abnormal scenarios 69, 465–6, Fig. 13.15
abstract domains 74–6, Fig. 2.4
accuracy goals 270, 568, 572
accuracy requirements 22, 25–6
accuracy statements 22
Achieve goals 266–7, Fig. 7.4
converse assertions 315–16, 621
identification 474–5, Fig. 13.20
obstruction patterns 617–18, Fig. 18.12
actigrams 133, Fig. 4.6
actions 456–9, Fig. 13.8
active observation 79
actors 30, 74, 136
agent behaviour 265, 431, 596–7
agent beliefs 403
agent concurrency 396, 424, 431–2
agent dependencies 274, 403–5, 432,
Fig. 11.5
see also dependency diagrams
agent diagrams 405–6, 411–13, Fig. 11.6
agent interfaces 398, 406–7, 414, Fig. 11.7,
Fig. 11.13
agent knowledge 403, 618–19
agent meta-model 490–1, Fig. 14.5
agent models
assignments 401
building 411–15, 515–17
interface with goal model 303, Fig. II.1
purpose 395
structural consistency 494–6
use in RD 564, 565
see also agent diagrams; context diagrams;
dependency diagrams
agent orientation 278
agent substitution 350–1, Fig. 9.11
agent wishes 314, 402
agents
capabilities 395, 396, 397–9, Fig. 11.1
categories 396–7
commitments 430–1
decomposition 408–11, 453, Fig. 11.10,
Fig. 13.5
definition 260, 396–7
features 397
identification 411–12
modelling 298, 384
non-determinism 396, 431, 596
as operation performers 401–2, 427,
Fig. 11.4
refinement 408–11, 453, Fig. 11.10,
Fig. 13.5
responsibilities 399, Fig. 11.2
aggregation
agents 408
objects 376–7, 386, Fig. 10.9
agile development 53–5
algebraic specification 167–72, 173
Alloy analyser 208
alternative goal refinements 304–5, Fig. 8.8,
Fig. 8.9
formal patterns 605, Fig. 18.3
soft goal identification 313–14, Fig. 8.16
alternative options
evaluation 14, 32, 105–7, 274, 517–18
modelling 303–7
product line projects 524
qualitative assessment 106–7, 557–60
quantitative assessment 107, 560–2
soft goals and 268
sources 87, 105, 557
alternative responsibility assignments 305, 515–17
evaluation 16–17, 306
modelling 305–6, 401, Fig. 8.10
analogy reuse 541–4, Fig. 16.2
Analytic Hierarchy Process (AHP) 109–12
AND-refinement
goals 297–300, Fig. 8.2, Fig. 8.3
obstacles 341, Fig. 9.3
obstruction propagation 342–3, Fig. 9.5
vs OR-refinement 318–19, Fig. 8.18
AND/OR graphs 307–8
animations 198
anti-goals 552
formal generation 618–22
formal specification 591
identification 554, 620–1
refinement 621
architectural design 42–3, 566–7
architectural refinement patterns 572–4,
Fig. 16.15, Fig. 16.16
architectural requirements 27, 570
arity
associations 368
relationships 131
artefact-driven elicitation techniques 62, 64–76
background study 64
card sorts 66
conceptual laddering 67
data collection 65
knowledge reuse 72–6
prototypes 70–2
questionnaires 65–6
repertory grids 66
storyboards/scenarios 67–70
associations 362, 363, 366–71
arity 368
of associations 379, Fig. 10.11
constraint patterns 369–70
controlled 398
derived 378
domain properties 370–1
goals constraining 370–1
identification 381
instances 366–7, Fig. 10.3
modelling 384–5, 386
monitored 397
multiplicities 368–71, Fig. 10.4, Fig. 10.5
OR-associations 378–9, Fig. 10.10
ordered 379, Fig. 10.10
reflexive 368
stability of 593
vs relations 373
assumptions 19
analysis 15–16
definition 20, 264
atomic features 239
attackers 553, 554
attributes 131, 371–2
controlled 398
derived 378
elementary 371
modelling 384, 385
monitored 397
multiplicity 372, Fig. 10.6
names 388
as pointers 387, Fig. 10.15
of relationships 131–2
rigid 372, Fig. 10.6
structured 371
auxiliary episodes 466
availability requirements 25

