We’d like to hear your suggestions for improving our indexes. Send email to index@oreilly.com.

Symbols

# (protected visibility), 22, 35
* (connector multiplicity), 49
* operator (OCL), 111
- (private visibility), 22, 35
- operator (OCL), 111
+ (public visibility), 22, 35
+ operator (OCL), 111
... (ellipses), 2
attributes and operations, 20
operations, 35
/ (derived attribute), 22
/ operator (OCL), 111
@pre keyword, 113
{ } (curly braces), 2
collections, 17
tagged values, 15
~ (package visibility), 22, 35, 44
« » (guillemots), 2
stereotypes displayed within, 8

A
abs() operator (Integer), 111
abstract classes, 37
communication diagrams, 87
abstract operations, 36
access stereotype, 9, 44
action labels, 91
action states, 101
action-expression, 93
action-label-or-event, 92
actions, 79, 91
pins, 103
activities, 91
protocol state machines and, 99
activity diagrams, 4, 101–110
action states, 101
activity edges, 102–106
decision and merge, 107
fork and join nodes, 108
partitions, 109
signals, 110
subactivity state, 106
activity edges, 102–106
connectors, 106
exceptions, 104
object flows, 102
actor stereotype, 9
actors, 69
aggregation, 29
composition, 30
alt operator, 84
alternate flows (use case documents), 75
alternatives, 84
anchor line (notes), 14
application-level patterns, 46
architectural patterns, 4
arguments (message syntax), 80
artifacts, 61–63
deployment specifications, 67
manifestation of another
classifier, 63
properties, 62
representing types, 62
assert operator, 86
assertions, 86
association classes, 32
association qualifiers, 34
association stereotype, 9
associations, 27–29
classes, 112
component diagram, 58
constraints between
attributes, 28
implicit constraint, 18
named, 29
naming attributes with, 27
relationships between
nodes, 64
roles, 27
use cases and actors, 70
Xor constraint, 18
asynchronous calls, 80
return messages, 81
attribute (message syntax), 79
attributes, 22–23
artifacts, 62
class relationship, 26
components, 57
constraints, 114
constraints between, 28
inlined, 22
naming using association, 27
properties, 24–26
static, 23
attributeType, 23
author tags, 15

B
backup server stereotype, 65
basic flow (use case
documents), 75
become stereotype, 9
behavioral diagrams, 4, 68–76
use case, 68–76
behavioral port, 50
behavioral state machines, 91
bind stereotype, 9
binding, 39
explicit, 40
implicit, 39
black-box view
(components), 61
body keyword, 113
bodyCondition constraint
(operations), 38
Boolean guard condition,
flows, 84
Boolean type (OCL), 111
break operator, 85
brief description (use case
documents), 75

