

# Index

---

## A

- Acceptance
  - comparison with validation and verification, 342–346
  - defined, 475
  - as key systems engineering concept, 64–65
  - in qualifications chain, 344, 345
  - as term in testing and qualification, 341–342
- Acceptance plans, 153, 168, 192, 305, 475
- Acceptance testing, 362–366
- Acyclic digraphs, 128
- Adjacency, in graphs, 127
- Agrawala, A.K., 242, 306, 307, 329
- Air bag restraint system, case study in stakeholders' requirements, 196–198
- Akao, Y., 404
- Alexander, C., 78
- Alford, M., 27, 385
- Allen, R.H., 260
- Allocated architecture
  - analyzing functional activation and control structure, 305–308
  - approaches for solving allocation problem, 292–297
  - conducting performance and risk analysis, 308–310
  - defined, 285, 475
  - defining allocation problem, 291–292
  - defining functional activation and control structure, 305–308
  - deriving requirements, 299–305
  - developing, 284–316
  - documenting, 310
  - finishing allocation problem, 297–299
  - functional allocation principles, 294
  - and human vs. machine responsibilities, 292–293
  - major development activities, 285–286
  - mapping functions to components, 289–299
  - obtaining approval, 310
  - overview, 27, 29, 51, 52, 284–289
  - relationship to IDEF0 model, 93
  - tracing non-input/output requirements, 299–305
- Ambler, S.W., 23, 25
- Analogies, 264
- Analytic models, 76, 77
- Anderson, T., 242
- Antisymmetric relations, 124, 131
- Apollo 13, case study in stakeholders' requirements, 198–201
- Application entities (AEs), 328
- Application layer, OSI reference model, 327, 329, 330

- Apportionment, 300–301, 344, 475
- Architectures. *See also* Allocated architecture;  
Functional architecture; Physical architecture
- behavioral model, 27
  - data model, 27
  - documenting, 310
  - interface architecture, 51, 52
  - message passing, 322–323
  - network, 323–325
  - overview, 27–30
  - process model, 27
  - shared memory, 323
  - system allocated architecture, 51, 52
- Arciszewski, T., 266
- Ariane 5 case study, 368–370
- Arrow's impossibility theorem, 433
- Asymmetry, as property of unary relations, 131–132
- Attainable (requirements attribute), 171, 172, 475
- Attribute listing, 264–265
- Ayton, P., 418
- B**
- Baier, C., 359
- Barrett, B.E., 410
- Barron, F.H., 409, 410
- Bayes rule, 416–418
- Baylin, E.N., 222
- Behavior diagrams, 385–388. *See also*  
Function flow block diagrams (FFBDs)
- Behavior models
- behavior diagrams, 385–388
  - control flow diagrams, 394–395
  - defined, 375, 475
  - finite-state machines and state-transition diagrams, 388–390
  - overview, 383, 385
  - Petri nets, 395, 397–399
  - statecharts, 390–394, 396
- Beizer, B., 354, 355, 356
- Bell Telephone Laboratories, 6–7, 16, 17
- Berube, M.S., 4, 10, 82
- Bias, R.G., 188
- Big-bang integration process, 352, 353
- Binary relations, 113, 114, 483
- Bipartite graphs, 127, 139–140, 475
- Birnbaum, J., 283
- Black box testing, 362, 363, 370, 475
- Blanchard, B.S., 20, 309
- Block definition diagrams, 98, 99, 100–101, 102
- Block diagrams
- aircraft control system example, 267, 268
  - defined, 74
  - internal, 98–100
  - in physical architecture development, 267, 268
  - SysML semantics, 98–100
  - SysML syntax, 98–100
- Blum, B.I., 151
- Boar, B.H., 22
- Bock, C., 27
- Boehm, B.W., 20, 32, 33, 44
- Bohm, C., 96
- Bottom-up integration process, 346, 347, 351, 352
- Boulton, Matthew, 227
- Braasch, M.S., 313
- Brainstorming, 264, 265
- Brainstorming game, 266
- Brainwriting pool, 266
- Brandt, J.C., 47
- Bresnick, T.A., 167, 185
- Brooks, C.G., 174
- Brown, C.M.L., 189
- Browning, T., 295
- Buede, D.M., 106, 152, 167, 185, 224, 401, 404, 405, 409, 412
- Bus architecture, 324, 325
- C**
- Cardinality, in graphs, 126
- Carnahan, J.V., 167, 187
- Cartesian product of two sets,  $A \times B$ , 113, 476
- Category class, CORE, 67
- CCITT (International Telephone and Telegraph Consultative Committee), 326
- Centralized architecture, 271, 476
- CFDs (control flow diagrams), 394–395
- Chambers, G.J., 153
- Chapanis, A., 189
- Chapman, W.L., 241
- Charbonneau, S.M., 391, 392, 393, 394, 395, 396
- Checkland, P., 19
- Choisser, R.W., 167
- Chu, W.W., 296
- Chusho, T., 362
- Circuits, in digraphs, 127
- CIs (configuration items), 5, 7, 11, 12, 14
- Clarke, L.A., 362
- Classes, defined, 24
- Clousing, D., 275