B
background study 64
behaviour meta-model 492–3, Fig. 14.7
behaviour models 449–50
building 463–77, 521–4
formal synthesis 627–35
interface with goal model 303, Fig. II.1
structural consistency 495–6
use in RD 565
see also scenarios; state machine models
behavioural goals 265–8, Fig. 7.3, Fig. 7.4
coverage 303
realizability 399–400
satisfaction of 265
scenarios and 276, 474–5, Fig. 7.8
binary associations 368
boundary conditions 88–9
  avoiding 92–3
    regression-based derivation 623–5
bounded model checkers 208
bounded SAT solvers 608–9, 610,
  Fig. 18.7
brainstorming 81
branching temporal logics 154
brownfield projects 40

C
capability instance declarations 398, 406
card sorts 66
case studies 6–12
case-driven refinement patterns 322–5
causal links 592–3
change
  causes 224–5, 247, Table 6.1
  deferred 249
  documentation 223
  dynamic 249–51
  evaluation 248–9
  incorporation 249
  prioritization 247, 248
types 221, Table 6.1
  see also requirements management
change anticipation 223–5
change control 246–9, Fig. 6.16
change management see requirements management
change requests 247
checklist-based reviewing 189
  defect-based checklists 191–2, Table 5.1
  domain-specific checklists 193
  language-based checklists 193–5
  quality-specific checklists 192–3
circularity checking 203
class behaviours
  in meta-model 492–3
  modelling 449, 454–63
class diagrams 144
  derivation 380–3, Fig. 10.12
  meta-model representation 487
commonality 524
  levels of 223, Fig. 6.2
commutativity axiom pattern 171
completeness 35, 335
completeness checking 203, 204–5
compliance requirements 27
component requirements 96
components 4–5, 566
composite features 239
composite state 459
composition 376, 386, Fig. 10.9
Computational Tree Logic (CTL) 154, 208
concept-driven acquisition 66–7
conceptual laddering 67
conceptual objects
  agent monitoring/control 397–8
  definition 360–1
  features 364–6, Fig. 10.2
  identification 380–1
  modelling 359, 384, 385
  names 388
  types 362–3, Fig. 10.1
  see also requirements management
change anticipation 223–5
change control 246–9, Fig. 6.16
change management see requirements management
change requests 247
checklist-based reviewing 189
  defect-based checklists 191–2, Table 5.1
  domain-specific checklists 193
  language-based checklists 193–5
  quality-specific checklists 192–3
circularity checking 203
conflicts
  causes 89
  documentation 91
  in goal models 271, 273–4, 301–2, 314, Fig. 8.7
  identification 32, 90–1, 512
  as source of change 225
  types 88–9
confusion obstacles 403
connectors 566
consistency checkers 244
consistency checks 203–4
consistency rules 88–9, 493–6
constraint analysis 15–16
content analysis 64
context diagrams 127–8, 130, Fig. 4.2
context diagrams (continued)
  agent interface view 406–7, 517, Fig. 11.7, Fig. 11.8, Fig. 15.13
  generation 413–15, Fig. 11.14, Fig. 11.15
contextual enquiry 81
contribution links 262
control links 397, 405
controlled variables 21, 469
countermeasures
  identification 99–101, 350–3, 556
  reuse 100
  run-time monitoring 99
  selection 101, 353
coverage links 303
cross referencing 237
customer-driven projects 40
cut-set trees 97, Fig. 3.4
data collection 65
data dependencies 34
data-flow architecture 568–70, Fig. 16.11
  quality refinement 571–4
  style refinement 570–1
dataflow diagrams (DFD) 134–6, Fig. 4.8
datagrams 133, Fig. 4.7
decision tables 122–3, 194, Table 5.2
decomposition-by-case pattern 322–3, 505, Fig. 8.21
Defect Detection Prevention (DDP) 102–5, Fig. 3.5
defects
  evaluation 189–90
  types 36–7, Table 1.1
deferred requirements 221
definitions 20, 593
dependency chains 404–5, 407
dependency diagrams 407–8
dependency links 228, 404, Fig. 6.5
derivation links 230–1, Fig. 6.9
  bi-directional nature 383, 441–2
  satisfaction arguments and 231–2, 541, Fig. 6.10
  traceability 540
derived associations 378
derived attributes 378
descriptive statements 17, 264
desired behaviours 28
development requirements 27–8
diagrammatic notation 127–45
  activities and data 133–4
  conceptual structures 130–3
  information flows 134–6
  inspection checklists 194–5
  interaction 136–8
  stimuli and responses 142
  system behaviour 138–42
  system operations 136
  system scope 127–30
  view integration 142–4
diagrammatic visualisation 200
discriminators 376
dissatisfaction obstacles 338
distribution constraints 27
divergence 88–9, 301
  analysis 545–6
  formal patterns 627, Fig. 18.13
  formal resolution 625–7
  identification 548–9
  see also conflicts
divergence patterns 625, Fig. 18.13
divide-and-conquer pattern 324–5, Fig. 8.24
documentation 33, 43, 46, 119
  see also requirements document (RD)
domain knowledge 14
domain post-condition 426–7, Fig. 12.3
  formal specification 594
  required conditions 439–40
  domain pre-condition 426–7, Fig. 12.3
  formal specification 594
  required conditions 439–40
domain properties
  definition 19, 264
  formal specification 592
  from obstacle analysis 338, Fig. 9.2
  modelling 298, 365–6, 370
  satisfaction arguments and 300, Fig. 8.6
  domain understanding 30–1, 46, 61–2
  see also requirements elicitation
domain-specific visualisation 200
dynamic change 249–51

