication structure between, 69
 - compatibility, 77
 - compilation of EPNs and dynamic reconfiguration facilities, 342
 - conformance, 83
 - event communication servers, 342
 - execution management, 341–342
 - functionality of, 341–343
 - hierarchy management, 342
 - infrastructure, 340–343
 - layered architectures, 70
 - pattern matching and rule execution engines, 342
 - plug-and-play techniques, 70, 81–84, 224
 - Internet
 - communication flexibility, 8–9
 - new activities, 17–18
 - society's dependence on, 25
 - Interprocess communication, 79
 - Intruders, 23
 - Intrusion detection EPN, 235
 - Intrusion event, 128
 - Intrusions
 - into computers, 23–25
 - views, 231–232
 - Intrusion-warning events, 131, 334–335
 - Invariant, 244
 - IPAccesstype array, 233
 - IPCountType array, 233
 - ISO (International Standards Organization), 58
 - IT (information technology) layer, xv
 - monitoring tools, 44, 336–339
 - subscribing to messages, 91
- J**
- J2EE (Java 2 Platform Enterprise Edition) servers, 330
 - Java, 294
- K**
- Keep customer happy rule, 77–78
- L**
- Late Data Transfer Warning rule, 121–122

- Layered architectures, 70, 224
 - abstraction principle, 83–84
 - implementation as code, 81
 - lower-level architecture, 81
- Layered interface communication
 - architecture, 70, 222–224
- Layered systems, 10
- Layering plug-and-play, 224
- Layers, 10–17
 - agents, 22
- Level of Service Check constraint, 125
- Levels, 56
- Loggers, 156
- Logical operators, 166
- Lower-level events, aggregation of sets or groups of, 16–17
- M**
- M502 event, 269–270, 271, 274–275
- M502 message, 109
- M513 event, 270, 273, 274–275
- M514 events, 270, 271, 272, 273
- M514 message, 109
- M515 events, 271
- M515 message, 109
- M517 event, 271
- M517 message, 109
- Macro expansion, 172, 173
- Mail requests for quotations, 183
- Map class, 193
- Map keyword, 193
- Maps, 177, 184, 230
 - for checking constraints, 196–198
 - detecting suspicious IP address scans, 233–234
- Market and STS (stock-trading system), 295
- MarketTransactionsType type, 200
- Matching bids and RFQ process, 159–160
- Max time units, 192
- Members, 127–129
- Memory leak, 241
- Message routing timing requirement, 98–99
- Message-oriented middleware, 91
- Messages, xvi, 4
- Messaging protocols, 26
- Messaging systems information buses, 13
- Middleware
 - aggregation of events, 129
 - communications, 14
 - publish/subscribe, 91, 98
- Middleware layer, 13–14
- Military information systems, 5
- Monitoring, 134
- Monitors, 205
- MsgId message identifier, 254
- MT502 events, 121
- MT502 message, 107, 108, 263
- MT509 message, 263
- MT513 events, 121
- MT513 message, 107, 108, 263
- MT514 message, 107, 108, 263
- MT515 message, 107, 108, 263
- MT517 message, 108, 263
- Multienterprise financial trading
 - system, 4
- Multievent matcher, 349
- Multiple basic connections, 215–216
- Multiple viewing, 53–56
- Multiple-event matching with context, 346–347
- N**
- Negotiation protocol, 80
- .NET, 330
- NetFilter action name filter, 225–226
- NetMngmt type, 225
- Network crashes, 14
- Network layer, 14–15
 - system monitoring, 44–45
- Network layer tools, 45
- Network management, 44
 - causality, 104
 - products, 14
- Network monitoring domain, 230
- Network monitoring scalability, 235
- Network monitoring system and intrusion warning event, 334–336
- Network problems view, 54
- Network router monitor observed events, 334–335
- Network viewing
 - EPNs (event processing networks), 230–235
- Network-level detection products, 24

- Network-level instrumentation, 93
 - Network-level monitoring, 45–46
 - Networks