- Client-server architecture
    - defined, 274–275, 476
    - as model for CORBA, 332
    - and morphological box, 262–263
  - Closed-loop control processes, 227, 228
  - Cockburn, A., 175, 176
  - Common object request broker architecture (CORBA) standards, 332–336
  - Comparable (requirements attribute), 168, 171, 172, 476
  - Complete (requirements attribute), 168, 171, 172, 196, 476
  - Complete functionality, 215
  - Component class, CORE, 67
  - Components
    - defined, 476
    - deriving qualification requirements, 304–305
    - deriving trade-off requirements, 302–304
    - as key systems engineering concept, 61, 64
    - mapping functions to, 289–299
  - Conceptual validity
    - defined, 80, 342–343, 476
    - in qualifications chain, 344, 345
    - role in validation, 345
  - Concise (requirements attribute), 172, 476
  - Concurrent engineering, 7
  - Configuration items (CIs), 5, 7, 11, 12, 14, 476
  - Connectedness, in graphs, 128–129
  - Connell, J.L., 187
  - Consistent (requirements attribute), 168, 171, 172, 194, 476
  - Context of a system. *See* System context
  - Control flow diagrams (CFDs), 394–395
  - Control systems
    - closed-loop processes, 227, 228
    - earliest in history, 227–228
    - feedback processes, 118
    - open-loop processes, 227, 228
  - Controlling cycle, 37, 38
  - Cook, S.C., 42
  - CORBA (common object request broker architecture) standards, 332–336
  - Core cycle, 37
  - CORE systems engineering tool
    - classes, 66, 67
    - documents, 69–70
    - overview, 66–70
    - relations, 66–69
  - Corner, J.L., 402, 410
  - Correct (requirements attribute), 171, 172, 476
  - Cost requirements, 191, 300, 301, 476
  - Coulouris, G, 271
  - Cox, C.B., 17, 174
  - Cox, M.E., 366
  - Craik, K.J.W., 78
  - Crowe, D., 152
  - Cycle model, 37–38
  - Cycles, in digraphs, 128, 138–139
- D**
- Daly, E., 32
  - Dam, S., 26
  - Daniels, J., 167
  - Data flow diagrams (DFDs), 379–383
  - Data link layer, OSI reference model, 327, 329, 331
  - Data models
    - defined, 375, 476
    - entity-relationship diagrams, 377–378
    - higraphs, 378–379
    - overview, 376–377
  - Data validity, 80
  - Davis, A.M., 151, 153, 180
  - DCE (distributed computing environment), 336
  - De Finetti, B., 417
  - De Foe, J.C., 154, 156, 183, 194
  - De Marco, T., 96
  - De Weck, O.L., 41–42
  - Deadlock, 306–308, 476
  - Decentralized architecture, 271, 476
  - Decision analysis. *See also* Multiattribute value analysis
    - axioms, 403–404
    - choice rule, 403
    - decision-making modes, 401
    - elements of decision problems, 402
    - equivalence rule, 403
    - MPWS sample application, 433–442
    - order rule, 403
    - substitution rule, 403
    - uncertainty, 415–432
  - Decision trees, 420–421, 422
  - Decisions, defined, 402, 477
  - Decomposition process
    - in functional architecture development, 239–242
    - in physical architecture development, 257–259
    - in Vee model, 10–11
  - Defense systems Management College (DSMC), 425
  - Defined Term class, CORE, 67
  - Definitive models
    - defined, 75, 477
    - semantics, 75–76
    - syntax, 75–76

- Demand function, 433
- Denning, P.J., 388
- Descriptive models, 75, 76, 477
- Design
- American Heritage Dictionary* definition, 4
  - as decomposition process, 5
  - defined, 477
  - detailed functions, 38–41
  - as engineering consideration, 4
  - in life-cycle diagram, 5–6
  - products of process, 7
  - racecar example, 12–13
  - in Vee model, 10–11, 50, 51–54
- Design independent (requirements attribute), 171, 172, 477
- Design structure matrix (DSM), 295
- Design validity
- defined, 344, 477
  - in qualifications chain, 344, 345
  - role in validation, 345–346
- Development decisions, 14, 15
- DFDs (data flow diagrams), 379–383
- Dickinson, B.W., 231
- Dietrich, B.L., 224
- Digraphs
- acyclic, 128
  - circuits in, 127
  - comparing with IDEF0 diagrams, 139–141
  - connectedness in, 128–129
  - cycles in, 128, 138–139
  - defined, 124, 125–126, 477
  - paths in, 127
  - semicircuits in, 128
  - semicycles in, 128, 139
  - semiwalks in, 128
  - trails in, 127
  - and unary relations, 130–133
  - walks in, 127
- Directed forests, 137–138
- Directed graphs. *See* Digraphs
- Directed trees, 137
- Discipline engineers, 14
- Discrete mathematics
- functions, 116–118
  - overview, 104–106
  - relations, 113–116
  - sets, 106–113
- Distributed architecture, 271, 477
- Distributed computing environment (DCE), 336
- Distributed systems, 271, 274–275
- Document class, CORE, 67
- Documents. *See also* Stakeholders' Requirements Document (StkhldrsRD)
- IDEF0 Model of the Engineering of a System, 455–474
  - outlines, 451–454
  - System Description Document (SDD), 69–70, 454
  - System Requirements Document (SysRD), 31, 32, 155, 453, 485
  - System Requirements Validation Document (SRVD), 31, 32, 453
- DoD Architecture Framework (DoDAF), 23, 25–26
- Domain Set class, CORE, 67
- Dorny, C.N., 231
- Drebbel, Cornelis, 227
- Driscoll, P.J., 6
- Duato, J., 308, 325
- Duffy, M.A., 224
- Dyer, J.S., 404
- E**
- Early validation, 342, 477
- Edge labeling, in graphs, 127
- Edwards, W., 409
- Edwards, W., 402, 409
- EFFBDs (extended function flow block diagrams), 74, 93–97
- Elam, J., 402
- Engineering, defined, 10, 477
- Engineering of systems
- approaches for implementing, 17–23
  - at Bell Labs, 6–7
  - comparison of definitions, 8–10
  - decomposition process, 5
  - defined, 10, 477
  - document outlines, 451–454
  - examples of decisions, 15
  - Hall's definition, 7
  - history, 6–7
  - modeling approaches, 23–27
  - overview, 5–17
  - recomposition process, 5
  - spiral implementation model, 20–23
  - team members with different specialties needed, 14
  - TTDSE implementation model, 19–20
  - value added, 42–45
  - waterfall implementation model, 20, 21
- Engstrom, E.W., 6
- Enhanced FFBs (EFFBDs), 74, 93–97
- Entity-relationship (ER) diagrams, 377–378, 477

