

Meta-modelo CC

1 Introducción

Propósito

El meta-modelo CC (*Constraint Characterization*) constituye una propuesta de meta-modelo para los modelos mediante los cuales se formula la información de caracterización de las restricciones especificadas sobre un meta-modelo de dominio.

2 Clases principales

El meta-modelo CC presenta una estructura convencional, con una clase contenedor principal (`CC_Model`) y una clase raíz (`ConstraintCharacterization`) de la que heredan el resto de clases, conformando en conjunto la jerarquía de clases destinadas al modelado de (los datos necesarios para caracterizar) restricciones.

La Fig. 2.1 muestra las clases contenedor principal y raíz, así como las principales subclases de ésta última.

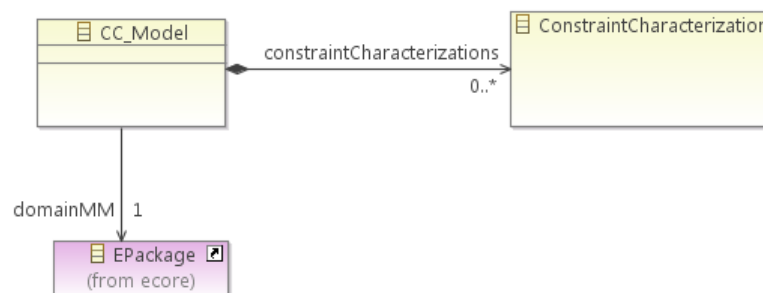


Figura 2.1 – Núcleo del meta-modelo CC

Un modelo conforme a CC posee una única instancia de `CC_Model`, la cual contiene a través de la asociación-composición `constraintCharacterizations` al resto de elementos de modelo, instancias de CC o de cualquiera de sus subclases.

2.1 Clase `CC_Model` (contenedor principal)

El meta-modelo CC define una clase contenedor principal llamada `CC_Model` cuya única instancia en un modelo conforme tiene por finalidad albergar, a través de la asociación-composición `characterizations`, el resto de elementos del modelo, instancias que contienen los datos base para describir el conjunto de caracterizaciones correspondientes a las restricciones especificadas sobre un meta-modelo. Estos elementos de modelo son instancias del resto de clases del meta-modelo, bien de la clase raíz CC o de cualquiera de sus subclases.

Esta clase contenedor principal define además una asociación llamada `domainMM` que sirve para referenciar al meta-modelo de dominio sobre el que se ha especificado el conjunto de restricciones que se desea caracterizar.

2.2 Clase ConstraintCharacterization (clase raíz)

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase permite modelar de forma mínima la caracterización de restricciones de cualquier tipo.

Únicamente con esta clase raíz sería suficiente para formular como modelo el conjunto de caracterizaciones correspondientes a las restricciones especificadas sobre un meta-modelo, pues la información que describe es apta para cualquier restricción, independientemente de su naturaleza, semántica o formulación OCL.

- **Estructura.**

Esta clase define únicamente referencias a la propia restricción (`constraint`), a la clase del meta-modelo de dominio que juega el rol de contexto (`contextClass`) y a la clase del meta-modelo CVD seleccionada para formular posibles incumplimientos suyos (`cvdClass`).

También define atributos para una descripción textual opcional de la restricción (`constraintDescr`) y para especificar la severidad que se desea asignar en caso de violación (`severity`).