C
call message, 79
call stereotype, 9
class diagrams, 19–21
attributes, 22–23
class relationships, 26
inlined, 22
properties, 24–26
static, 23
class relationships, 26–32
aggregation, 29
association, 27–29
composition, 30
dependency, 26
generalization, 31
realization, 32
class_name (interaction participants), 77
classes, 6, 19
abstract, 37
actor stereotype, 9
association, 32
attributes (see attributes)
class names, pathnames, and scope, 21
disjoint constraint, 18
enumeration stereotype, 10
exception stereotype, 10
ImplementationClass, 11
incomplete constraint, 18
interface stereotype, 11
multiplicity, 21
overlapping constraint, 18
powertype stereotype, 12
process stereotype, 12
with same name in merged packages, 47
signal stereotype, 13
stereotype, 13
structured, 52
tagged values, 15
thread stereotype, 13
type stereotype, 14
utility stereotype, 14
classifier scope attributes, 23
operations, 39
classifiers, 6
classification in OCL expressions, 111
interface, 40
manifestation in artifacts, 63
metaclass, 12
ports, 51
classname (connectors), 48
client stereotype, 65
collaboration occurrences, 54
collaborations, 53–55
use case realization, 73
Collection class, 115
collections
OCL (Object Constraint Language), 115
ordered and unique properties, 25
combined fragments, 83, 88
break, 85
example, 84
option, 85
comments
OCL expressions, 112
requirement stereotype, 12
responsibility stereotype, 12
communication diagrams, 87
links, 87
compartments
components, 57
nodes, 64
complete constraint, 17
completeness of models, 5
component diagrams, 55–61
component modeling, 58–60
component views, 60
components, 55–58
compartments, 6, 55–58
components, 57
document stereotype, 10
executable stereotype, 10
file stereotype, 10
library stereotype, 11
operations, 57
stereotypes, 56
table stereotype, 13
composite property, 26
composite states, 95–98
regions, 96
submachine state, 97
composite structures, 4, 47–55
port, 49–52
structured classes and properties, 52
structures, 48
composition, 30
concat( ) operator (String), 111
concurrent property, 38
concurrent transitions, 98
conditional logic, 83
evaluating constraints in OCL, 114
conditions (see constraints)
connections, 65
connectors, 48
activity diagrams, 106
naming syntax, 48
ports, 50
constraints, 17–18
attributes, 23, 28
common, listed, 17
connectors, 48
invariant stereotype, 11
OCL (Object Constraint Language), 17, 111–116
basic syntax, 111–112
collections, 115
conditionals for evaluating constraints, 114
constraints on attributes, 114
constraints on operations, 113
operations, 36, 38
postcondition stereotype, 12
precondition stereotype, 12
contained classes, component compartment, 57
context (OCL expressions), 111
copy stereotype, 9
create stereotype, 9, 78
critical combined fragment, 84
critical operator, 86
critical region, 86
cycles in interaction overview
transitions, 88
D
database modeling, 5
database stereotype, 65
database tables, 13, 57
datatypes, 7
ports, 51
(see also types)
decision and merge nodes, 88
decision points in activity diagrams, 107
in use case basic flow, 75
decisionInput stereotype, 107
decomposition (interaction participants), 77
def keyword, 115
defaultValue (operation parameters), 36
DenialOfServiceFilter, 51
dependencies
access stereotype, 9
bind stereotype, 9
call stereotype, 9
class dependency, 26
component diagram, 58
components, 59
contained classes in components, 57
derive stereotype, 9
extend stereotype, 10
friend stereotype, 10
import stereotype, 11
include stereotype, 11
InstanceOf, 11
instantiate stereotype, 11
between a node and components, 64
between packages, 44
power type stereotype, 12
define stereotype, 12
refine stereotype, 13
trace stereotype, 13
use stereotype, 14
deployment diagrams, 61–67
artifacts, 61–63
deployment specifications, 66
nodes, 63–67
deployment specifications, 66
derive keyword, 114
derive stereotype, 9
derived attribute, 22
design and architectural patterns, 4
destroy stereotype, 9, 78
destroyed constraint, 18
devices, 66
Diagram Interchange, 3
diagrams
activity diagrams, 101–110
behavioral diagrams, 68–76
use case, 68–76
class diagrams, 19–21
component diagrams, 55–61
composite structures, 47–55
deployment diagrams, 61–67
interaction diagrams, 76–89
notes, 14
package, 42–47
statechart, 90–100
static modeling, 6
UML 2.0, 4
using only what’s needed, 4
direction (operation parameters), 35
directionality of associations, use cases and actors, 70
disjoint constraint, 18
do (action label), 92
document stereotype, 10, 56
documentation property, 16
dynamic specifications, 6

E
effect activities, 93
protocol state transitions and, 99
dependence association, 85
dependent stereotype, 56
dependent association, 85
dependent stereotype, 56
enumeration stereotype, 10
events
create stereotype, 9
destroy stereotype, 9
exception handler node, 105
exception handling, found messages, 81
exception pins, 104
exception stereotype, 10
executable stereotype, 10, 56
executing environments, 66
device node with, 66
exists ( ) operation (Collection), 116
exit (action label), 92
explicit binding, 40
extend stereotype, 10
extending UML, 5
stereotypes, 8–14
extension use cases, 72

F
facade stereotype, 45
file stereotype, 10, 56
floor ( ) operator (Real), 111
forAll ( ) operation (Collection), 116
fork nodes, 88, 108
forking or joining in state transitions, 98
found messages, 81
for example, 83
framework stereotype, 10, 46
friend stereotype, 10
fully qualified component names, 55

G

generalizations
class relationships, 31
classes with same name in merged packages, 47
component diagram, 58
use case, 71
global stereotype, 10
guard (transitions), 93
guard-condition (action label or action), 92
guarded property, 37