- Eppinger, S.D., 295
- Equivalence, 301, 477
- Error detection and recovery functionalities, 239, 242–245
- Errors, 242, 354, 477
- Exit class, CORE, 67
- Expected utility, 426
- Extended FFBDs (EFFBDs), 74, 93–97
- External interface requirements, 165, 218, 246, 247, 477
- External systems diagrams
- defined, 477
  - as key systems engineering concept, 55–56
  - in Mobile Protected Weapons System application, 436–437
  - for stakeholders' requirements, 179–181
- F**
- FAA Wide-Area Augmentation System (WAAS) case study, 311–316
- Fabrycky, W.J., 20, 191, 309
- Fagan, M., 32
- Fagen, M.D., 6
- Failures, 242, 354, 478
- Farry, K., 173
- Faulk, S., 151–152
- Fault tolerance
- achieving by using redundancy, 276–281
  - as functional architectural design goal, 242–243
- Faults
- categorizing, 355
  - defined, 242, 354, 478
  - role in qualification process, 354–356
  - taxonomy of consequences, 355
- FBI fingerprint identification system case study, 268–271, 272, 273
- Federal Information Processing Standard (FIPS) Publications, 85, 93, 376
- Feedback and control
- closed-loop control systems, 227, 228
  - defined, 227, 478
  - earliest control systems, 227
  - in functional architecture design, 227–231
  - IDEF0 illustration, 229, 230
  - negative feedback process, 228–229
  - open-loop control systems, 227, 228
  - positive feedback process, 228–229
- Ferrarini, L., 307
- FFBDs (function flow block diagrams), 74, 93–97
- Figure of merit (FOM), 182, 478
- Fingerprint identification system case study, 268–271, 272, 273
- Finite-state machines (FSMs), 388–389
- Finkelstein, A.C.W., 194
- FIPS. *See* Federal Information Processing Standard (FIPS) Publications
- Fitts, Paul, 7, 292
- Fitts list, 292, 293
- Flowdown. *See* Requirements flowdown
- Folkeson, A., 359
- Forests, in graph theory, 137–138
- Foroughi, A., 401
- Forsberg, K., 5, 9, 10, 38, 40
- Frankel, E.G., 309
- Franklin, G.F., 231
- Frantz, W.F., 171
- French, S., 404, 405
- Fricke, E., 275, 276
- Friedenthal, S., 27
- Friend, J., 263, 402
- FSMs (finite-state machines), 388–389
- Function class, CORE, 67
- Function flow block diagrams (FFBDs), 74, 93–97. *See also* EFFBDs (extended function flow block diagrams)
- Functional architecture
- assigning system's functions, 220–239
  - common mistakes, 241–242
  - creating functional decomposition, 239–241
  - decomposition vs. composition, 218–220
  - defined, 215–216, 478
  - defining by using IDEF0 model, 93, 215, 216, 217, 224, 229, 239
  - developing, 211–249
  - distinctions between system modes, states, and functions, 211–212, 213, 214
  - documenting, 310
  - evaluating hierarchies for shortfalls and overlaps, 231–239
  - feedback and control in design, 227–231
  - hierarchies for development and manufacturing phases, 224–227
  - IDEF0 process model overview, 216–218
  - levels of detail, 215–216
  - matching to physical architecture, 257
  - overview, 27, 28, 51, 52, 211–216
  - partitioning functions into subfunctions, 220–224
  - tracing requirements to elements, 245–248
- Functional requirements, 53, 54, 57, 164, 165, 190, 478
- Functionality
- complete, 215

- defined, 215, 478
  - simple, 215
  - Functions (engineering)
    - defined, 212, 478
    - as key systems engineering concept, 60–61, 62, 63
    - mapping to components, 289–299
    - one-to-one and onto, 257, 259
  - Functions (mathematical)
    - bijective, 117
    - composition, 117–118
    - defined, 478
    - injective, 116–117
  - Fundamental objectives, 182, 183, 186, 478
  - Fundamental objectives hierarchy, 182–183, 184, 478. *See also* Objectives hierarchy
- G**
- General Motors, 16, 17
  - Generic physical architecture, 255, 256, 257, 259
  - Gentner, G., 78
  - Ghahramani, S., 415
  - Glegg, G.L., 354
  - Gobinath, P., 296
  - Gomaa, H., 389
  - Gondran, M., 309
  - Goode, H.H., 7
  - Gotel, O.C., 194
  - Grady, J.O., 151, 153, 171, 300, 342
  - Graphical models, 77, 376. *See also*
    - Behavior models; Data models; Process models
  - Graphs, in mathematics. *See also* Digraphs
    - adjacency in, 129
    - bipartite, 127
    - comparing with IDEF0 diagrams, 139–141
    - connectedness in, 128–129
    - defined, 122, 124, 478
    - isomorphic, 135
    - Konigsberg bridge problem, 124–125
    - ordering relations, 133–134
    - overview, 122–127
    - reachability in, 129
    - as trees, 135–138
    - and unary relations, 130–133
  - Griffith, P.B., 376, 377
  - Guindon, R., 11, 253
  - Gupta, R., 296
- H**
- Haefele, J.W., 266
  - Hall, A., 6, 7
  - Hardware redundancy
    - active, 278–280
    - defined, 276, 479
    - hybrid, 280
    - passive, 277–278
  - Harel, D., 378, 390, 394
  - Harker, P.T., 404
  - Harwell, R., 153
  - Haskins, B., 33, 44
  - Hasse diagrams, 133, 134
  - Hastings, D., 167
  - Hatley, D.J., 20, 220, 221, 222, 224, 240, 380, 394
  - Hazelrigg, G.A., 433
  - Hickling, A., 263, 402
  - Higraphs, 378–379
  - Ho, J., 402
  - Hoang, N.T., 40
  - Hogarth, R.M., 418
  - Holmberg, K., 359
  - Honour, E.C., 43
  - Hooks, I., 171, 173
  - Hoppe, M., 356
  - Howard, R.A., 401, 402, 403
  - Hubble Space Telescope, 46–47
  - Human-designed systems, 50, 156–157, 485
  - Hunger, J.W., 175, 176
- I**
- ICAM (Integrated Computer-Aided Manufacturing), 24, 85
  - ICOMs (inputs, controls, outputs, and mechanisms), 87, 90, 92, 139, 140, 479
  - IDEF family of modeling languages, 85–86
  - IDEF0 (Integrated Definition for Function Modeling)
    - background, 24, 85–86
    - call arrow concept, 92–93
    - defined, 479
    - exit rules concept, 92
    - functional activation rules concept, 92
    - in graph theory terms, 139–141
    - introduction, 56–57
    - loops concept, 91–92
    - Model of the Engineering of a System, 455–474
    - semantics or elements, 86–87
    - shortcomings as modeling process, 85
    - in static behavioral process modeling, 85–93
    - syntax, diagram, 87–89
    - syntax, model, 89–91
    - system engineering use, 93
    - tunneling concept, 92

