

contains the implementations for the constraints defined by the datatype descriptor. A datatype implementation can be used to create a datatype instance. The modeling tool using datatype instances must handle failures thrown by the constraint implementations placed upon that data instance.

[0301] A datatype implementation has a one-to-one association relationship with a datatype descriptor that is a reference to the datatype descriptor the datatype implementation implements. A datatype implementation has a zero-to-many association relationship with access constraints that are an implementation of constraints placed upon the access allowed to values. A datatype implementation has a zero-to-many association relationship with occurrence constraints that are an implementation of constraints placed on the number of values a data instance may hold. A datatype implementation has a zero-to-many association relationship with value constraints that are each is an implementation of constraints placed upon the values a datatype instance may hold. A datatype implementation has a zero-to-one aggregation relationship with a premutator operation implementation that is an implementation of the operation(s) to execute just before changing the data value. A datatype implementation has a zero-to-one aggregation relationship with a postmutator operation Implementation that is an implementation of the operation(s) to execute just after changing the data value. A datatype implementation has a zero-to-one aggregation relationship with a preaccessor operation implementation that is an implementation of the operation(s) to execute just before returning the data value.

[0302] A datatype implementation adds no additional events.

[0303] A datatype instance of the present invention is an instance that holds a piece of data. A datatype instance references its implementation to ensure its definition is enforced. A datatype instance holds only values acceptable to its implementation.

[0304] A datatype instance references the data implementation for which the data instance is an instance. The datatype instance may hold constraint instances or may implement these constraints directly. A fully virtual implementation holds the constraint instances. These constraints may throw data errors if the values fail to meet the restrictions imposed by the constraint.

[0305] A datatype instance has zero-to-many association relationship with values that are holders for instances of data that meet the constraints placed upon them. A datatype instance has zero-to-many association relationship with values access constraint instances that zero or more security checks place on accessing the value. A datatype instance has zero-to-many association relationship with occurrence constraint instances that zero or more checks place on the number of values allowed in this data instance. A datatype instance has zero-to-many association relationship with value constraint instances that zero or more checks place on the values allowed in this data instance.

[0306] A datatype instance may include the following operations: instance `getValue()` that returns the current value. Retrieves the first value if more than one value exists, iterator `getValues()` that returns the current values as an iterator to those values, `setValue(Instance)` that removes any previous value and sets it to the new value after checking all

constraints, `setValues()` that is an array that removes any previous values and sets new values after checking all constraints, `addValue()` that is an instance that add the instance given to the current values after checking all constraints, `removeValue(Instance)` that removes the given instance from the current values after checking to see if the value is being held and checking occurrence constraints, and `clearValues()` that remove all values after checking occurrence constraints.

[0307] A data event is fired whenever a value is added, changed, or removed.

[0308] An attribute implementation simply has to apply all its constraints to any value being set or added. An attribute implementation must apply occurrence constraints when clearing all values or removing a single value. The virtual attribute descriptor must always apply the access constraints whenever used. An attribute implementation must also properly notify the signal implementation whenever a change successfully occurs.

[0309] An attribute implementation has a one-to-one association relationship with an attribute descriptor that is the attribute descriptor for which this implementation provides access to an implementation. An attribute implementation has a zero-to-many association relationship with prechange signal implementations that provide the mechanism to register an appropriate instance's interest in receiving notification when the attribute is about to change. 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. An attribute implementation has a zero-to-many association relationship with postchange signal implementations that provide the mechanism to register an appropriate instance's interest in receiving notification after an attribute has changed. An attribute implementation 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. An attribute implementation has a zero-to-many association relationship with access signal implementations that is provide the mechanism to register an appropriate instance's interest in receiving notification that an attribute value is being accessed. An attribute implementation has a one-to-one aggregation relationship with a data implementation that is the implementation describing the actual data to be held.

[0310] An attribute implementation adds no additional events.

[0311] An attribute instance of the present invention uses the attribute implementation from the model implementation to constrain its values. The attribute instance also contains any attribute listeners registered to listen to it. The model instance also contains listeners for each signal it exposes. If an attribute listener registers with the model instance to listen to all attributes, the model instance will register itself with each attribute instance and will forward all attribute events to its attribute listeners.

[0312] An attribute instance has a one-to-one association relationship with an attribute implementation that is the implementation for which the attribute instance is an