 - degradation of performance, 231
 - denial-of-service attacks, 23
 - event abstraction hierarchy for viewing activity, 134–138
 - event logs, 45
 - keeping running, 14
 - not solving problems with network-level monitoring, 45–46
 - views, 231
 - Never constraints, 124, 195, 196, 199–201, 200
 - Never regulation, 325
 - New operation, 185
 - New statement, 218
 - NewOrder message, 89
 - Notational simplicity, 146
 - Notify(...) event, 183
 - Nuisance activities, 23
- O**
- Object-oriented programming
 - languages, 83
 - Observable system, 110
 - Observation step, 90
 - Observed events
 - network router monitor, 334–335
 - publish/subscribe middleware, 331
 - Offline EPAs (event processing agents), 212
 - Offline monitoring constraints, 269
 - Off-the-shelf infrastructure, 343–346
 - One-at-a-time event processing, 186
 - On-the-fly process evolution, 37–39
 - Open commerce, need for, 25
 - Open electronic society and cyber warfare, 23–26
 - Open enterprise, xv, 17, 28
 - Operating in global event cloud, 30–33
 - Operating systems
 - breaking and entering, 24
 - pattern languages, 114
 - Operational description, 56, 58
 - Operations, virtual, 56
 - Opportunistic trading, 20–21
 - ORBs (Object Request Brokers), 13, 92
 - Order events, 115, 152, 194
 - Order pattern, 163
 - Orderly observation, 109–110, 338
 - OSI (Open System Interconnect)
 - reference model, 58
 - Out actions, 179, 180, 184–185, 214, 219, 222
 - Out events, 213, 214
 - out_Action action, 213
 - out_Action event, 215, 217
 - Output events causal relationships, 212
 - Outsourcing, 9
- P**
- Parallel event pattern rules, 177
 - Parallel operator (||), 164
 - Parallel processes, 35–37
 - Parallel reactive rules, 120
 - Partial class interfaces, 204–205
 - Partial-ordering, 95
 - Pass rules, 181, 188, 189
 - Pass statement, 178, 179, 180, 181, 193
 - Pattern languages, 114, 131, 146–147
 - Pattern macros
 - recursive, 173–174
 - shortening notation, 172–173
 - Pattern matchers, 146
 - complex events, 348–351
 - Pattern matching, 113–114
 - content-based, 167–168
 - context-based, 168
 - placeholders, 158–163
 - STRAW-EPL, 117
 - Pattern pairs, 246–247, 302–305
 - Pattern-based monitoring, 279
 - Patterns
 - detecting, 195
 - global type context, 148, 149
 - local type context, 149
 - matching, 149
 - type consistency, 149
 - Perl (Practical Extraction and Reporting Language), 346
 - Personalization concept abstraction
 - hierarchies, 133–141
 - Personalized high-level views, 320

- Placeholders
 - basic patterns, 160–161
 - bindings, 159–161
 - misordering, 161–162
 - pattern matching, 158–163
- Plug-and-play, 67, 81–83, 224
 - layering, 224
 - with reduced risk of errors, 84
 - techniques, 70
- PM (Pattern macro), 172
- Point-to-point direct communication, 288
- Policy enforcement, 20
- Polling phase, 47
- Posets (partially ordered set of events),
 - 100–101, 102–104, 106, 163–167, 354
 - collaborative business agreement, 265–266
 - collaborative business agreement analysis, 269–274
 - real-time, 102–109
 - recording incomplete transactions, 271
 - relationship between, 164
 - templates, 114–116
 - triggering events, 181
 - use-cases, 266
- POV (poset viewer), 321–322
- Power of expression, 146
- Powerful pattern languages, 131
- Precise semantics, 146
- Predefined
 - aggregation rules, 133
 - attributes, 155–157
 - types, 149–150
- Predictive monitoring, 46
- PriorityOrder event, 123
- Privacy, 15
- Private attributes, 155–156
- Problem of immediate interest, 51–52
- Problem-relevant events, 52
- Problem-relevant information, 43–44, 51–53
- Process architect, 351, 353
- Process architectures, 62