- IDL (interface definition language), 332, 333, 334
- In degrees, in graphs, 126
- Incidence, in graphs, 126
- INCOSE (International Council on Systems Engineering), 1996, 9
- Influence diagrams, 420–425
- Information redundancy, 280, 479
- Input requirements, 165, 173, 189, 190, 246, 247, 248, 479
- Input/output requirements
  - as category of stakeholder requirement, 165, 166, 189–190
  - defined, 57, 479
  - defining, 69, 165, 189–190
  - deriving for internal items, 299–300
  - tracing, 218, 245–248
- Input/output traces, 152, 177–178, 180, 479
- Instantiated physical architecture, 255, 257, 258, 259
- Integrated Automated Fingerprint Identification System (IAFIS), 268–271, 272, 273
- Integrated Computer-Aided Manufacturing (ICAM), 24, 85
- Integration
  - alternatives to bottom-up process, 351–353
  - as big-bang process, 352, 353
  - as bottom-up process, 346, 347, 351, 352
  - defined, 341, 480
  - as engineering consideration, 4
  - incremental, 353
  - major functions, 346–348
  - overview, 346–351
  - phase, 353
  - racecar example, 12–14
  - as recomposition process, 5
  - at subsystem level, 348–351
  - as top-down process, 351, 352, 353
  - in Vee model, 11–12, 50, 54–55
- Interconnections, defined, 328
- Interface class, CORE, 67
- Interface control information, 328
- Interface data units (IDUs), 328
- Interface definition language (IDL), 332, 333, 334
- Interfaces
  - architectures, 322–325
  - common object request broker architecture, 332–336
  - defined, 480
  - design process, 336–338
  - as key systems engineering concept, 61
  - OSI reference model, 327–331
  - overview, 319–322
  - standards, 325–326
- International Council on Systems Engineering (INCOSE), 1996, 9
- International Organization for Standards (ISO), 325, 326, 332
- International Telephone and Telegraph Consultative Committee (CCITT), 326
- Intransivity, as property of unary relations, 132–133
- Irreflexivity, as property of unary relations, 131
- ISO (International Organization for Standards), 325, 326, 332
- Isomorphisms, 135
- Issue class, CORE, 67
- Item class, CORE, 67
- Items, defined, 61, 480
- J**
- Jackson, S., 296
- Jacky, J., 368
- Jacobson, I., 175, 176
- Jacopini, G., 96
- Jagacinski, R.J., 78
- Jalote, P., 242
- Jelassi, M., 401
- Johnson, B.W., 276, 277, 278, 279, 280
- Johnson-Laird, P., 78
- Judgmental models, 76, 77
- K**
- Karangelen, N.E., 40
- Katoen, J.-P., 359
- Kee, C., 313
- Keeney, R.L., 182, 183, 267, 402, 404, 405, 412
- Keller, L., 402
- Kirkwood, C.W., 402, 406, 410
- Kleindorfer, P.R., 418
- Klir, G.J., 19
- Kluger, J., 198, 201
- Konigsberg bridge problem, 124–125
- Kwinn, M.J., 404
- L**
- Lake, J., 5
- Lano, R.J., 23, 173
- Larsen, R.F., 106
- Law of total probability, 416
- Layers, OSI reference model, 327, 329–331
- Lee, D., 242, 359