H

happy path, 75
hardware components, 66

I
.icons representing stereotypes, 8
if-then-else execution, 84
implementation stereotype, 11
ImplementationClass, 11
implicit binding, 39
implicit constraint, 18
implies keyword, 114
import stereotype, 11, 45
in keyword, 115
include (action label), 92
include stereotype, 11, 72
incomplete constraint, 18
inheritance
operations, 36
showing with
generalization, 31
init keyword, 114
inlined attributes, 22
input pins, 104
instance scope
attributes, 23
operations, 39
InstanceOf, 11
instances
artifact, 62
class multiplicity and, 21
destroyed constraint, 18
new constraint, 18
structured classes, 53
transient constraint, 18
instantiate stereotype, 11
Integer type (OCL), 111
interaction diagrams, 4, 76–89
communication diagrams, 87
interaction overview, 88
sequence diagrams, 76–86
timing diagrams, 90
interaction operators, 83
alt, 84
assert, 86
break, 85
critical, 86
loop, 87
neg, 86
opt, 85
par, 85
interaction overview diagrams, 88
interfaces, 7, 40
class realizing, 32
communication diagrams, 87
component, 57, 59
expanded, representation of, 40
icon representation of, 41
interface stereotype, 11
ports, 50
invariant conditions, 83
protocol state machines, 99
invariant stereotype, 11
isQuery property, 37
J
J2EE application server, node stereotyped as, 66
J2EE modeling, 5
join nodes, 88, 109
joining (state transitions), 98

L
languages, mapping UML to, 5
let and in keywords, 115
library stereotype, 11, 56
lifelines (object)
  invariant conditions in sequence diagrams, 83
  sequence and interaction overview diagrams, 88
states, in timing diagrams, 90
links
  association stereotype, 9
  communication diagrams, 87
  destroyed constraint, 18
  global stereotype, 10
  local stereotype, 11
  new constraint, 18
  parameter stereotype, 12
  self stereotype, 13
  transient constraint, 18
  local stereotype, 11
  location property, 16
  loop operator, 87
  looping logic, 83
  loops, 88
  lost messages, 81
  example, 82

M
manifest values in a Java .jar file, 57
manifestation of component as artifact, 63
mapping of UML to a particular language, 5
MDA (Model Driven Architecture), 4
merge (action states), 107
merge nodes, 88
merging packages, 47
messages, 79–81
  become stereotype, 9
  copy stereotype, 9
  create stereotype, 9
  destroy stereotype, 9
  lost and found, 81
  return, 81
metaclass stereotype, 12
methods, 3
  operations vs., 36, 40
Model Driven Architecture (MDA), 4
model stereotype, 12
modeling
  components, 58–60
  general guidelines, 4
  static, with UML, 6
multiplicity attribute, 23
  indicated with properties, 24
class, 21
composite structures, 49
  operation parameters, 35
  specifying for each class in associations, 28

N
name (connectors), 48
named associations, 29
names
  class, 21
  component, fully qualified, 55
  neg operator, 86
  negative fragment, 86
  nested packages, visibility and, 43
network failure causing undelivered message, 81
new constraint, 18
nodes, 7, 61, 63–67
deployed components, 64
exception handler, 105
flow final node, 108
fork, in activity diagrams, 108
join, 109
modeling, 64
multiple compartments, 64
object, 102
specialized, 66
devices, 66
execution environments, 66
note anchor line, 14
notEmpty() operator, 112, 116
notes, 14
connector constraints, 48
constraints in, 17
requirements stereotype, 12, 15
null association, 112

O
Object Constraint Language (see OCL)
object flows, 102
Object Management Group (OMG), 1
UML specifications, 3
object node, 102
object.name (interaction participants), 77
object-oriented programming (OOP), 1
objects
communication diagrams, 87
creation and deletion, 78
messages passed to, 79–81
Observer/Observable collaboration, 53
collaboration occurrence, 54
OCL (Object Constraint Language), 3, 17, 111–116
basic syntax, 111–112
collections, 115
conditionals, 114
constraints on attributes, 114
constraints on operations, 113
OMG (Object Management Group), 1
UML specifications, 3
operating-system processes, 12
operations, 3, 34–41
abstract, 36
Collection class (OCL), 115
components, 57
constraints, 38, 113
interfaces, 40
methods vs., 36
polymorphism, 36
properties, 37
scope, 39
syntax, 35
user-defined values, 35
template classes, 39
opt operator, 85
options, 85
order (attributes), 24
output pins, 103
overlapping constraint, 18

