

with at least one number and one symbol. A value constraint could also ensure a composite value does not contain certain characteristics. A product order containing zero-to-many line items could be checked to have at least one line item in order to enter the sendOrder( ) operation. The range constraint used as an example above for occurrences could also be used to enforce value constraints. The value being checked by the constraint is the occurrence count in the occurrence constraint; the value being checked by the constraint is the value in a value constraint.

[0279] A value constraint implementation is a simple extension of constraint implementation. The data implementations vary with the constraint being imposed.

[0280] A constraint instance of the present invention is an instance. A constraint instance performs the actual checks based on its configuration. A constraint instance has a one-to-one association relationship with a constraint implementation that is the constraint implementation of which the constraint instance is an instance. A constraint instance has a one-to-one association relationship with a model instance that is the model implementing the constraint. This model must implement the correct constraint interface to be understood by the constraint instance and must follow the description provided by the constraint implementation.

[0281] A check of a constraint instance is defined by the submodel of constraint and is implemented by the model. A constraint check event occurs when a constraint is applied to a specific value. The event is fired after the constraint to notify of the success or failure of the constraint check.

[0282] An access constraint instance of the present invention extends constraint instance and feature instance. The constraint has a pointer to its parent and may use its parent information to perform the security check. For example, does the user have permission to use an object named 'marketing database'.

[0283] An access constraint instance of the present invention has a one-to-one association relationship with an access constraint implementation that is the access constraint implementation of which the access constraint instance is an instance.

[0284] Constraints may have attributes defined by the metamodel and implemented by the model implementation. Constraint instances, including access constraint instances, do not place any additional attribute requirements on the model implementation of a constraint.

[0285] An access constraint instance include a check( ) operation that checks to see if the current user has the required access privileges. An access constraint defines no additional signals.

[0286] An occurrence constraint instance of the present invention extends a constraint instance and feature instance. The constraint instance has a pointer to its parent and may use its parent information to perform the occurrence check.

[0287] An occurrence constraint instance has a one-to-one association relationship with an occurrence constraint implementation that is the occurrence constraint implementation of which the occurrence constraint instance is an instance.

[0288] Constraints may have attributes defined by the metamodel and implemented by the model implementation.

The constraint instance does not place any additional attribute requirements on the model implementation of a constraint.

[0289] An occurrence constraint instance includes a check(number) operation that checks to see if the given number of occurrences is an allowable count. An occurrence constraint defines no additional signals.

[0290] A value constraint instance of the present invention extends another constraint instance and a feature instance. The constraint has a pointer to its parent and can use any of its parent's attributes as part of the constraint check. It is allowable to have interplay between attributes set on the parent, and values allowed by this constraint.

[0291] A value constraint instance has a one-to-one association relationship with a value constraint implementation that is the value constraint implementation of which the value constraint instance is an instance.

[0292] Constraints may have attributes defined by the metamodel and implemented by the model implementation. Therefore, a value constraint instance does not place any additional attribute requirements on the model implementation of a constraint.

[0293] A value constraint instance includes a check(Value) operation that checks to see if the given value is acceptable to this constraint. The value will generally be the value about to be assigned to the implementation. Some constraints may require more than one value. In these cases the value passed into the check method can be an array, a collection, a map, or some other composite data object.

[0294] A value constraint defines no additional signals.

[0295] A feature implementation of the present invention is an implementation that belongs to (are part of) a larger implementation. Feature implementations contain a reference to the implementation of which they are a part. Features may require access permissions in order to be used.

[0296] A feature implementation has a one-to-one association relationship with a feature descriptor that is a reference to the feature descriptor the feature implementation implements. A feature descriptor has a zero-to-one association relationship with a parent implementation that is the implementation of which the feature implementation is a part. A feature implementation has a zero-to-many association relationship with access constraint instances that the access constraints placed on this feature.

[0297] A feature implementation adds no additional events.

[0298] A feature instance of the present invention is an instance that allows access to the parent instance of which they are a feature. A feature instance has a one-to-one association relationship with a feature descriptor that is a reference to the feature implementation of which the feature instance is an instance. A feature instance has a one-to-one association relationship with a parent instance that is the parent instance of the feature instance.

[0299] A feature instance adds no additional events.

[0300] A datatype implementation represents a model of data. To hold an instance of data, a datatype implementation is used to create a data instance. A datatype implementation