- Lepoeter, K., 38  
 Levardy, V., 356  
 Levi, S., 242, 306, 307, 329  
 Levis, A.H., 26, 27, 157, 296  
 Life cycles  
   cost commitment and incursion, 8  
   defined, 480  
   organizing in StkhldrsRD, 169–171  
   representation of phases, 5–6, 7  
   systems, defining, 33–38  
   as systems engineering consideration, 3–4  
 Lindley, D., 417  
 Link class, CORE, 67  
 Lions, J.L., 370  
 Livelock, 308, 480  
 Loops, 91–92, 124  
 Lovell, J., 198, 201
- M**
- Machol, R.E., 7  
 MacKinnon, D., 326, 327, 329  
 Magee, C.L., 41–42  
 Magnuson, E., 283  
 Manna, Z., 355  
 Manos, K.L., 153  
 Manufacturing, defined, 480  
 Mar, B.W., 171  
 Marca, D.A., 239  
 Maroni, M., 307  
 Marshall, C., 189  
 Martin, J., 194  
 Martin, J.N., 18  
 Mathematical symbols, 106–107  
 Matrix analysis, 266  
 Maxwell, D.T., 404  
 Maxwell, J.C., 227  
 Mayhew, D.J., 188, 189  
 Mayr, O., 227  
 McGowan, C.L., 239  
 McMenamin, S.M., 222  
 Mead, M., 402  
 Mead, Thomas, 227  
 Measure of effectiveness (MOE), 182, 480  
 Measure of performance (MOP), 182, 480  
 Meisenzahl, J., 356  
 MENS (Mission Element Needs Statement), 451  
 Mental models  
   defined, 77–78, 480  
   vs. quantitative models, 78  
 Merkhofer, M.W., 418  
 Mesh architecture, 324, 325  
 Message passing architectures, 322–323  
 Meta-systems, 17–19  
 Miliken, D.L., 12  
 Miliken, W.F., 12  
 Miller, J.G., 222, 225  
 Miller, R.A., 78  
 MIL-STD 499A, 9  
 MIL-STD 499B, 9, 153  
 MIL-STD-881B, 254  
 Ministry of Defence (MoD) Architecture Framework (AF), 25  
 Mission Element Needs Statement (MENS), 451  
 Mission requirements, 30, 55, 153–154, 173, 175, 179, 182, 480  
 Mobile Protected Weapons System (MPWS), 433–442  
 MoD (Ministry of Defence) AF (Architecture Framework), 25  
 Mode of a system. *See* System mode  
 Models. *See also* Process models; Qualitative models; Quantitative models  
   comparing types, 78, 79, 80  
   defined, 75, 480  
   definitive, 75–76  
   descriptive, 75, 76  
   examples, 75  
   mental, 77–78  
   normative, 75, 76  
   overview, 75–80  
   physical, 76, 77  
   potential uses in systems engineering, 40, 80  
   purpose in developing, 79  
   semantics, 75–76  
   syntax, 75–76  
   taxonomy, 76–78  
   validating, 80  
 Modifiable (requirements attribute), 168, 171, 172, 480  
 Monte Carlo simulations, 77  
 Mooz, H., 5, 9, 10, 38, 40  
 Morgenstern, O., 403  
 Morphological box  
   automobile navigation support system  
     example, 261–262  
   background, 260  
   caution, 262–263  
   defined, 257, 481  
   in fingerprint identification case study, 271, 273  
   hammer example, 260, 261  
   overview, 253, 257, 259, 260–264  
 Mott, J.L., 106  
 Mowbray, T.J., 332, 336



- Multiattribute value analysis
  - defined, 404, 481
  - eliciting value functions, 405–407
  - eliciting value weights, 407–415
- Mobile Protected Weapons System
  - application, 433–442
  - overview, 404–405
- Multigraphs, 124, 125
- Murata, T., 397, 398
- Murray, C., 17, 174
  
- N**
- Nachtmann, H., 309
- NAF (NATO Architecture Framework), 25
- Nagel, S.S., 404
- NASA (National Aeronautics and Space Administration), 14, 46–47, 174, 200
- NATO Architecture Framework (NAF), 25
- Negative feedback control process, 228–229
- Network architectures, 323–325
- Network layer, OSI reference model, 327, 329, 330
- Newell, A., 253
- Nielsen, J., 188, 365
- Nii, J., 253
- Node labeling, in graphs, 127
- Normative models, 75, 76, 481
- N-squared ( $N^2$ ) diagrams, 23, 24, 93, 383, 384
  
- O**
- Object Management Group, Inc. (OMG), 23, 24, 80, 331–332
- Object request broker (ORB) interface, 332–334
- Objectives hierarchy
  - adding value curves, 185, 186
  - defined, 182, 481
  - as key systems engineering concept, 57
  - in Mobile Protected Weapons System application, 436, 438
  - for performance requirements, 182–186
- Object-oriented (OO) design, 23, 24
- Objects, defined, 24
- Observance requirements, 168, 192, 305, 481
- OMG (Object Management Group, Inc.), 331–332
- OO (object-oriented) design, 23, 24
- Open architecture, 274, 481
- Open Software Foundation (OSF), 336
- Open systems, defined, 328
- Open Systems Interconnection (OSI) reference model, 326–331
- Open-loop control processes, 227, 228
  
- Operational concept
  - defined, 481
  - as key systems engineering concept, 55, 57
  - in Mobile Protected Weapons System application, 435–436
  - for stakeholders' requirements, 173–179
- Operational requirements and tests, 12
- Operational validity
  - defined, 80, 342, 481
  - in qualifications chain, 344, 345
- ORB (object request broker) interface, 332–334
- ORBOS (ORB-object service) architecture, 334, 335
- Order of G, in graphs, 126
- Orfali, R., 334
- OSF (Open Software Foundation), 336
- OSI (Open Systems Interconnection) reference model, 326–331
- Ottaway, D.B., 198
- Out degrees, in graphs, 126
- Output requirements, 165, 189, 190, 246, 247, 248, 481
- Overlap in the functional architecture, 232–239, 481
  
