Aspect's sections

aspect Observer {
structure { ... }
instantiations { ... }
messages { ... }
}

Structural View

structural view	structure { ... }
class	class Observer { ... }
partial class	class |Observer { ... }
attribute	int myAttribute
public operation	+ startObserving(|Subject)
private operation	- privateOperation()
protected operation	~ |update()
associations section	associations { ... }
association named mySubject	|Observer -> 0..1 |Subject { mySubject }
unidirectional association	|Observer -> |Subject { mySubject }
bidirectional association	|Observer & |Subject { mySubject }
super type	class Derived : Supertype { ... }
abstract class	abstract class AbstractClass { ... }

Instantiations

instantiations section	instantiations { ... }
depends on	dependsOn ZeroToMany { ... }
extends	extends ZeroToMany { ... }
class mapping	|Data -> |Subject
operation mapping	|Data<|modify> -> |InterceptStateChange<|modifyData>
attribute mapping	|Data<sourceAttribute> -> |InterceptStateChange<targetAttribute>

Message View Headers

message view declaration	messageView |Data.add(|Associated a) { ... }
aspect message view	aspectMessageView initializationAssociation { ... }
message view reference	mesageView |Data.|create() affectedBy initializeAssociation

Control Flow in Messages View

alt combined fragment	alt [" i > 5 "] { ... }
opt combined fragment	opt [" i > 5 "] { ... } else { ... }
loop combined fragment	loop ["i = 0; i < 10; i++"] { ... }

Message View Body

lifelines declaration	lifelines { ... }
reference class lifeline	ref target:StockGUI
association lifeline	assoc mySet:Set
attribute lifeline	attr s:size
reference local variable	ref target:StockGUI { window:Window }
attribute local variable	ref target:Map { int size }
attribute with value local variable	ref s:Socket { String ip = "localhost" }
message ocurrence	add(parameter)
message ocurrence with assignment	result := remove(a)
return message	return result
interaction	target => mySet { result := remove(a) }
original behaviour ocurrence	target => *
destruction ocurrence	socket => X

Message View Aspects

pointcut	pointcut create(mySocket)
advice	advice { ... }

By **mjorod**

cheatography.com/mjorod/

Published 10th February, 2014.
Last updated 1st November, 2014.
Page 1 of 1.