

Index

A

abstract symbols 182, 183–93,
203–4
abstract syntax 29, 55
action classes
 development methodologies
 62–4, 155–6
 notation 193–4
action diagrams
 development methodologies 175
 notation 201, 207
agent-oriented methodology 101
architecture *see* infrastructure
associations 46, 56–7
attributes 46
Australian Standard AS4651 43, 70
auxiliary concepts 195–6

B

backward-looking models 23–4
Behavior Trees Technique 71–2
builds 147, 184–5

C

cardinality 24
CASE Data Interchange Format
 (CDIF) 57
Catalysis 12
categorization 36–41, 43
CDIF *see* CASE Data Interchange
 Format
chunk model 103–4, 152
CIM *see* computationally independent
 model
clabjects *see* class–object pairs
class diagrams 55–7, 79, 97, 150
class–object pairs (clabjects)
 conflicts 90–2, 94

 deep instantiation 113–14
 development methodologies
 135–6, 157–8, 164–70, 177
 infrastructure 40, 47
 powertypes 114, 118, 122–4
code bridges 67
composite work products 190
computationally independent model
 (CIM) 66
concrete syntax *see* notation
conflicts 83–95
conglomerates 195–6
constraints 154, 192–3
cross-boundary associations 85–7
CRUD approach 63

D

deep instantiation 112–14, 128–9
deontic values 195, 197–8
development methodologies 1–20,
133–62
 action classes 155–6
 characterization 3–7
 conflicts 85
 creating from metamodel
 163–80
 definitions 2–3
 endeavours 8–9, 14–16, 163–5,
 167–8, 170–2, 175–8
 extensions to ISO/IEC24744
 157–61
 fragments 163, 167–75
 frameworks 70–2
 high-level view 141–4
 language 11–13, 15–18
 metamodelling 11–19, 62–5,
 133–62

- development methodologies
 - (*continued*)
 - Method domain dualities 138–41
 - method engineering 6–11
 - model kind class 148–9, 156–7
 - philosophy and scope 133–41
 - powertypes 122–4, 163, 165–75
 - process classes 142–3, 144–8, 152–4
 - producer classes 142–3, 151–2
 - product classes 142–3, 148–51, 152–4
 - representations 13–15, 16
 - requirements 8
 - software engineering 7–11
 - stage classes 146–7
 - support classes 154–5
 - tailoring 5–6
 - tasks 9–10
 - Type–Instance relationship 17
 - usefulness 4–7
 - user perspectives 134–8
 - discriminants 41, 115, 117
 - documents 135–6, 141, 150, 164–7, 188
 - Domain-Specific Language (DSL) 13, 28, 55
 - dual representations 32–5
 - duration 183
 - dynamic specialization 121
- E**
- elicit requirements 167–9, 201
 - enactment diagrams 178, 199–200, 206
 - endeavours
 - conflicts 89–90
 - development methodologies 8–9, 14–16, 134–44, 155–6, 163–72, 175–8
 - infrastructure 36–7, 44–6
 - new approaches 112, 113, 122, 128
 - notation 181
 - epistemology 33–4
 - Extended Modelling Infrastructure *see* Xome Modelling Infrastructure
- F**
- formalization 40
 - forward-looking models 23–4, 30, 35
 - fragments
 - development methodologies 163, 167–75
 - guidelines and solutions 105
 - quality aspects 101–3
- G**
- Gantt charts 178, 199–200
 - generalization
 - conflicts 87, 93, 95
 - development methodologies 159
 - modelling languages 56–7
 - powertypes 124–5
 - quality aspects 99–100
 - generic symbols 196–8
- H**
- hardware items 189–90
 - homomorphism 22, 24, 26–30
- I**
- illegal associations 85–7
 - infrastructure 21–51
 - assumptions 75–6
 - backward-looking models 23–4
 - basic modelling elements 46–7
 - cardinality 24
 - categorization 36–41, 43
 - conflicts 83–95
 - domain-specific languages 28
 - endeavours 36–7, 44–6