 - architectural structure, 68
 - architecture diagram, 63
 - architecture diagrams, 70–72
 - behavior specification, 63, 72–74
 - component substitution, 67
 - consensus on acceptability, 65
 - constituents, 67–74
 - constraint constituent, 66
 - conveying process design to stakeholders, 65
 - correcting process design, 65–66
 - design constraints, 63, 66, 74
 - designing processes, 65
 - implementing process, 66
 - integrating process, 66
 - interface communication architectures, 68–70
 - modifying process system, 66–67
 - monitoring runtime operation of upgraded system, 66
 - process interfaces, 67
 - process lifecycle roles, 63–67
 - simulating new process on event streams, 65–66
 - testing process, 66
- Process class, 68
- Process class interfaces, 70
- Process design, 61–62
- Process interfaces, 67, 68, 72–73, 81
- Process lifecycle, 63–67
- Processes, 62
 - asynchronous, 35–37
 - class, 73
 - collaborating activity between and causal modeling, 106–109
 - driving events, 32–33
 - event pattern behavior, 72
 - exceptional situation, 33
 - exceptions, 39–41
 - humans in control of, 38–39
 - implementation, 81
 - increased situational complexities, 33
 - on-the-fly evolution, 37–39
 - parallel, 35–37
 - reaction to various inputs, 72–74
 - reduced time scales, 33
 - roles of architecture in lifecycle, 63–67
 - simulation, 65–66
 - subprocesses, 81
 - type, 73
 - types of events, 72
- ProcessOrder process, 30–33
- Product Shipped event, 32
- Public attributes, 155
- Publish event, 98, 99, 129

- Publish/subscribe middleware, 91, 98
 - content-based routing activity, 128–129
 - observed events, 331
- PurchaseOrder event, 248
- Put event, 201
- R**
- Range type, 193
- RAPIDE-EPL, xviii, 146, 348
 - attributes of events, 155–157
 - basic data types, 147
 - basic event patterns, 147, 157
 - complex event patterns, 163–173
 - context, 147
 - data types, 148
 - event types, 147, 148, 151–153
 - execution types, 148, 153–154
 - guarded patterns, 167–169
 - mathematical semantics, 147
 - pattern macros, 147, 172–174
 - pattern operators, 147, 166
 - predefined attributes, 155–157
 - predefined types, 149–150
 - relational operators, 166
 - repetitive patterns, 169–171
 - strong typing, 147
 - strongly typed, 148–155
 - structured types, 150–151
 - subtyping executions, 155
 - syntax, 147–148
 - temporal operators, 147, 169
 - type context, 148–149
 - writing event patterns, 161–163
- Rapidly reconfiguring running monitoring application, 230
- Reactive rules, 32, 73–74, 119–124, 177–183, 355
 - actions, 119
 - parallel, 120, 177
 - rule table, 120–121
 - Select Vendor process, 36
 - sequential, 120, 177
 - triggers, 119, 177
- Real-time decision making, 21, 52
- Real-time monitoring constraints, 269
- Real-time posets (partially ordered set of events), 102–109
- RecAck event, 129, 130, 137
- Receive event, 129
- ReceiveOrder pattern, 183
- Receiver
 - acknowledging data, 135
 - acknowledging message, 254
- Receives event, 98, 99
- Records, 150
- Recursive pattern macros, 173–174
- Regulatory monitoring, 21
- Related events, 45
- Relating views, 53, 57
- Relational operators, 100
 - binary operators, 164
 - complex patterns, 163–168
 - logical operators, 166
 - RAPIDE-EPL, 167
 - set operators, 166
 - structural operators, 165–166
- Relationships
 - asymmetric, 95
 - partial-ordering, 95
 - transitive, 95
- Relativities, 52
- Relativity, 88–90
- Relevance, 52
- Remove event, 201
- Repetition counter variable, 171
- Repetition pattern (*), 170
- Repetition profile, 169–170
- Repetitive patterns, 169–171
- Reply pattern macro, 173
- Report event, 178
