

mechanism to register an appropriate instance's interest in receiving notification when the attribute generates an event. Also provides the mechanism to remove interest in receiving notification for an event. The instance registering interest in receiving event notification must implement the appropriate interface to match the listener type described in the signal descriptor.

[0339] A method instance includes an execute(ParameterList) operation that attempts to perform the operation by executing the operation implementation. First all access and precondition constraints are checked for the method implementation and parameter implementations. Then the method is attempted. If the method implementation is successful, the postcondition constraints are checked for the method implementation and parameter implementations.

[0340] An operation event with the status of the preconditions is fired to all interested parties after execute(ParameterList) method performs all the precondition tests. A second event is sent at the successful completion of the operation. A third event is fired with the status of the postconditions or the operation. A fourth event is fired only if a failure occurs during the execution of the operation.

[0341] A signal implementation of the present invention is an assembly of operation implementations related to sending events. A signal implementation has a one-to-one association relationship with a signal descriptor that is the signal descriptor for which this implementation provides access to an implementation. A signal implementation has a zero-to-many aggregation relationship with a listener registration operation that calls upon the operation held by the signal implementation to perform registration for event notification. A signal implementation has a zero-to-many aggregation relationship with listener deregistration operation implementations that calls upon the operation held by the signal implementation to remove an instance from registration for event notification. A signal implementation has a zero-to-many aggregation relationship with listener access operation accessors that call upon the operation held by the signal implementation to list all instances registered for event notification. A signal implementation has a zero-to-many association relationship with registration constraints that are each an implementation of constraints to check for certain conditions of the objects being registered. Registration constraints may also be used to restrict the type and number of listeners.

[0342] A signal implementation may include the following operations: register(Instance) that attempts to register the given instance as a listener. Registration constraints are applied to the listener before performing the registration event, deregister(Instance) that attempts to remove the given instances from listening to event notifications, and listListeners() that returns an enumeration to the listeners currently listening for events from this signal.

[0343] A registration event with the status of the registration constraints is fired to all interested parties after register(Instance) method performs all the precondition tests. A second event is sent at the successful completion of the registration operation. Events are also fired when deregister(Instance) and listListener() methods are called.

[0344] A signal instance of the present invention contains the listeners associated with an instance that generates

events. A signal instance has a one-to-one association relationship that is the implementation for which the signal instance is an instance. A signal instance has a zero-to-many association relationship with listeners that are holders for listeners implementing the type of the listener to receive events.

[0345] A signal instance includes a sendEvent(event) that sends an event to all listeners

[0346] A listener event is fired whenever a listener is added, changed, or removed.

[0347] A constructor implementation of the present invention passes through the parameters and failures given by the operation implementations it will perform during construction. A constructor implementation has a one-to-one association relationship with a constructor descriptor that is the constructor descriptor for which this implementation provides an implementation. A constructor implementation has a zero-to-many association relationship with failures that may be thrown if the constructor parameters do not match or if the constructor operation implementations throw an error during execution.

[0348] A constructor implementation inherits attributes from operation implementation but does not add new attributes.

[0349] A constructor implementation includes newInstance(ParameterList) operation that is the method call to perform a construction with the given values as parameters. The parameters in the parameter list must match the type and be accepted by the parameter implementations in order for construction to be successful.

[0350] An implementation change event is fired whenever an attribute value is added, changed or removed from the constructor implementation.

[0351] A construction event with the status of the preconditions is fired to all interested parties after newInstance(ParameterList) method performs all the precondition tests. A second event is sent at the successful completion of the operations related to construction. A third event is fired with the status of the postconditions. A fourth event is fired only if a failure occurs during the execution of the construction.

[0352] A constructor instance of the present invention may be unnecessary for many applications. The constructor implementation performs the work associated with destruction. The constructor implementation can always be retrieved from an instance by going through the metamodel implementation for that instance. A constructor instance does provide support for activities that model current reflection packages available in many programming languages.

[0353] A constructor instance has a one-to-one association relationship with a constructor implementation that is the implementation for which the constructor instance is an instance

[0354] Depending on the needs of the application, a destructor implementation may simply destroy the virtual model to free memory, or a destructor can first destroy any persistent data related to the virtual model, and then destroy the virtual model. The determination is defined by the operation implementations assigned to the virtual destructor implementation.