eager behaviour 431, 596–7
elementary attributes 371
entities 131, 362, 366
modelling 363, 384
entity-relationship diagrams 130–3, Fig. 4.5
entry/exit actions 457–9, Fig. 13.10
environment 5
environmental agents 410, 451
environmental operations 424
episodes 450, 453, Fig. 13.4
errors 36–7, Table 1.2
cost 33, 48, 187
impact 49–51, 187
scale of problem 48
ethnographic studies 79–80
event notifications 457, Fig. 13.9
event tables 165, Table 4.2
event trace diagrams 136–8, Fig. 4.10
event-based specification 163–7, 173
animation of 198
events 363
internal/external 455
modelling 384
stimuli 466
evolutionary prototypes 71, 72
expectations
definition 263, 264, 293
formal specification 590
in goal models 298
external events 455

fault trees 96–7, Fig. 3.3, Fig. 3.4
FAUST animator 198, 200, Fig. 5.3
feature diagrams 239–40, Fig. 6.15
features 23, 220
final sub-states 460, 462, Fig. 13.14
first-order predicate logic 148–50
first-order refinement patterns 607–8
first-order specification languages 150–1
fit criterion 123–4, 296
flaws 37–8, Table 1.3
fluents 437–9, 632–3, Fig. 12.9
focus groups 80
formal specification 46, 127, 145–74
algebraic 167–72
event-based 163–7, 173
first-order languages 150–1
goal models 588–91
history-based 151–5, 173
inspection checklists 195
object models 592–3
operation models 594–6
state-based 155–63, 173
strengths and limitations 173–5, 583, 603
traceability management and 242–3, 246
formal verification 188, 202–11
four-variable model 21, Fig. 1.4
frame diagrams 129, Fig. 4.4
frame problem 160, 166, 597
free documentation 120–1
functional goals 269, Fig. 7.5
functional prototype 70
functional requirements 23, 28
gauge variables 561–2, Fig. 16.8
generalization
modelling 387
multiple 375
see also specialization
generative semantics 166, 597
global templates 125–7, 194
glossary of terms 31, 89, 388
from object model 359, 563
goal diagrams 293–4
as AND/OR graphs 307–8
class diagrams and 380–3, Fig. 10.12
goal meta-model 488–9, Fig. 14.3
goal models 293–4
agent diagrams from 411–13
annotations 294–6, Fig. 8.1
building 309–27, 503–6, 507–10
conflicts 301–2, Fig. 8.7
context diagrams from 413–15
interface with other views 293, 302–3, Fig. II.1
obstacle diagrams and 340, Fig. 9.4
product line projects 525, Fig. 15.18
refinement graphs 297–300
risk analysis on 273, 335
scope 316–17
structural consistency 494–5
use in RD 564–5, 566
see also goal diagrams
goal refinement 297–301
features 308, Fig. 8.12
formal methods 604–9
goal refinement (continued)

refinement patterns 319–27
satisfaction arguments and 272–3, 300, Fig. 7.6
vs operation refinement 441
see also alternative goal refinements

goal restoration 352
goal satisficing 268, 308
goal specification patterns 589–90, Fig. 17.1
goal substitution 350, Fig. 9.11
goal weakening 351–2, 514, 626–7, Fig. 15.10
goal-agent co-refinement 408–9, Fig. 11.11
goal-agent co-refinement pattern 410, Fig. 11.12

goal-oriented RE
agent orientation compared 278
definition 260
object orientation compared 278
top-down analysis and 279