- P**
- Pages, A., 309
- Palmer, J.F., 222
- Papaccio, P.N., 20
- Parametric diagrams
  - defined, 74
  - for performance modeling, 101, 102
  - semantics, 101, 102
- Parnell, G.S., 404
- Partially ordered sets, 133, 134
- Partition on a set  $A$ , 111, 481
- Pathfinder case study, 339–340
- Paths, in digraphs, 127
- Pennington, N., 78
- Perdu, D.M., 296
- Performance analysis, defined, 309, 482
- Performance modeling, in SysML, 100–102
- Performance requirements, 163, 168, 179–185, 267, 482
- Perry, W.E., 351
- Petersen, C.C., 47
- Petri nets (PNs), 24, 395, 397–399
- Petroski, H., 344
- Physical architecture
  - centralized vs. decentralized, 271

comparison with military's Work  
     Breakdown Structure, 254–255  
 creativity techniques, 259–267  
 defined, 252, 482  
 design flexibility, 275–276  
 developing, 252–283  
 development issues, 267–281  
 development overview, 257–259  
 distributed, 271, 274  
 documenting, 310  
 generic *vs.* instantiated, 255, 256, 257,  
     259  
 graphic representations, 267  
 matching to functional architecture,  
     257  
 morphological box technique, 260–264  
 option creation techniques, 264–267  
 overview, 27, 28, 29, 51, 52, 252–253  
 Physical layer, OSI reference model, 327,  
     329, 331  
 Physical models  
     defined, 76, 77, 482  
     *vs.* qualitative models, 79  
 Pipeline architecture, 324, 325  
 Pirbhai, I.A., 20, 220, 221, 222, 224, 240,  
     380, 394  
 PNs (Petri nets), 24  
 Pohl, E., 191, 309  
 Positive feedback control process, 228–229  
 Power set of set  $A$ , 112–113, 482  
 Prang, J., 8  
 Prasad, B., 7  
 Presentation entities (PEs), 328  
 Presentation layer, OSI reference model, 327,  
     329, 330  
 Price, H.E., 294  
 Probability theory, 403–404, 415–417. *See also*  
     Uncertainty  
 Problem Situation of Mission Element Need  
     Statement, and Systems Engineering  
     Management Plan (SEMP), 31, 32  
 Problem-solving techniques, 264–267  
 Process models. *See also* IDEF0 (Integrated  
     Definition for Function Modeling)  
     data flow diagrams, 379–383  
     defined, 74, 375, 482  
     N-squared charts, 383, 384  
     overview, 379–383  
 Protocol control information, 328  
 Protocol data units (PDUs), 328  
 Protocols, 328  
 Prototypes, 187, 482  
 Pugh, S., 404, 408

## Q

Qualification  
     acceptance testing, 362–366  
     compared with testing, 341–342  
     defined, 341, 482  
     methods categories, 359–362  
     overview, 354–356  
     planning levels, 356–359  
 Qualification methods  
     analysis and simulation, 359  
     defined, 482  
     demonstration, 361  
     inspection, 359  
     testing, 359, 361–362, 363  
 Qualification plans, 13–14, 353, 356–358  
 Qualification requirements  
     categories, 305  
     as category of stakeholder requirement, 168,  
         169, 192–194  
     defined, 482  
     deriving for components, 304–305  
     in qualification planning process, 358  
     tracing, 304–305  
 Qualitative models  
     behavior models, 375, 383–399  
     data models, 375, 376–379  
     defined, 77, 482–483  
     graphical techniques, 375–399  
     process models, 375, 379–383  
     *vs.* quantitative models, 79, 80  
 Quantitative models  
     defined, 76–77, 483  
     *vs.* mental models, 78  
     *vs.* qualitative models, 79, 80

## R

Racecar example, 12  
 Raiffa, H., 404, 405, 412  
 RAND Corporation, 7  
 Rank-order centroid (ROC), 409–410  
 Rasmussen, J., 78  
 Reason, J., 189  
 Recomposition process, 5, 11–12  
 Reed, M.A., 254  
 Reference model for OSI, 326–327  
 Reflexivity, as property of unary relations,  
     131  
 Regression testing, 351, 483  
 Reitman, W.R., 253  
 Relations  
     binary, 113, 114  
     Cartesian products, 113  
     equivalence, 116

- Relations (*Continued*)
    - ordered pairs, 113
    - ordering, 133–134
    - partial ordering, 116
    - unary, 113, 114–116, 131–133
  - Relevance diagrams, 418–420
  - Requirement class, CORE, 67
  - Requirements. *See also* Stakeholders' requirements
    - common documents, 30–32
    - compared with specifications, 30
    - defined, 29–30, 153
    - as key systems engineering concept, 57–60
    - mission, 30
    - modeling in SysML diagrams, 100
    - overview, 30–33, 153
    - performance, 30
    - stakeholder, 30
  - Requirements flowdown, 300–301, 302, 313, 344, 346, 483
  - Requirements statements, 171, 483
  - Requirements validity
    - defined, 343–344, 483
    - in qualifications chain, 344, 345
    - role in validation, 345
  - Resource class, CORE, 67
  - Richardson, D.J., 362
  - Ring architecture, 324, 325
  - Risk, defined, 425, 483
  - Risk analysis, 309, 425–426, 483
  - Risk avoidance, 425, 483
  - Risk class, CORE, 67
  - Risk issues, uncertainty in, 44–45
  - Risk management, 425, 483
  - Risk preference
    - assessing, 427–429
    - exponential, 429–432
    - overview, 425, 426–427
  - Risk tolerance, 431
  - Risk transference, 425, 483
  - Rittel, H., 253
  - Roberts, R.A., 415
  - Rosen, K.H., 106
  - Ross, A.M., 167
  - Ross, D., 86
  - Royce, W.W., 20
  - Ruh, W.A., 332, 336
- S**
- Saaty, T.L., 404, 412
  - SADT (Structured Analysis and Design Technique), 24, 86
  - Sage, A.P., 9, 20, 309
  - Sailor, J.D., 9, 153, 185, 187
  - Samson, D., 362
  - Savage, L.J., 417
  - Scenarios, 55–56, 57, 74, 160, 175–179, 483
  - Schedule requirements, 168, 173, 191, 484
  - Scheiber, S.F., 8
  - Schkade, D.A., 216
  - Schlager, K.J., 6
  - Schmekel, H., 306
  - Schulz, A.P., 275, 276
  - Schwartz, M., 326, 329
  - SDD. *See* System Description Document (SDD)
  - Semantics
    - block definition diagrams, 100–101, 102
    - block diagrams, 98–100
    - defined, 74, 82, 484
    - EFFBDs, 93–94
    - IDEF0, 86–87
    - in models, 75–76
    - parametric diagrams, 101, 102
    - sequence diagrams, 83–84
    - use case diagrams, 83
  - Semicircuits, in digraphs, 128
  - Semicycles, in digraphs, 128, 139
  - Semiwalks, in digraphs, 128
  - SEMP (Systems Engineering Management Plan), 31, 32, 452
  - Sen, A.K., 433
  - Sequence diagrams
    - elevator example, 84, 85
    - as input/output traces, 177–179
    - semantics, 83–84
    - syntax, 84
  - Service access points (SAPs), 328
  - Service providers, 328
  - Session layer, OSI reference model, 327, 329, 330
  - Sets
    - defined, 106, 484
    - describing members, 107–108
    - equality, 109
    - examples, 107
    - finite, 108
    - inclusion, 108–109
    - infinite, 108
    - mathematical symbols, 106–107
    - operations on, 109–111
    - partitions, 111–112
    - power, 112–113
    - singleton, 108
    - special, 108–109
    - subsets, 108–109
    - writing memberships, 107

