

Index

A

abstract syntax 72
 abstracting mapping
 defined 54
 introduction 21
 when to use 87
 abstraction
 advantages 36
 benefits 31
 versus classification 27
 defined 13
 examples 28, 29
 forms 30, 31, 33
 and generalization 27
 and hardware platforms 93
 language 29
 language versus subject matter 32
 and long-hop mappings 112
 mapping between layers 3
 and mapping functions 85
 and metamodels 38
 and model portability 113
 and model-driven process 120
 raising level 2, 4
 and refining mappings 54
 and reversibility of mappings 87
 and short-hop mappings 112
 and software platforms 93
 and vertical mappings 54
 abstraction hierarchy 117
 action language 95
 Agile Alliance 103, 105
 Agile Manifesto 104

agile MDA
 and adding code to models 100
 defined 103
 and Executable UML 109
 introduction 22
 agile methods 22, 104
 analysis model
 examples 29, 31, 97
 sample information 49
 architectural mismatch 9, 92
 Arendt, Hannah 135
 automated mappings 62
 automated transformations 47, 52

B

Balcer, Marc 2, 143
 banking model
 analysis model 49
 design model 50
 design-level class diagram 35
 domain-level class diagram 29
 PIM 29
 PSM 35
 and security model 30, 97, 113
 and UML metamodel classes 41
 Beck, Kent 105, 143
 Bock, Conrad 143
 bridges
 between layers of abstraction 10
 building 121
 examples 125
 as mapping functions 23, 113

C

- Capability Maturity Model (CMM) 107
- classification
 - versus abstraction 27
 - advantages 36
 - defined 13
 - examples 29
 - and generalization 27
 - and metamodels 38
- cobblers' children xv, 26
- code
 - adding to models 84, 85
 - and archetypes 99
 - and executable models 91
 - versus executable models 22
 - incorporating into MDA architecture 89
 - versus models 11
 - and PSMs 100
 - and subject matters 115
 - See also* glue code
- composed realization 15, 114
- constraints
 - as part of UML profile 77
 - back-propagation 117
 - defined 20
 - examples 117
 - verifying 118
- correlations 57

D

- design model
 - example 35
 - sample information 50
- design-time interoperability
 - defined 10
 - introduction xv
- discriminating marks 123
- domain chart
 - defined 113
 - example 114
 - introduction 23
- domain-specific languages (DSLs) 14

E

- EJBs
 - See* Enterprise Java Beans (EJBs)
- Enterprise Java Beans (EJBs) 48
- entity bean
 - defined 48
 - example mapping functions 50, 51
 - examples 50
- executable models
 - versus code 22, 91
 - defined 91
 - introduction 21
 - reasons to use 92
 - verification 96
 - and verification gap 106
- Executable UML
 - as concurrent specification language 96
 - defined 93
 - essential components 93
 - execution engine 100
 - introduction 22
 - and mappings 97
 - and model translation 96
 - and OCL 77
 - and translatability 96
- Extreme Programming (XP) 103, 105

F

- fan-out 87
- formalizing knowledge
 - and abstraction 4
 - and models 121
 - primary activities 120
- four-level metamodel hierarchy
 - described 41
 - illustrated 42

G

- Garlan, David 9, 143
- generalization
 - and abstraction 27
 - and classification 27

examples 29
introduction 28
and metamodels 38
glue code 7, 9

H

horizontal mappings 55
Human Usable Textual Notation (HUTN)
72
HUTN
See Human Usable Textual Notation
(HUTN)

I

intermediate model
example 116
manual changes 115
as read-only artifact 85
interoperability
design-time xv, 10
of metamodels 44
run-time 8
invariant 78

J

join points 58

L

language
contents 72
graphical notations 79
and MOF 73
reasons for building 70
and UML profiles 74
who defines 71
legacy code 87, 89
long-hop mapping
defined 112
disadvantages 117

M

M0 layer 41
M1 layer 41
M2 layer 42
M3 layer
introduction 43
and MOF 43
mapping chain
and constraints 118
defined 111
introduction 23
and marking models 116
mapping functions
archetype-based approach 52
declarative approach 53
defined 48
determining 48
examples 50
imperative approach 52
introduction 16
and mappings 47
and marking models 18
and marks 123
and merging models 97
and metamodels 51
model-driven approach 53
as repeatable design decisions 48
specifying 121
mapping rules
defined 16
introduction 16
mappings
construction 48
defined 48
introduction 16
and mapping functions 47
reversibility 87
synchronization 48
verifying 123
marking models
defined 61
example 65
identifying 115
introduction 18
and mapping chains 116