goals
ambiguities 319
categories 265, 269–71, Fig. 7.5
conflicts 271, 301–2
definition 260, 293
features 293, 294–6, Fig. 8.1
formal specification 588–91
granularity 262
identification 270, 271, 275–6, 309–16
importance 272–5
model checking 277–8
obstructions see obstacles
operations compared 317–18, Fig. 8.17
realizability 399–401, Fig. 11.3
satisfaction 231–2, Fig. 6.10
scenarios and 276, 473–5, Fig. 7.8
specification 295
strengths and limitations 463, 464, Table 13.1
types 265–9, Fig. 7.2
unrealizability 400–1
use cases and 277
greenfield projects 40
group sessions 80–1, 98
grouping rule 497
guard-introduction pattern 323–4, Fig. 8.23
formal use 604–5, Fig. 18.1

guarded transitions 139–40, 456, 471, Fig. 13.7
guidewords 96–7

H

hazard obstacles 338
higher-order functions 172
history-based specification 151–5, 173
animation of 198
hypotheses
definition 264
formal specification 592
modelling 298
satisfaction arguments and 300

I

i* diagrams 274, 404, 407–8, Fig. 11.9
IEEE Std-830 template 52, 125–6, 564–6, Fig. 4.1
in-house projects 41
inaccuracy obstacles 338
inconsistencies 87, 88–9
see also conflicts
inconsistency management 88–93, 188
independence axiom pattern 171
information goals 270, 338
inheritance
inhibiting 132, 375
multiple 375
transitions 459–60, 462
see also specialization
inheritance conflict 375
initial sub-states 460, 462, Fig. 13.14
initializations 365, 366, 367
formal specification 592
input variables 21
inspection reports 189, 190
inspections and reviews 187, 188–96, 211
checklists 191–5
guidelines 190
process 188–90, Fig. 5.1
installation constraints 27
instance behaviours
in meta-model 492
modelling 449, 450–4
integrity requirements 25
intentional specifications 277
intentional threats 552, 553–6
inter-view checks 538–9
inter-view consistency rules 142–4, 493–6
traceability and 242–3
interaction events 450, 451, 468–9
interaction matrix 91
interface requirements 26
interleaving semantics 141, 596
internal events 455
interoperability requirements 26
interviews 77–9
intra-view rules 539–40
invariants 155

J

JAD (Joint Application Development) 80

K

knowledge acquisition 61, 62–3
knowledge reuse 72–6
analogue reuse 541–4, Fig. 16.2
domain-independent 72–4
domain-specific 74–6
process 72, 543–4
and risk identification 98

L

labelled transition system 628–35
lazy behaviour 431, 596–7
lifelines 451
linear temporal logic (LTL) 151–4, 584–91
absolute time bounds 587–8
model-checking 208
relative time bounds 587
state assertions 584–5
temporal assertions 585–6
variable-dependent time bounds 588
liveness property 207, 278
load analysis 540, Fig. 16.1
logic 146–51
first-order predicate 148–50
propositional 146–7

M

machine solution 4, Fig. 1.1
macro-events 454
Maintain goals 267–8, Fig. 7.4
identification 315, 475
obstruction patterns 617–18, Fig. 18.11
malevolent agents 397, 512
see also threats
market studies 82
market-driven projects 40
message sequence charts (MSCs) 137, 629, 631
meta-models 485–93
definition 485
domain level 486, Fig. 14.1
instance level 486, Fig. 14.1
meta level 485–6, Fig. 14.1
reuse 73–4, Fig. 2.3
role 486–7
structure 487–8, Fig. 14.2
milestone-based dependencies 405
milestone-driven refinement pattern 321–2, 405, 505, 513, Fig. 8.19
formal use 604, Fig. 18.1
misinformation obstacles 338
mission-critical goals
completeness 298, 337
obstacle analysis 335
mission-critical systems
change procedures 247
importance of RE 49–51
model checking 205
mock-ups 71
mode transition tables 164, Table 4.1
model checking 205–8, Fig. 5.4
goals and 277–8
model checklists 540
model configurations 524, 525–8
model databases
analogue reuse 541–4
browsing 538
query-based analysis 493, 538–44
structural consistency 538–40
traceability management 540–1
view generation 540
model-driven elicitation 82
models
requirements for 288–9
role in RE 287–8
see also individual models
monitored condition 467–8
monitored variables 21, 469
monitoring links 397, 405
multiplicity
attributes 372, Fig. 10.6
relationships 132