- Shachter, R.D., 418, 421, 423
- Shafer, L., 187
- Shared memory architectures, 323
- Shea, Joe, 14, 16–17
- Shneiderman, B., 189
- Shortfall in the functional architecture  
   categories, 231–232  
   defined, 231, 484  
   identifying, 232–239
- Shuey, R.L., 271
- Signal flow graphs, 231
- Simon, H.A., 253
- Simple graphs, defined, 124
- Simulation models, 76–77
- Sinnott, R.W., 47
- Size of G, 126
- Snider, W.D., 410
- SofTech, Inc., 86
- Software engineering  
   spiral model, 20–23  
   waterfall model, 20, 21, 28–29
- Software redundancy, 280, 484
- Spanning trees, 136–137
- Specifications  
   compared with requirements, 30  
   defined, 30, 484
- Spetzler, C.S., 418
- Spiral model, software engineering, 20–23
- Spitzer, Lynn, 46
- Spoke architecture, 324, 325
- SRD (System Requirements Document), 31, 32, 155, 453, 485
- SRVD (System Requirements Validation Document), 31, 32, 453
- Stael von Holstein, C.A., 418
- Stakeholders, 14, 484. *See also* Stakeholders' requirements; Stakeholders' Requirements Document (StkhldrsRD)
- Stakeholders' requirements  
   air bag restraint system case study, 196–198  
   analyses, 187–188  
   Apollo 13 case study, 198–201  
   categorizing, 161–164  
   characteristics, 171, 172  
   defined, 30, 52, 151, 484  
   defining design problem, 157–161  
   defining requirements, 189–194  
   developing, 157–169  
   external systems diagram, 179–181  
   input/output requirements, 165, 166, 189–190  
   managing requirements, 194–195  
   objectives hierarchy for performance requirements, 168, 169, 182–187  
   operational concept, 173–179  
   overview, 154–157  
   prototyping, 187  
   qualification requirements, 168, 192–194  
   StkhldrsRD, 169–171  
   structuring using partition, 164–169  
   system-wide and technology requirements, 167, 190–192  
   trade-off requirements, 167–168, 192  
   usability testing, 188–189  
   writing, 171–173
- Stakeholders' Requirements Document (StkhldrsRD)  
   defined, 484  
   format overview, 169–171  
   outline, 452–453  
   overview, 31, 32
- Standards, for interface design, 325–326
- Starvation, 308, 484
- Stassen, H.G., 78
- State of a system, 212, 214, 484
- Statecharts, 390–394, 396
- State/Mode class, CORE, 67
- State-transition diagrams (STDs), 388–390
- STDs (state-transition diagrams), 388–390
- Stevens, A.L., 78
- Stevens, R., 194
- Stillwell, W.G., 409
- Strategic check cycle, 37, 38
- Structured Analysis and Design Technique (SADT), 24, 86
- Structured programming, comparison to FFBD constructs, 96
- Subnetworks, defined, 328
- Suh, N.P., 53
- Suitability requirements, 191, 484
- Surge or race, 308, 484
- Symbolic models, 77
- Symbols, mathematical, 106–107
- Symmetry, as property of unary relations, 131
- Syntax  
   block diagrams, 98–100  
   defined, 74, 82, 484  
   EFFBDs (extended function flow block diagrams), 93–94  
   IDEF0, 87–91  
   in models, 75–76  
   sequence diagrams, 84  
   use case diagrams, 83