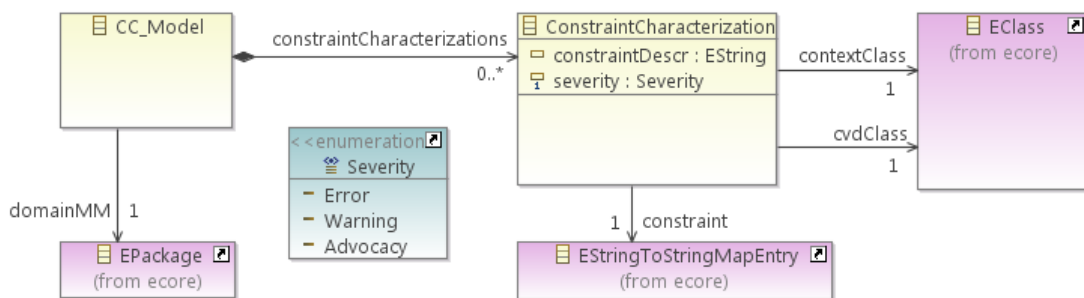


Figura 2.2

- **Relación con el meta-modelo CVD.**

En caso de caracterizar una restricción empleando esta clase, esto es, de forma mínima, para establecer el *mapping* a una clase CVD sólo puede escogerse su clase raíz, puesto que una caracterización mínima no incluye los datos necesarios que habrían de ser asignados a las propiedades de clases CVD más especializadas.

- **Ejemplos en MAST-2.**

Dada cualquiera de las restricciones de integridad especificadas sobre el meta-modelo MAST-2, podría perfectamente modelarse su caracterización mediante una instancia de esta clase.

Por ejemplo:

- i_1_2_a

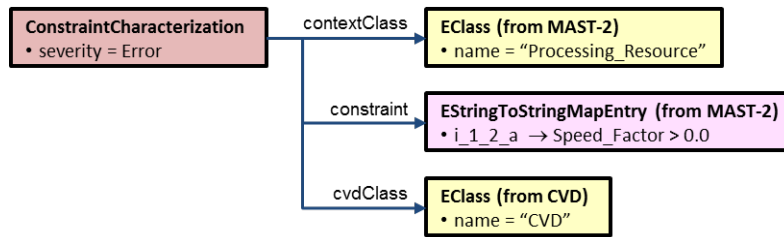


Figura 2.3

Como se mostrará en las siguientes secciones, puesto que los meta-modelos CC y CVD se encuentran íntimamente relacionados, ya que el

El mapeo entre clases de este meta modelo y del meta modelo CVD (representado por la asociación `cvdClass`) mediante el que se establece cómo modelar los datos necesarios para describir incumplimientos de la restricción que se está caracterizando sigue la regla de que la clase CVD escogida no puede encontrarse en un nivel más profundo

Sin embargo, en el meta modelo CC se define toda una jerarquía de subclases de la clase raíz `cc` —representativa de la caracterización mínima de restricción— con el objetivo de posibilitar el modelado de caracterizaciones más completas acerca de restricciones. Así, en lugar de formular mediante ella las caracterizaciones de cada restricción especificada para un meta modelo, será posible asociar a cada restricción la subclase que sea más adecuada para caracterizarla con mayor nivel de detalle. Estas subclases se presentan en la siguiente sección.