N
N-ary associations 368, Fig. 10.4
narratives 67
natural language
in QA 212
structured 121–7
unrestricted 120–1
negative scenarios 68–9, 450
obstacle coverage and 465
as sequence diagrams 452, Fig. 13.3
negotiation 89–90
nested states 459
NFR framework 106–7, Table 3.5
non-functional goals 269, Fig. 7.5
non-functional requirements 24
categories 24–8, Fig. 1.5
functional overlap 28
non-functional statements see accuracy
statements
non-rigid variables 454, 468
normal design projects 40
normal scenarios 69, 465–6, Fig. 13.15

O
object instances 360–2
object meta-model 489–90, Fig. 14.4
object models 359, 363–4
annotations 364–6, Fig. 10.2
building 380–8, 506, 510–11
content 387–8, Fig. 10.16
formal specification 592–3
interface with goal model 303,
Fig. II.1
product line projects 525, Fig. 15.19
structural consistency 494–5
use in RD 563, 565
see also class diagrams
object orientation 278
objective identification 13–15
objects see conceptual objects
obligations 396, 402, 428, 430, 440
observation 79–80
obstacle analysis 314, 344–53, 512–15,
Fig. 9.7, Fig. 15.9
obstacle diagrams 339–40, Fig. 9.1
annotations 343, Fig. 9.6
obstacle mitigation 352, 513
obstacle models 339–43
building 512
interface with goal model 303, Fig. II.1
see also obstacle diagrams
obstacle monitoring 349–50
obstacle prevention 351, Fig. 9.11
obstacle reduction 352
obstacle refinement
AND-refinement 340, 341, Fig. 9.3
OR-refinement 342–3, Fig. 9.5
refinement patterns 349
obstacles
categories 338
completeness 337–8
confusion 403
definition 336–7
evaluation 349
features 343, Fig. 9.6
formal generation 613–18
formal specification 591
in goal meta-model 488–9
identification 344–9, 549–51
resolution 349–53, 513
obstruction links 303
obstruction patterns 617–18
operation meta-model 491–2, Fig. 14.6
operation models
building 437–42, 518–21
completeness 433
formal specification 594–6
links to other models 303, 441–2, Fig. II.1
purpose 421–2
structural consistency 494–6
use in RD 565
see also operationalization diagrams; use
case diagrams
operation performers 401–2, 427, Fig. 11.4
Index

operational specifications 277
operationalization 277, 424, 427–34
   by multiple operations 430
   correctness 432–3
   formal checking 609–13
   formal specification 594–6
   of multiple goals 429
   satisfaction arguments and 432–4
   traceability and 434
operationalization diagrams 435, 442, Fig. 12.6, Fig. 12.7
operationalization links 294, 303, 424, 435
operationalization patterns 610–13
operations
   application 422–3, Fig. 12.1
   atomicity 423–4
   categories 424
   definition 422
   features 425
   formal specification 595–6
   generators 168–9
   goals compared 317–18, Fig. 8.17
   identification 437–9
   modifiers 159, 168
   obligations 396, 402, 428, 430, 440
   observers 159, 169
   permissions 396, 402, 428, 430, 440
   required conditions 427–9, Fig. 12.4
   scope 425–6, Fig. 12.2
   signature 422, 425, 430, 435, Fig. 12.2
OR-associations 378–9, Fig. 10.10
OR-refinement
   goals 274, Fig. 7.7
   obstacles 342–3, Fig. 9.5
   vs AND-refinement 318–19, Fig. 8.18
ordered associations 379, Fig. 10.10
output variables 21
outsourced projects 41
overspecification 38–9

P
packages 496–7, Fig. 14.8
passive observation 79
performance instance declaration 401, 406
performance links 405, 435
performance requirements 26
permissions 396, 402, 428, 430, 440
positive scenarios 68–9, 450, 463
   and goal coverage 465
   notation 137
post-conditions 155
pre-conditions 155
preferred behaviours 29
Prefix Tree Acceptor (PTA) 629–34, Fig. 18.15
prescriptive statements 17, 264
prioritization see requirements prioritization
privacy requirements 25
probabilistic goals 269
problem diagrams 128–9, 130, Fig. 4.3
problem world 4, Fig. 1.1
process algebras 172
process-based reviewing 189, 190
process-related risks 94, 95
product-line projects 41, 524–8
product-related risks 94, 95, 335
project management 44
project types 40–2
property testers 633–4, Fig. 18.19
propositional logic 146–7
protocol analysis 79
prototyping 42, 70–2, 198, Fig. 2.1
pruning semantics 166, 597