- Requests causing replies, 47
- ReSend events, 116, 137, 138
- ReSends event, 129
- Reset action, 189
- Reset event, 189
- Return expression, 179
- RFQ events, 36
- RFQs (requests for quotations), 35, 36
 - matching bids, 159–160
- Right-side patterns predicate guards, 249
- Risk mitigation, 20
- Rules confusion, 353
- Rules engines, 208, 351
 - organizing rules, 209
 - pattern triggers, 209
 - patterns of events, 36
 - rules manager, 352
- Rules management, 351–353

- Rules manager, 352, 353
- Rules reduction, 352
- Runtime checks, 125
- Runtime drill-down diagnostics, 41
- Runtime infrastructure, 339–340
- runtime monitoring, 66
- RV, 294
- S**
- Safety constraints, 80
- Satisfied event, 200
- Scalability
 - network monitoring, 235
 - pattern matchers, 349
- Scalable pattern matching, 146
- Schliemann, Heinrich, 11
- SEC Regulation Checkers, 321
- Security, 15
 - denying access, 25
 - intrusion detection EPN, 235
 - intrusions, 231–232
 - locking things up, 25
 - traditional technology, 25
- Select Vendor event, 32
- SelectVendor process, 35–36, 39
- Send action name, 154, 226
- Send event, 116, 129, 130, 137, 192, 199–200, 225, 226, 256
- Send/Ack system, 254
 - defining causal model as set of pattern pairs, 256
 - describing causal relationship between events, 255–256
 - specifying events monitored, 255
 - writing causal maps, 256–258
- Sender, 135
 - acknowledgment from receiver, 135, 254
- Send(O) event, 171
- SendRFQ event, 183
- Sequence of levels of activities, 131
- Sequential event pattern rules, 177
- Sequential reactive rules, 120
- Servers, authentication of, 26
- Set event, 201
- Set of event aggregation rules for each level, 131
- Set operators, 166
- SetofEvents object, 201
- Ship Order event, 32
- Shipping complete orders only rule, 78
- Significance, of an event, 88–90
- Simple air traffic control architecture, 227–230
- Simulation, of an architecture, 65–66, 77
 - before implementation, 84
- Single-event, content-based matching, 346
- Single-event matcher, 349–350
- SLAChecker (service-level agreement checker), 226–227
- Sniffers, 338, 339
- Sniffing events, 337
- SOES (small order execution system) regulations, 326
- Spam, 23
- SPARC V9 events, 153
- SpecialOffer event, 123
- SpecialOrder process, 79–80, 82
- Start actions, 185
- Start time, 96
- State change rule, 188
- State constraints, 196, 203–204
- State variables, 186
- Statements, 177
- Static architectures, 78–79
- Static EPN architectures, 213
- Stock component, 295
- StockCustomer component, 296
- StockMarket component, 295
- StockTrade Message Test pattern, 118–119
- Stock-trading activity and event
 - abstraction hierarchies, 138–141
- Strategic planning level, 11–13
- Strategic-level events, 7
- STRAW-EPL, 116–117
 - writing patterns, 117–119
- Strawman EPL (event pattern language), 116–119
- String pattern matching, 346
- Strong induced causality, 282
- Structural operators, 165–166
- Structural relational operators, 169
- Structured types, 150–151
- STS (small stock-trading systems)
 - abstraction effect, 326–327
 - activities organized by levels, 298–299

- animation of application-level
 - communication, 296
 - behavior of applications, 297
 - broker performance, 313–316
 - Broker Response Time, 317
 - Broker Surveillance Monitor view, 319
 - causal model with pattern pairs,
 - 302–305
 - causality at application level, 297
 - components, 295–296
 - constraints, 320
 - Customer Initiated Transactions
 - view, 317
 - customer trading activities, 312–313
 - detecting constraint violations, 324–326
 - diagnostics driven from high-level
 - views, 320
