

Index

A

activity diagrams 20, 26, 31, 39, 42, 52, 56, 153, 272, 537
 antipattern 62, 73, 76
 application deployment 402–416
 application specification language (ASL) 402–433, 505
 architectural alternatives, basic 161–199, 203–210
 architecture and development pattern 64
 architecture description language (ADL) 107, 108, 261, 273
 architecture patterns 260, 332, 538
 aspect-oriented programming (AOP) 269, 270, 321–333, 499, 509, 540
 AspectWerkz 321, 329, 509
 avatar 223, 242, 245, 247, 249

B

batch processes 381–395
 behavior, non-deterministic 269
 behavioral analysis 290, 292, 293, 294, 326
 block diagrams 26, 27, 37, 39, 46
 business model 445, 446
 business process 3, 18, 20–30, 39, 41, 53, 54, 56, 57, 66, 67, 70, 111, 253, 363, 475, 506, 537, 538, 545
 business process engineering 21, 22, 23, 26, 53
 bytecode counting 317

C

common object request broker architecture (CORBA) 86, 103, 105, 110, 112, 128, 129, 130, 259–281, 309–369, 475, 478, 485–497, 500–544
 complexity indicators 81, 89, 98
 component diagrams 31, 34, 46, 226, 272
 component frameworks 102–117, 125, 336
 component middleware 105, 110, 114, 125, 126, 127, 128, 132, 342, 474, 475, 476, 477, 478, 479, 482, 483, 496, 498, 504, 516
 computer aided software engineering (CASE) 107, 482
 configuration management 3, 6, 8, 14, 107, 212, 223, 259, 281, 378, 515
 constraint checking 372, 373, 377
 constraint satisfaction problem 384, 395
 control engineering 21, 22, 26, 35, 40, 45, 47, 223
 CORBA component model (CCM) 105, 110, 112, 114, 130, 272, 345, 474–504, 537

D

data management 405, 455, 473, 544
 data store service 69, 70
 deadlock 30, 184, 293, 295, 325, 326
 design decision 107, 189, 351
 development methodology 284, 509
 disruptive technology 440, 448

Index

domain-specific modeling (DSM) 107, 109, 111, 126, 128, 131, 132, 335, 337, 361, 375, 400–433, 474, 477, 496, 515, 520, 542
domain-specific modeling languages (DSML)
103–126, 337, 341–376, 404, 477–496
domain model 102–106, 111–117, 120–130, 228–237, 540
dynamic analysis 290, 293, 326–333, 506
Dyninst 303, 304, 305
Dyninst API 303, 305

E

ease-of-use 179, 190, 193, 199, 206, 391
editing executable library (EEL) 302, 331, 528
embedded constraint language (ECL)
122, 123, 124, 125, 126
event QoS aspect language (EQAL) 112–126

F

formal semantics 27, 347
Foundation for Intelligent Physical Agents (FIPA)
264, 265, 285, 517
front-end gate (FEG) 63, 64, 65, 69
functional integration 474–498
functional requirement 97

G

GCC profiling 301, 302
GNU gprof 301
GNU Manifesto 439, 441
GNU public license (GPL) 136, 441, 442, 443
gprof 301
grid computing 402–408, 432, 433, 437, 505, 540

H

hazard analysis 26, 44, 51, 52, 54, 55, 519, 530

I

in-circuit emulators (ICE) 324, 325
insilico 456, 458, 463
instrumentation, compiler-based
290, 302, 324, 328
instrumentation, source code 333
intellectual property (IP) 86, 307, 342, 444, 445, 447, 448
Intel Xeon processor 295, 323
interfaces 6, 31, 32, 46, 60–84, 108, 118, 146, 148, 150, 155, 185, 187, 227, 234–278, 312–392, 403–495, 517–540

J

Java virtual machine tracing interface (JVMTI)
318, 319, 320, 322, 323, 334

K

Komorium 316, 317, 320

L

latency 92, 113, 116, 226, 295, 312–335
livelock 293, 295, 326, 328
local guidance 381–394