Q
QFD (Quality Function Deployment) 80
qualitative labels 558–60
quality assurance see requirements quality assurance
   software 43
quality requirements 24–6
questionnaires 65–6

R
R-net diagrams 142, Fig. 4.14
RAD (Rapid Application Development) 82
radical design projects 40
reachability graphs 207
real-time temporal logic 152–3, 584–91
refinement graphs/trees 297–300, 326–7,
   Fig. 8.2
   see also AND/OR graphs
refinement links 293–4, 297
   satisfaction arguments and 300, 432
two way nature 299–300
Index

refinement patterns
  common patterns 321–7
  completeness checking and 298, 319
  formal use 604–8, Fig. 18.1
  obstacle analysis 349
  reuse 320
reflexive associations 368
relations 373
relationships 131–2
reliability requirements 25
repertory grids 66
required conditions 427–9, Fig. 12.4
  consistency 429, 430
  domain conditions 439–40
  multiple 429–30
required post-condition 428
  on domain conditions 439–40
required pre-condition 428
  on domain conditions 439–40
required specification 594–5
required trigger condition 428
  on domain conditions 439–40
  formal specification 594
required taxonomies 24, 28–9, 72–3, Fig .1.5
requirements see software requirements; system
  requirements
requirements animation 187–8, 198, 211
  model extraction 199
  model simulation 199–200
  strengths and limitations 200–1
  visualisation 200, Fig. 5.3
requirements completeness 35, 335
requirements consolidation 33
requirements database 144
  conflicts in 91
  queries on 187, 196–7, 211
  see also model databases
requirements document (RD) 33
  changes 219, 220–2, Table 6.1
  defects 36–40, Table 1.1
  disciplined documentation 121–7
  errors 36–7, Table 1.2
  flaws 37–8, Table 1.3
  formal specification 127, 145–74
  free documentation 120–1
  model-driven generation 562–6
  qualities 35–6

quality assurance 187–8
semi-formal specification 127–45
structure 119, 124–7, 273
templates 125–6, 563–6, Fig. 4.1
traceability see traceability management
uses 42–4, 51–2, Fig. 1.7
requirements elicitation 31–2, 46
  artefact-driven techniques 62, 64–76
  combining techniques 81–2
  model-driven 82
  objectives 61
  stakeholder-driven techniques 62, 76–81
requirements engineering
  agile development and 53–5
  definition 3, 6
  dimensions 13–17, Fig. 1.2
  importance of 47–51
  obstacles to good practice 52–3
  other disciplines and 45–7
  process 34–5, 54, Fig. 1.6
  role 51–2
  software engineering and 45
  system design and 44–5
requirements evaluation 32–3, 46, 87–8
  QA and 188
techniques 88–113
requirements evolution 47, 219–22
  documenting 225–46
  goals and 275
  time-space dimensions 220–1
  types and causes 221–2
  see also requirements management
requirements inspection checklists 191–5, 494
requirements lifecycle 30–5
requirements management 220, 251
  change anticipation 223–5
  change control 246–9
  policy 251
  traceability management 225–46
requirements monitoring 249–51
requirements prioritization 32, 87, 108–13
  constraints 108
  value-cost comparison 109–12
requirements quality assurance 187–8
  animation based 187–8, 198–201
  database queries 196–7, 87
  early analysis 212–13
  formal checks 188, 202–11
ideal process 212
inspection and reviews 187, 188–96
requirements specification 33, 43, 46, 119
requirements validation 33, 198
requirements verification 33, 202–12
resolution links 353, Fig. 9.1
responsibility assignments 16–17, 395
alternative see alternative responsibility assignments
features 308, Fig. 8.12
goal realizability 399–401
modelling 294, 412–13
as source of change 225
responsibility instance declaration 399, 406
responsibility links 298, 303, 305, 405
reuse
abstract domains 74–6, Fig. 2.4
frame diagrams 129, Fig. 4.4
meta-models 73–4, Fig. 2.3
refinement patterns 319–27
refinement trees 326–7
requirement taxonomies 72–3
see also knowledge reuse
review board 247–8
review meetings 63, 189, 190
reviews see inspections and reviews
revision links 229, Fig. 6.7
revisions 220–1, Fig. 6.1
rigid attributes 372, Fig. 10.6
risk analysis 88, 93–105
on goal models 273, 335
QA and 188
risk assessment 32, 98–9
risk assessment tables 98–9, Table 3.2
risk checklists 95
risk control 99–101
risk exposure 99
risk levels 98
risk management 95–101, 102–5, Fig. 3.2
risk trees 96–7, 613–18, Fig. 3.3, Fig. 3.4
risk-reduction leverage 101
risks
definition 94, 335
documentation 101–2
identification 95–8
as source of change 225
types 94
RPNI 629–31
runtime monitoring 249–51, 349