 - drill-down diagnostics, 320–324
 - dynamic aspects of, 294
 - EPNs (event processing networks), 317
 - event abstraction hierarchies, 298–316
 - event-driven viewers, 317
 - human control, 319–327
 - implementing hierarchical viewing,
 - 316–319
 - information gaps, 296–298
 - market, 295
 - monitoring, 297–298
 - performance of applications, 297
 - personalized high-level views, 320
 - publish/subscribe middleware, 294
 - statistics on activity at application
 - level, 297
 - users, 294–295
- Subprocesses, 41, 81, 83
- Subtyping executions, 155
- Successful-Two-Phase-Commit
 - event, 57, 58
- Synchronous communication, 92
- Systems
 - monitoring network layer, 44–45
 - views, 53
- T**
- Target system
 - adapters, 286
 - applying event abstraction hierarchy,
 - 286–287
 - hierarchical viewing, 287
 - interfacing to infrastructure, 335–336
- TCA (terminal control area), 80
- Templates
 - pattern matching, 114
 - posets (partially ordered set of events),
 - 114–116
- Temporal operators, 169
- Terminate action, 185
- Terminate keyword, 218
- Terminated event, 200
- Time, 88, 97–100
 - augmenting with causality, 103
- Time relationship, 94–95
- Timelines, 274–275, 355
- Timeout action, 189
- Timeout event, 129, 137, 171, 271
- Timestamps, 94, 96–97, 163
- Timing constraint, 75
- Timing operator ($P < Q$), 165–166
- Timing operators, 179
- Timing relationship filters, 188
- Timing requirement, 98
- Trade agreements, automating, 19
- Trade event, 194, 195
- Trading documents, 19
- Trading hubs, 19–20
- Transaction managers, 19
- Transaction master, 46–47
- Transaction protocol level, 57
- Transaction system and causal modeling,
 - 105–109
- Transactions, 46, 272
- Transfer action, 161
- Transfer event, 190
- Transitive, 95, 100, 244
- Transitive relationships, 95
- Triggering events, 32, 181
- Triggers, 32, 74, 119, 176, 177, 179
- Two-phase commit protocol, 46
- Type context, 148–149
- Type definitions, 150–151
- Type-correctness tests, 76–77
- Types, 185
- U**
- UML, 70
- Uncertainty principle, 110–111, 330

- Understandability, 52
- Union operator (U), 166
- Unsolicited e-mail, 23
- User interfaces, 12
- Users
 - application availability, 13
 - STS (small stock-trading system), 294–295
- V**
- Valid Order event, 32, 36
- Variables, 100
- Vendor Selected event, 32, 39
- Vertical causality, 10, 15–16, 131, 157, 321
- VHDL, 198
- Viewing, 53–55, 134
 - a fabrication line, 123–133
 - hierarchical, 56–59, 316–319
 - network activity, 134
 - stock trading activity, 136
- Views, 231
 - coordinating, 55–56
 - creation of multiple, 55–56
 - event driven, 54–55
 - humanly understandable forms, 55
 - intrusions, 231–232
 - modifying, 55
 - network traffic, 232
 - problem of interest, 54
 - relating, 53, 57
 - relevant events for automated decision making processes, 55
- Violated event, 226
- Violation events, 12, 125, 199
- Virtual activities, 56
- Virtual events, 12, 16
- Virtual operations, 56
- Virus attachments, 8
- Viruses, 23
- Visibility laws and event pattern rules, 179–180
- Visio, 70
- Visualizers, 205
- W**
- Wait_M515 event, 109
- Warning action name, 149, 154
- Warning events, 126, 148, 151–152, 225
- Watchdog, 124
- Watchdog event, 270, 271
- Web sites
 - automated trading, 9
 - denial-of-service attacks, 23
 - trading hubs, 19–20
- Where clause, 167, 203
- Where condition, 306
- While loops, 179, 218
- Who’s-talking-to-whom view, 53–54
- Withdraw events, 164, 168, 169, 190
- Workflow
 - activities carried out in, 33–34
 - specification, 31
- Workflow rules, 31–32, 33
- Workflow rules engine, 31
- World Wide Web, 7–8
- X**
- XML (Extensible Markup Language), 19