- and portability 66
 - representing in UML 80
 - structure 67
- marks
 - defined 61
 - as inputs 18
 - introduction 17
 - and legacy code 89
 - and mapping functions 123
 - as outputs 18
 - partitioning 66
 - representing in UML 80
- MDA
 - See model-driven architecture (MDA) 11
 - See also MDA process
- MDA process
 - building 112
 - charting 113
 - constraints 117
 - defined 111
 - executing 119
 - identifying marking models 115
 - identifying metamodels 115
 - identifying models 114
 - introduction 23
 - mapping chains 116
 - and model verification 124
- Mellor, Steve 2, 143
- merge conflicts 86
- merging mapping
 - and correlations 57
 - defined 55
 - and merging models 97
- meta-case tools 79
- meta-interfaces 98
- metamodels
 - and archetypes 52
 - benefits 37
 - and constraints 118
 - defined 38
 - examples 15, 38
 - hierarchy 42, 43
 - identifying 115
 - interoperability of 43
 - introduction 14
 - and language definitions 72
 - and M2 layer 42
 - and mapping functions 51
 - and models 14
 - and MOF 43
 - and PIMs 34
 - and platforms 14
 - types of mappings 53
 - and UML profiles 75
- Meta-Object Facility (MOF)
 - building languages with 73
 - core concepts 73
 - defined 43
 - and language building 73
 - and UML 43
 - versus UML 44
 - and UML metamodel 15
 - and XMI 44
- Meta-Zone 38
- migrating mappings 55
- model compiler
 - and application 100
 - and architectural structure 100
 - defined 99
 - introduction 21
 - tasks 100
- model-driven architecture (MDA)
 - versus APIs 98
 - basic infrastructure 44
 - basic vision 10, 133
 - future 129
 - introduction xv, xviii
 - and marks 68
 - and models 14
 - and tool chains 132
 - value proposition 129
- model-driven development
 - building bridges 121
 - example process 124
 - formalizing knowledge 120
 - iterating process 126
 - overview 119
- models
 - as assets xv, 11
 - bidirectional synchronization 87
 - defined 25
 - elaborating 83

executable 106
 good qualities 26
 implementing marks 122
 introduction 13
 manual changes 85
 and metamodels 14
 optimizing 64
 and platforms 14
 portability 113
 projections 33
 protected areas 85
 reuse 113
 reverse engineering 87
 transforming 124
See also intermediate model
 MOF
 See Meta-Object Facility (MOF)

N

network effect 131
 Notation Police 70

O

Object Constraint Language (OCL) 78
 Object Management Group (OMG) xv

P

PIM
 See platform-independent model (PIM)
 platform stack 15
 platform-independent model (PIM)
 defined 34
 example 29
 and Executable UML 93
 introduction 18
 and marks 62
 and PSMs 19
 platforms
 defined 15
 and metamodels 14, 115
 and model-driven development 125

and models 14
 and realizations 15
 platform-specific model (PSM)
 as code 100
 defined 34
 example 35
 introduction 19
 and PIMs 19
 portability
 and mapping chains 112
 and marking models 66
 trade-offs 113
 postcondition 78
 precondition 78
 primitive realization 15
 problem domain
 defined 30, 113
 introduction 23
 and metamodels 115
 profile
 constructing 75
 CORBA 71
 defined 75
 introduction 20
 and language building 69, 74
 PSM
 See platform-specific model (PSM)

Q

quantity marks 64

R

refining mappings 54
 repeatable design decisions 48
 representing mapping
 defined 54
 example 55, 80
 and merging models 97
 requirements gathering 120
 reuse
 at code level 9
 and components 6
 and frameworks 7

- of mappings 51
- and objects 6
- raising level 5
- run-time interoperability 8

S

- security model
 - and banking model 30, 97, 113
 - domain-level class diagram 31, 97
- Selic, Bran 131, 143
- short-hop mapping
 - advantages 116, 117
 - defined 112
- software-platform independence
 - and executable models 92
 - and hardware-platform independence 4
 - and modeling languages 5
 - and models-as-assets 133
- Stachowiak, Herbert 144
- Star Trek 1
- stateful session bean
 - defined 48
 - example 50
 - example mapping functions 50, 51
- stereotype
 - defined 20, 75
 - defining 75
 - mapping to UML 80
 - as part of UML profile 75
- sticky notes 17

T

- tag definitions 80
- tagged values 76
- testing 121, 126
- thin air 114
- tool chain 132
- transformation chain 124
- translatability 95
- translation
 - and Executable UML 96
 - of models 96

U

UML

- See Unified Modeling Language (UML)

Unified Modeling Language (UML)

- abstract syntax 77
- building languages within family 70
- as family of languages 79
- future 134
- and M2 layer 42
- metamodel 14, 15
- metamodel excerpt 38
- metamodel excerpt in MOF 74
- and MOF 43
- versus MOF 44
- profiles 20, 74
- representing marks 80

V

- verification gap 104
- vertical mapping
 - and change in abstraction level 112
 - defined 54
- virtual machine 100

W

- Weis, Torben 144
- well-formedness rules 78

X

XMI

- See XML Model Interchange (XMI)

XML Model Interchange (XMI) 44

XP

- See Extreme Programming