- exotic modelling elements 47
- formalization 40
- forward-looking models 23–4, 30, 35
- guidelines and solutions 104–6
- homomorphism 22, 24, 26–30
- interpretive mappings 24–6, 35
- linguistic metamodels 79–83, 105
- logical representations 32–5
- metamodelling 41–4, 60–1, 68–72
- model units 26–30
- modelling languages 25–36
- models and modelling 21–6, 42–4, 46–7
- ontological metamodels 79–83, 105
- physical representations 32–5
- powertypes 40–1, 43, 47, 122–4
- process classes 60–1
- quality aspects 95–104
- relationships between models and metamodels 42–4
- representations 21–6, 30, 32–41
- seamless transitions 31–2
- specialization 38–9
- subjects under study 21–7, 30, 32–5, 38, 68–9
- syntax 28–9
- tasks 41, 47–8
- terminology and culture clashes 76–8
- utilization 47–8
- veterinary management system 37–41
- input–output transformation engines 63–4
- instance-of relationships
 - development methodologies 134
 - metamodelling 75–6, 78–82
 - powertypes 111, 117
- instantaneous stages 147, 185
- instantiation
 - conflicts 85–7, 93, 95
 - deep 112–14, 128–9
 - development methodologies 159, 177
 - modelling languages 56–7
 - new approaches 111–14, 124–5
 - notation 197–8
 - potency 112–14, 128–9
 - powertypes 121, 124–5, 128
 - quality aspects 99
- interpretive mappings 24–6, 35
- ISO/IEC 15504 134
- ISO/IEC 19501 55
- ISO/IEC 24744
 - action classes 155–6
 - auxiliary concepts 195–6
 - conflicts 95
 - constraints 192–3
 - development methodologies 133–62, 169–73, 175–8
 - diagram types 198–201, 205–7
 - extensions 157–61
 - generic symbols 196–8
 - high-level view 141–4
 - infrastructure 43
 - metamodelling 60, 62–3, 70
 - Method domain dualities 138–41
 - model kind class 148–9, 156–7
 - notation 181–207
 - philosophy and scope 133–41
 - process classes 142–3, 144–8, 152–4
 - producer classes 142–3, 151–2, 190–2
 - product classes 142–3, 148–51, 152–4
 - relationship concepts 193–5
 - stage classes 146–7, 182, 183–6
 - support classes 154–5
 - user perspectives 134–8
 - work products 187–90

- ISO/IEC 24744 (*continued*)
 - work units 186–7
- isotypical mappings 25
- J**
- Java 66
- L**
- language
 - definitions 35–6
 - development methodologies
 - 11–13, 15–18, 148–50
 - infrastructure 25–36
 - semantics 92–4
 - syntax 28–9, 55
 - transformation 65–8, 70, 72
 - see also* modelling languages; Unified Modeling Language
- library metaphor 95
- lifecycle diagrams 173, 198–9, 205
- linguistic metamodelling
 - conflicts 79–83
 - guidelines and solutions 105
 - quality aspects 96–7
- logical metamodelling *see* ontological metamodelling
- logical representations 32–5
- M**
- MAS *see* multi-agent systems
- materialization 77–8, 118
- MDA approach 42, 66–7
- MDE *see* model-driven engineering
- meta-object facility (MOF)
 - assumptions 75
 - development methodologies 17
 - guidelines and solutions 105
 - infrastructure 35–6, 42–3
 - quality aspects 99–101
- metaclasses
 - conflicts 99–100
 - development methodologies
 - 141–2, 144, 165
 - powertypes 124
 - quality aspects 105
- metalevel hierarchies 43–5
- metametamodelling 42, 44
- metatypical mappings 25
- Method domain
 - development methodologies
 - 135–43, 159, 164–5, 168, 172, 175–7
 - dualities 138–41
 - new approaches 112, 121–4, 128–9
 - notation 181
- method engineering
 - conflicts 85, 95
 - development methodologies
 - 6–11, 136–8, 163–5
 - powertypes 122
- method fragments *see* fragments
- method-specific concepts 177
- methodbases *see* repositories
- methodologies *see* development methodologies
- milestones 185–6
- model classes 189
- model-driven engineering (MDE)
 - 65–8, 70, 72
- model kind class 148–9, 156–7
- model transformation 65–8, 70, 72
- model units 26–30, 53–4, 151, 176
- modelling languages
 - class diagrams 55–7
 - conflicts 92–4
 - definitions 35–6
 - development methodologies
 - 12–13, 15–18
 - infrastructure 25–36
 - integration 62–3, 84
 - metamodelling 53–7
 - syntax 28–9
 - transformation 65–8, 70, 72
 - see also* Unified Modeling Language

MOF *see* meta-object facility
 multi-agent systems (MAS) 101
 multiple methodologies 2

N

natural language 12
 notation
 auxiliary concepts 195–6
 constraints 192–3
 diagram types 198–201, 205–7
 generic symbols 196–8
 ISO/IEC24744 181–207
 modelling languages 29, 55
 producer classes 190–2
 product classes 148–50
 relationship concepts 193–5
 stage classes 182, 183–6
 work products 187–90
 work units 186–7

O

object code 65–6
 Object Constraint Language (OCL)
 55
 object-oriented modelling
 assumptions 75
 conflicts 88
 development methodologies 8, 12
 infrastructure 27, 32, 34
 new approaches 111, 114, 125
 quality aspects 104
 terminology 76–8
 objects 46–7
 OCL *see* Object Constraint
 Language
 OML *see* OPEN Modeling
 Language
 ontological metamodels
 conflicts 79–83
 guidelines and solutions 105
 infrastructure 33–4
 quality aspects 96
 OOSPICE metamodel 71

OPEN Modeling Language (OML)
 54–5, 61
 OPEN Process Framework (OPF)
 59–62, 102
 OPEN/Metis Methodological
 Framework 70–1
 OPF *see* OPEN Process Framework
 Orthogonal Classification
 Architecture 79–80, 104