P
package diagrams, 42–47
contents and their visibility, 43
dependencies, 44
merging packages, 47
package stereotypes, 45
tagged values, 47
visibility of package elements, 42–44
package stereotype, 10
package visibility, 22, 35
packages
  framework stereotype, 10
  model stereotype, 12
  stub stereotype, 13
  subsystem stereotype, 13
  par operator, 85
  parallel fragment, 85
  parallel interactions, 88
  parameter stereotype, 12
  parameterList (operations), 35
  parameterName (operation parameters), 35
  parameters, action label or action, 92
  parameterType (operation parameters), 35
  participants in sequence diagrams, 77
  partitions, 109
  pathnames, class, 21
  persistence property, 16
  pins, 103
  polymorphism, 36
  ports, 49–52
  interfaces, 50
  realizing, 50
  typing, 51
  post keyword, 113
  postcondition constraint (operations), 38
  postcondition stereotype, 12
  postconditions
    transitions in protocol state machines, 99
    use case documents, 76
  powertype stereotype, 12
  pre keyword, 113
  precondition constraint (operations), 38
  precondition stereotype, 12
  preconditions
    transitions in protocol state machines, 99
    use case documents, 75
private visibility, 22, 35
  package elements, 43
  process stereotype, 12, 56
  profiles, 5, 8
  properties, 24–26
    artifacts, 62
    attribute, 23
    operations, 37
    ordered and uniqueness,
      collection types for, 25
      structured classes, 52
      UML, pre-defined, 16
    user-defined, for elements, 15–16
  protected visibility, 22, 35
  package elements, 43
  protocol keyword, 99
  protocol state machines, 91, 98
  rules for, 99
  pseudostates, 100
  public visibility, 22, 35
  package elements, 43
Q
  qualifiers, association, 34
R
  readOnly property, 25
  Real type (OCL), 111
  realizations
    class relationships, 32
    component interfaces, 59
    ports, 50
    realization stereotype, 56
    use cases, 73–74
  redefines property, 25
  refine stereotype, 12, 74
  regions, 96
  reject() operation (Collection), 116
  requirements stereotype, 12
  use with notes, 15
device, 66
extend, 73
include, 72
nodes, 65
package, 45
package dependencies, 44
predefined, and UML elements to which they apply, 9–14
refine, 74
selection, 103
stereotype, 13
transformation, 104
stick-and-lollipop notation (interface realization), 32, 59
String type (OCL), 111
structured classes, 52
structures, 48
subactivity state, 106
submachine states, 97
subsets property, 25
substring( ) operator (String), 111
subsystem stereotype, 13, 46, 57
realization of an interface, 41
subsystems, 7
synchronous calls, 80
system boundary, 71
system stereotype, 13, 46
systems
requirements, 12
use cases, 7
(see also use case diagrams)

T
table stereotype, 13, 57
tag element, 16
tagged values, 15–16
manifest values in Java .jar file, 57
package with, 47
syntax, 15
tag element, 16
UML predefined properties, 16
value element, 16
template classes, 39
textual representation, package contents, 43
thread stereotype, 13
timing diagrams, 4, 90
trace stereotype, 13
transformation stereotype, 104
transient constraint, 18
transitions (state), 93–95
action states, 102–106
activity diagrams, decision and merge, 107
concurrent, 98
protocol state machines, 99
signals, 94
triggers
state transitions, 93
transitions in protocol state machines, 99
type stereotype, 14
types, 7
artifacts as, 62
OCL, 111

U
UML, 1
mapping to a particular language, 5
OMG specifications, 3
UML 2.0, 4
UML Infrastructure, 3
UML Superstructure, 3
Unified Modeling Language (see UML)
union property, 25
unique attributes, 24
<table>
<thead>
<tr>
<th>use case diagrams, 68–76</th>
<th>version tags, 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>modeling use cases, 71–73</td>
<td>views, component, 60</td>
</tr>
<tr>
<td>realization, 73–74</td>
<td>visibility</td>
</tr>
<tr>
<td>use case documents, 75–76</td>
<td>operations, 35</td>
</tr>
<tr>
<td>use cases, 69</td>
<td>operations defined for an</td>
</tr>
<tr>
<td>use case documents, 68, 75–76</td>
<td>interface, 41</td>
</tr>
<tr>
<td>use cases, 7, 69</td>
<td>package elements, 42–44</td>
</tr>
<tr>
<td>actors, 69</td>
<td>visibility attribute, 22</td>
</tr>
<tr>
<td>modeling, 71–73</td>
<td>W</td>
</tr>
<tr>
<td>extension, 72</td>
<td>web application deployment</td>
</tr>
<tr>
<td>generalization, 71</td>
<td>information, 67</td>
</tr>
<tr>
<td>inclusion, 72</td>
<td>white-box view</td>
</tr>
<tr>
<td>use case documents vs., 68</td>
<td>(components), 60</td>
</tr>
<tr>
<td>use stereotype, 14</td>
<td>X</td>
</tr>
<tr>
<td>utility stereotype, 14</td>
<td>Xor constraint, 18</td>
</tr>
<tr>
<td>value element, 16</td>
<td>variables, OCL, 115</td>
</tr>
</tbody>
</table>