- Synthesis, 301, 344, 484–485
- SysML (Systems Modeling Language)  
 compared with TTDSE (Traditional, Top-Down Systems Engineering), 65–66  
 defined, 23, 74  
 diagrams for requirements modeling, 100  
 meta-system and requirements modeling, 82–85  
 modeling overview, 80–82  
 overview, 26–27, 65–66
- SysRD (System Requirements Document), 31, 32, 155, 453, 485
- System context, 50, 157, 485
- System Description Document (SDD), 69–70, 454
- System mode, 211–212, 213, 480
- System requirements, 52, 104–105, 154, 160, 485
- System Requirements Document (SysRD), 31, 32, 155, 453, 485
- System Requirements Validation Document (SRVD), 31, 32, 453
- System tasks or functions, 50, 156, 485
- Systems. *See also* Engineering of systems  
 behavior modeling view, 40  
 data modeling view, 40  
 defined, 3, 50, 156, 485  
 design process functions, 50–54  
 design process overview, 49–55  
 distinctions between modes, states, and functions, 211–212, 213, 214  
 environmental modeling view, 40  
 implementation modeling view, 40  
 life-cycle considerations, 3–4  
 modeling views, 40  
 process modeling view, 40  
 role of design concept, 4  
 role of integration concept, 4  
 stakeholders, 3–4  
 ways to categorize, 41–42
- Systems analysis, role of RAND Corporation, 7
- Systems engineering. *See* Engineering of systems
- Systems Engineering Management Plan (SEMP), 31, 32, 452
- System's external systems, 50, 157, 485
- Systems Modeling Language. *See* SysML (Systems Modeling Language)
- System-wide requirements. *See* Technology and system-wide requirements
- T**
- Taguchi, G., 275
- Tan, L.M.-T., 296
- Taylor, J.R., 362
- TCP/IP (Transport Control Protocol/Internet Protocol), 331
- Technology and system-wide requirements  
 as category of stakeholder requirement, 167, 190–192  
 defined, 485  
 deriving subsystem-wide requirements, 300–301  
 tracing, 300–301
- Terminko, J., 266
- Testing. *See also* Acceptance testing  
 compared with qualification, 341–342  
 as qualification method, 359, 361–362, 363
- Textual models, 77
- Therac-25 case study, 368
- Thurston, D.L., 167, 187
- Time redundancy, 281, 485
- Top-down integration process, 351, 352, 353
- Traceable (requirements attribute), 172, 194, 196, 486
- Traced (requirements attribute), 172, 486
- Trade studies, 309–310, 485
- Trade-off requirements  
 as category of stakeholder requirement, 167–168, 192  
 defined, 486  
 deriving for components, 302–304  
 as indirect weight elicitation technique, 412–413  
 in Mobile Protected Weapons System application, 439  
 tracing, 302–304
- Trails, in digraphs, 127
- Transitivity, as property of unary relations, 131–132
- Transport Control Protocol/Internet Protocol (TCP/IP), 331
- Transport layer, OSI reference model, 327, 329, 330
- Trees  
 defined, 486  
 directed, 137  
 in graph theory, 135–136  
 spanning, 136–137
- TTDSE (Traditional Top-Down Systems Engineering)  
 compared with SysML, 65–66  
 defined, 19  
 key terms and basic process, 49–65

- layered iterative process, 19–20  
 modeling approaches, 23–24  
 overview, 19–20  
 as “peeling the onion” process, 368, 369
- U**
- Ulvila, J.W., 410
- UML (Unified Modeling Language), 23, 24–25. *See also* SysML (Systems Modeling Language)
- Unambiguous (requirements attribute), 172, 345, 486
- Unary relations, 113, 114–116, 131–133, 483
- Uncertainty in decisions  
 decision trees, 420–421, 422  
 expected utility, 426  
 influence diagrams, 420–425  
 probability theory, 415–417  
 relevance diagrams, 418–420  
 risk preference, 425–432
- Understandable (requirements attribute), 172, 345, 486
- Unified Modeling Language (UML), 23, 24–25. *See also* SysML (Systems Modeling Language)
- Unique (requirements attribute), 168, 172, 486
- U.S. Federal Aviation Administration, WAAS case study, 311–316
- Usability  
 categorizing users, 189  
 defined, 486  
 performance elements, 188  
 testing, 188
- Usability testing  
 defined, 188, 486  
 as part of verification testing, 365–366
- Use case diagrams, 74, 82–84
- Utility curves, 439, 440–442
- V**
- Validation. *See also* Conceptual validity; Design validity; Requirements validity  
 compared with verification, 13  
 comparison with acceptance and verification, 342–346  
 defined, 13, 341, 342, 486  
 as key systems engineering concept, 64  
 relationship to verification, 345  
 as term in testing and qualification, 341–342
- Validation plans, 168, 192, 305, 486
- Value functions, eliciting, 405–407
- Value weights. *See* Weights
- Value-focused thinking, 267
- Van de Vegte, J., 231
- Van den Hamer, P., 38
- VanGundy, A.B., 264, 266
- Vargas, L.G., 404
- Vee model, 10–12, 38, 50, 54
- Veldhuyzen, W., 78
- Verifiable (requirements attribute), 172, 194, 196, 486
- Verification  
 compared with validation, 13  
 comparison with acceptance and validation, 342–346  
 defined, 13, 341, 486  
 as key systems engineering concept, 64  
 overview, 344–345  
 in qualifications chain, 344, 345  
 relationship to validation, 345–346  
 as term in testing and qualification, 341–342
- Verification cycle, 37
- Verification plans, 168, 192, 305, 486
- Verification Requirement class, CORE, 67
- Verplanck, W.L., 292
- Voges, U., 362
- Von Braun, Werner, 174
- Von Neumann, J., 403
- Von Winterfeldt, D., 402
- W**
- Wagenhals, L.W., 26
- Waldinger, R., 355
- Walters, J.M., 187
- Walton, M., 167
- Warfield, J.N., 19
- Waterfall model, software engineering, 20, 21, 28–29
- Watson, S.R., 185, 401, 405, 409, 412
- Watt, James, 227
- Weights  
 analytical hierarchy process approach, 412, 413  
 balance beam approach, 413–414  
 direct elicitation techniques, 409–410  
 illustration, 411  
 indirect elicitation techniques, 412–415  
 in Mobile Protected Weapons System application, 439–442  
 and utility curves, 440–442
- Wenzel, S., 36
- West, P.D., 253, 402
- White box testing, 362, 363, 486

Wide-Area Augmentation System (WAAS)  
  case study, 311–316  
Wieringa, R.J., 157  
Wiklund, M.E., 188  
Wilner, D., 339  
Wingard, L., 306  
Work Breakdown Structure (WBS),  
  254–255  
Wright, G., 418  
Wymore, A.W., 9, 20, 164, 168, 196, 213, 407

**Y**

Yager, R.R., 404  
Yannakakis, M., 359  
Yoon, K., 404  
Yourdon, E., 306, 377, 380, 383

**Z**

Zahavi, R., 332  
Zwicky, F., 260