M

machine readability 425, 457, 465, 467
Matlab/Simulink 37, 39, 40, 53
metamodel 17, 102–145, 155–158, 349–396, 427–490, 500–534
metaprogramming frameworks 298
Microsoft Windows Performance Counters 311
mobile tasks 83, 89
mobile worker 81, 91, 101
model-driven development (MDD) 128, 158, 372–397, 514, 543
model-driven engineering (MDE) 102–127, 335–371, 426–499
model-driven software development (MDSD)
134–140, 143, 145, 148, 154, 155
model checking 290, 293, 326, 328
model compiler 426, 429, 430, 431
model composition 157, 478–496, 534
model driven architecture (MDA) 128, 131, 136, 154, 156, 157, 158, 399, 400, 503–531
model driven engineering (MDE) 102–127, 335–371, 426–499
modeling guidance 372, 377, 378, 381, 382
modeling wizards 425, 426, 427, 428, 430, 431
monitoring of component-based systems (MCBS)
312, 313, 314
multi-agent system (MAS) 263, 264, 267, 269
multi-processor, symmetric (SMP)
291, 292, 323, 325, 328

N

Nexus debugging interface (Nexus 5001) 331, 523

O

object-oriented design 256, 258, 262, 270
on-chip performance counters 323

open source revolution 439, 440, 441, 442, 447, 448, 450
 open source software 15, 17, 126, 251, 439–454, 509, 527, 534, 536, 540
 OVATION 314, 315, 330, 332, 519, 533

P

Paradyn 303, 304, 332, 531
 parallelism, effective 295, 326
 performance application programming interface (PAPI) 323, 331, 529
 Petri nets, colored 66
 Pin API 305, 306, 321
 Pin program analysis system 305, 306, 307, 308, 321, 331, 529
 priority inversion 295
 process-centred support environment (PSE) 6, 8, 10, 11, 12, 15
 process fragment 8, 10
 process manager (PM) 69
 process relation 7
 product line architecture (PLA) 102–127
 profiling 290–297, 300–370, 508–543
 profiling, active 293, 309
 profiling, common language runtime (CLR) 278, 315, 318, 319, 320, 322, 323
 profiling, passive 293
 profiling, virtual machine (VM) 315, 323
 proprietary source 444, 446, 448
 provenance 455–473, 510–546

Q

quality metrics 389, 455, 456, 469, 470

R

resource adapter (gateway) 480, 494
 reverse engineering 141, 228–292, 474, 496, 499

S

safety-critical systems 26, 219, 220, 295
 scenario view 26
 service-oriented (SO) paradigm 59, 62, 74
 service-oriented architecture (SOA) 30, 59, 60–77, 335–369, 408–498, 506
 software-intensive system 25, 52, 102, 124, 161,–198, 406, 450
 software architecture 5, 55, 105, 12–165, 211–288, 505–541

software architecture decomposition strategies 256, 258
 software architecture views 256
 software design principles 439
 software development process 2, 3, 4, 8, 11, 19, 107, 134, 135, 136, 156, 362
 software engineering 2, 4, 16, 17, 18, 19, 20–87, 100–183, 201–288, 409–453, 504–544
 software engineering methodologies 439
 software lifecycle 3, 103, 133, 270, 294, 345
 software process model 1–20, 98, 100, 213–219, 508–543
 standard test accesses port and boundary-scan architecture (JTAG) 324, 325
 state machines 26, 31, 32, 33, 34, 39, 44, 119
 static analysis 290, 292, 293, 295, 326, 328
 system architecture 25, 26, 94, 165, 209, 190, 280, 286, 338, 449, 481, 529
 system design 3, 98, 161, 173, 194, 201, 207, 209, 338, 396, 496
 system execution modeling (SEM) tool 327–345, 351–365
 systems engineering 23, 26, 156, 166, 177, 197, 209, 210–223, 511–545

T

threadmon 309, 310
 threads (lightweight processes) 259, 290–298, 304–331, 403, 410, 509, 528
 tool integration 126
 traceability 107–158, 159, 248, 280, 505–543
 trace analysis 141, 142, 152, 156, 516
 trace model 133–150
 trace normal form (TNF) 310, 311, 332, 532
 trampoline functions 302, 303, 304, 306, 334

U

unified modeling language (UML)
 4, 20, 23, 27, 29, 30–87, 10–159, 223–288, 34–399, 400–499, 502, 507, 510–542, 544, 545

V

variation point 231, 240
 virtual reality (VR) 223–250
 virtual scene 223–239

W

Web services 54, 73, 74, 77, 101, 251, 278, 410–498, 500–546

Index

workflow 3, 9, 17, 19, 25, 26, 30, 54, 68,
92, 101, 280, 344, 362, 363, 366, 403–
471, 505, 509, 519, 543
workload modeling language 346, 347, 348