P

partitioned type 114–15, 118, 121,
 165
 partitioning models 69
 Petri Nets 66
 phases 147, 184
 physical metamodels *see* linguistic
 metamodels
 physical representations 32–5
 platform independent model (PIM)
 66–7
 platform specific model (PSM)
 66–7
 postconditions 154, 192–3
 potency
 conflicts 90–1
 deep instantiation 112–14,
 128–9
 powertypes 126–9
 powertypes
 classification 116–17
 conflicts 90–1, 95
 development methodologies
 133–62, 163, 165–75
 infrastructure 40–1, 43, 47,
 122–4
 metamodelling 82–3
 new approaches 114–29
 patterns 114–20, 143
 potency 126–9
 process modelling 118–21
 quality aspects 102
 representations 124–6

- powertypes (*continued*)
 - stereotypes 125, 127
 - terminology 77–8
 - preconditions 154, 192–3
 - process classes
 - conflicts 83–95
 - development methodologies
 - 142–3, 144–8, 152–4, 174
 - integration with products 84, 111
 - metamodelling 58–63, 72
 - notation 186
 - powertypes 118–21
 - process data diagrams 96–8
 - process diagrams 173–4, 200–1, 206
 - producer classes
 - development methodologies
 - 13–16, 142–3, 151–2, 174
 - metamodelling 58, 72
 - notation 190–2
 - product classes
 - development methodologies
 - 142–3, 148–51, 152–4, 174–5
 - integration with processes 84, 111
 - programming by difference 158–9
 - project-specific elements 163, 177
 - Prolog 67
 - prototypical mappings 25
 - PSM *see* platform specific model
 - purpose-fit methodologies 5–7, 15
- Q**
- quality aspects 95–104
 - chunk model 103–4
 - labelling 96–9
 - stereotypes 99–103, 105–6
- R**
- refactoring tasks 64–5
 - relationship concepts 193–5
 - repositories 8
 - representations
 - deep instantiation 114
 - development methodologies
 - 13–15, 16, 134
 - infrastructure 21–6, 30, 32–41
 - powertypes 124–6
 - requirements specification documents
 - 135–6, 141, 150, 164–7
 - resources 92, 142
 - roles 171, 192
- S**
- seamless transitions 31–2
 - semantics 92–4
 - SEMDM *see* ISO/IEC 24744
 - serialization strategies 12
 - set-based representation 125, 126
 - shallow instantiation 113
 - situational method engineering (SME)
 - 163, 164–5
 - slot values 57, 169–72
 - SME *see* situational method engineering
 - software development *see* development methodologies
 - software items 189
 - software process improvement (SPI)
 - 9, 58
 - specialization
 - conflicts 85–6
 - dynamic 121
 - infrastructure 38–9
 - notation 196
 - static 121
 - SPEM 59–63
 - assumptions 75
 - conflicts 83–4
 - development methodologies
 - 9–10, 134
 - new approaches 111–12
 - SPI *see* software process improvement
 - stage classes 146–7, 182, 183–6

standardization
 conflicts 84–90
 development methodologies
 140–1, 177
 notation 181
 static specialization 121
 stereotypes
 conflicts 99–103
 powertypes 125, 127
 quality aspects 105–6
 strict metamodeling 42
 Structure–Process–Outcome model
 54
 subject under study (SUS) 21–7,
 30, 32–5, 38, 68–9
 support classes 154–5
 SUS *see* subject under study
 syntax 28–9, 55

T

tagging contexts 69
 tailoring 5–6
 tasks
 conflicts 84–5, 87, 89, 92–4
 development methodologies
 9–10, 139–40, 142–4, 154,
 157–61, 166–71
 infrastructure 41, 47–8
 metamodeling 60–1, 64–5
 new approaches 112–13, 120–3
 notation 187
 teams 191
 techniques 187
 templates
 conflicts 92, 94
 development methodologies 135,
 137, 141–2, 164–5
 powertypes 122–3
 time cycles 147, 183–4
 tools 192
 toolsets 69
 topological containment 198, 199
 transformation 65–8, 70, 72

tuples 47
 Type–Instance relationship 17
 Type–Object models 76, 77–8, 105

U

Unified Modeling Language (UML)
 assumptions 75
 conflicts 83–4, 89, 91
 development methodologies 13,
 17, 134–5, 142, 159–61, 164
 guidelines and solutions 104–5
 infrastructure 25–33, 35–6,
 42–3, 75–6
 metamodeling 54–7, 61–3,
 66–7
 new approaches 111–12, 115,
 127
 ontological metamodels 79–83
 quality aspects 96, 99–103
 stereotypes 99–103, 105–6
 terminology and culture clashes
 76–8
 Unified Modeling Library 95

V

values 47
 vertical realignments 85–7
 veterinary management system
 37–41
 virtual mapping 77

W

whole–part relationship 144–8,
 197–8
 Wikipedia 43
 work performances 195
 work products
 development methodologies
 13–15, 16
 metamodeling 53, 55, 58
 notation 187–90
 product classes 148–51

work units

- development methodologies
13–15, 16, 169–70, 176
- metamodelling 58–9
- notation 186–7
- process classes 144–8

X

- Xome Modelling Infrastructure
(XMIS) 45–8, 68–72
- XQuery 67