S
SADT (Structured Analysis and Design Technique) 133–4, 242, Fig. 4.6, Fig. 4.7
safety goals 327, 338, Fig. 8.28
safety property 207, 278, 634
safety requirements 24
SAT solvers 608–9, 618
satisfaction arguments 22–3
derivation links and 231–2, 541, Fig. 6.10
from goal models 272–3, 300–1, Fig. 7.6, Fig. 8.5
hypotheses and 505
operationalization and 432–4
satisfaction goals 270, 338
refinement trees 326–7, Fig. 8.27
scenarios 67–70
conflict resolution 514, Fig. 15.11
coverage 465–7
goals and 276, 473–5, Fig. 7.8
refinement 452–4
risk identification 98
as source of operations 439, Fig. 12.10
state conditions and 467–9
strengths and limitations 69–70, 463–4,
Table 13.1
types 68–9
see also sequence diagrams
SCR (specification language) 163–6, 203–5
security goals 270, 572
formal identification 620–1
formal specification 618–20
obstructions 338
refinement trees 327, Fig. 8.28
specification patterns 619–20
security requirements 25
semi-formal specification
alternative option evaluation 557–62
conflict analysis 544–9
diagrammatic notations 127–45
model database queries 538–44
obstacle identification 549–51
strengths and limitations 144–5, 174–5
threat analysis 551–6
traceability management and 242–3, 246
sequence diagrams 137, 144
agent decomposition 453, Fig. 13.5
episodes 453, Fig. 13.4
scenario refinement 452–4
sequence diagrams (continued)
  scenario representation 450–4, Fig. 13.1
  state conditions 467–9, Fig. 13.16
  state diagrams from 469–73
shared phenomena 4
simulations 198
single-product project 41
SMV 208
snapshot state 454
soft goals 268–9
  conflicts 302
  formalization 589
  identification 305, 306, 313–14, Fig. 8.16
  satisfaction 268, 308
software agents
  refinement 410
  in sequence diagrams 451
software data architecture 567–8
software design 566–7
software engineering 45
software lifecycle 42–4, Fig. 1.7
software operations 424
software prototypes 70
software quality assurance 43
software requirements
  definition 19, 263, 411
  notation 21
  system requirements and 21–3
software-to-be 5
specialization 132, 373–6, Fig. 10.7
  modelling 386–7
  multiple 376, Fig. 10.8
specialization links 373, 374–5
specification see requirements specification
specification languages 119
  first order 150–1
  SCR 163–6, 203–5
  Z 156–62
  see also formal specification; semi-formal specification
specification patterns 155
specification-based traceability management 242–3
SPIN 208
stability
  of associations 593
  levels of 223, Fig. 6.2
stakeholder-driven elicitation techniques 62, 76–81, 82
  group sessions 80–1
  interviews 77–9
  observation 79–80
  uses 92, 100
  stakeholders 30, 62–3
  state assertions 584–5
state conditions 467–9
state diagrams 144, 455–63, Fig. 13.6
  actions 456–9, Fig. 13.8
  event notifications 457, Fig. 13.9
  from operation models 475–7, Fig. 13.21
  from sequence diagrams 469–73
  guarded transitions 456, Fig. 13.7
state machine diagrams 138–42, Fig. 4.11
state machine models 454
  from operationalized goals 475–7
  from sequence diagrams 469–73
goal-driven synthesis 628
model checking 205–7
  refinement 459–63
scenario-driven synthesis 628–35
  strengths and limitations 464, Table 13.1
see also state diagrams
state variables 265, 361, 397, 454
state-based specification 155–63, 173
  animation of 198, 199–200
  semantics 166
statecharts 141, Fig. 4.13
  see also state diagrams
statement templates 123–4, 193
  see also state diagrams
statements
  goals and 264, Fig. 7.1
  local rules 121–4
  in object models 371
  overlapping 91–2
  scope 17–18, Fig. 1.3
  specification 119
  types 17–23
states 138–9, 454
  parallel decomposition 460–3, Fig. 13.12
  sequential decomposition 459–60, 462–3, Fig. 13.11
static semantics checking 203
stimuli events 466
stimulus-response goals 270
storyboards 67
structural consistency
  checks 538–40
  rules 493–6
structured attributes 371
structured interviews 77, 79
sub-states 459–63
synchrony hypothesis 166
syntax checking 202
system 4
system behaviour 265
representation 138
types 28–9
system boundary 17, 306, 317
system design 44–5
system model
behavioural view 289
building method 501–24, Fig. 15.1
functional view 289
integration mechanisms 485–97
intentional view 289
responsibility view 289
semantic picture 434–5, 596–7, Fig. 12.5
structural view 289
views 289, Fig. II.1
see also individual models
system proposal 31, 32, 61–2
system requirements 18–19, 263, 293, 411
behaviours and 28–9
deferred 221
formal specification 590
functional 23, 28
non-functional 24–8
notation 21
software requirements and 21–3
system scope 274
diagrammatic notation 127–30
system services 15–16
system state variables 361
system-as-is 5, 12–13
modelling 294, 450, 503–6
system-to-be 5, 13
modelling 294, 450, 507–24
systems-to-be-next 6, 219

\[
\text{T}
\]
tacit knowledge 63, 79
tasks 424
technology opportunities 14
temporal assertions 585–6
temporal logics 151
test data

\[
\text{from scenarios} \quad 201, 450, 463
\]
sources 123, 212
specification 362
textual visualisation 200
theorem proving 208–11, 604
threat analysis 314, 512, 551–6
threat graphs 554–6, Fig. 16.6
threat models 552–3
threat trees 96, 553
threats 338, 552
throwaway prototypes 71, 72
top-down analysis 279
traceability databases 240–1
traceability graphs 235, Fig. 6.12, Fig. 6.14
traceability link generators 243–4
traceability links 226–33
direction 226, Fig. 6.3
ER model 232–3, Fig. 6.11
identification 233–5, 275, 434
maintenance 236–7
types 228–31, Fig. 6.4
uses 235–6
traceability lists 238, Table 6.3
traceability management 225–46, Fig. 6.13
cost-benefit trade-off 233–5, 243–6
goals and 275
specification-based 242–3
techniques 237–44
using model databases 540–1
traceability matrices 237–9, Table 6.2
traceability model databases 241–2
traceability policy 233, 244, 246
traces 140, 141
tracking associations 382–3, Fig. 10.13
transition-based specification see event-based specification
transitions 139–40, 455–6
guarded 139–40, 456, 471, Fig. 13.7
inheritance 459–60, 462
trigger conditions 456
type checking 202–3

\[
\text{U}
\]
unacceptable behaviours 29
uncontrollability-driven refinement patterns 326, Fig. 8.26
Unified Modelling Language (UML)
use in RE 144, 289
Unified Modelling Language (UML) (continued)

see also class diagrams; sequence diagrams;
state diagrams; use case diagrams
unintentional threats 552
Unique Controller constraint 401
Unique Performer constraint 427
unmonitorability-driven refinement patterns
325, Fig. 8.25
unrealizability-driven refinement patterns
325–6, 382, 401
formal use 606–7, Fig. 18.4
unstructured interviews 77
unuseability obstacles 338
use case diagrams 136, 144, 435–7, Fig. 4.9,
Fig. 12.8
generation 442, 521
use cases 277
use links 229–30, 541, Fig. 6.8
useability goals 338, 572
useability requirements 26
user interface prototype 70
user story 54

validation scenarios 198
value-cost comparison 109, Fig. 3.6
variant links 228–9, 239–40, Fig. 6.6,
Fig. 6.15
variants 220–1, Fig. 6.1
modelling 228–9, 239–40, Fig. 6.15
variations 525
versions 220–1, 306, Fig. 6.1
VOLER template 126
vulnerability analysis 403, 405

walkthroughs 188
weighted matrices 107, 560–1, Table 3.6,
Fig. 16.3
WHAT dimension 15–16
WHO dimension 16–17
WHY dimension 13–15
wish links 402

XlinkIt 244

Z (specification language) 156–62
combining schemas 161–2
data schemas 156, 157–9
initialization schemas 156, 158–9
inspection checklist 195
language checks 202–3
operation schemas 156, 159–61
traceability and 242