3 Subclases de ConstraintCharacterization

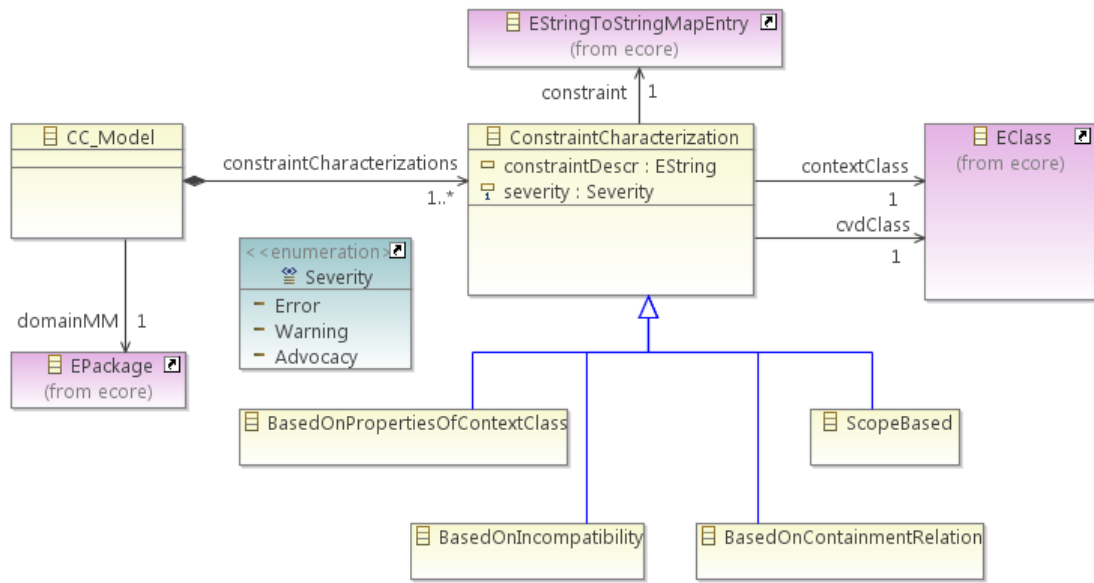


Figura 3.1

3.1 BasedOnPropertiesOfClass

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones especificadas sobre un conjunto de propiedades (atributos y referencias) de la clase contexto.

- **Estructura.**

Junto a las posibilidades de modelado heredadas de la clase `ConstraintCharacterization`, esta clase define una asociación `properties` mediante la que especificar aquellas propiedades involucradas en la restricción.

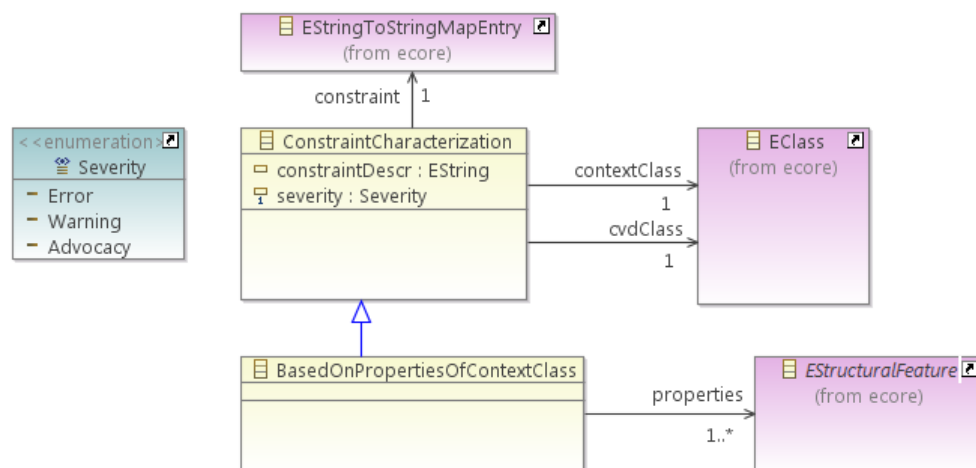


Figura 3.2

- **Restricciones de integridad.**

- Las `EStructuralFeature` especificadas han de pertenecer a la clase contexto.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`BasedOnPropertiesOfClass`).

También sería válido establecerla a cualquiera de sus superclases (en este caso únicamente la clase raíz `CVD`), asumiendo una descripción de incumplimiento más pobre de la que es posible, desaprovechando información contenida en la caracterización de la restricción (la información relativa a las propiedades de la clase contexto involucradas).

- **Ejemplos en MAST-2.**

Sobre el meta-modelo `MAST-2` se encuentran definidas numerosas restricciones de integridad basadas únicamente en propiedades de la clase contexto, y que por tanto su caracterización podría ser perfectamente modelada mediante esta clase.

Algunos ejemplos son:

- `i_1_2_a`

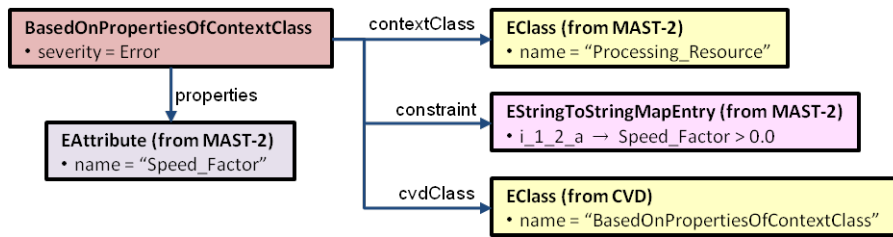


Figura 3.3

o i_1_3_a

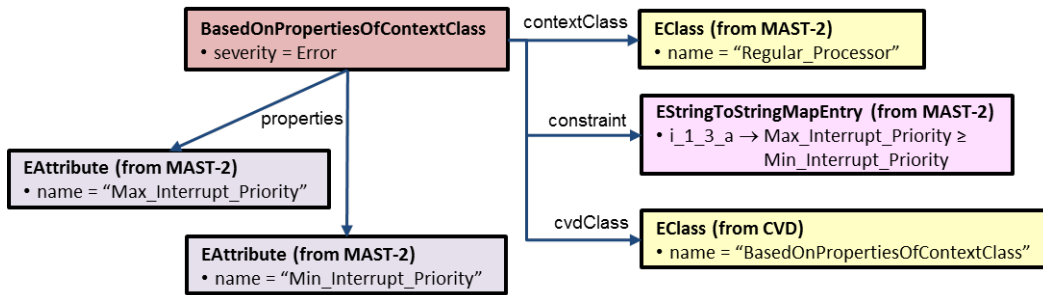


Figura 3.4

3.2 BasedOnIncompatibility

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en establecer incompatibilidades entre subclases de dos clases (típicamente abstractas) del meta-modelo de dominio que se hallan relacionadas con la clase contexto a través de sendas cadenas de asociaciones.

Se puede dar el caso particular de que una de las clases coincida con la propia clase contexto, en cuyo caso la correspondiente cadena de asociaciones sería nula.

- **Estructura.**

Junto a las posibilidades de modelado heredadas de la clase `ConstraintCharacterization`, esta clase define dos asociaciones `refChain1` y `refChain2` mediante las que especificar aquellas cadenas de referencias que enlazan las clases involucradas en la restricción (aquellas sobre las que se establece la incompatibilidad) con la clase contexto.

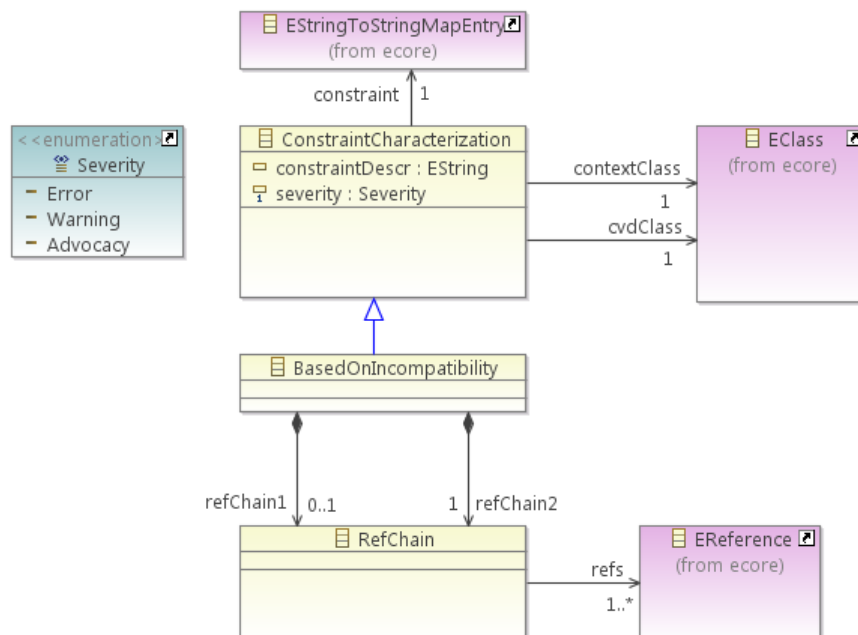


Figura 3.5

La asociación `refChain1` posee multiplicidad opcional para permitir la posibilidad de no ser especificada, indicando así que una de las clases es la propia clase contexto.

- **Restricciones de integridad.**

- La primera `EReference` de cada cadena ha de pertenecer a la clase contexto.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a una de las subclases de la clase CVD del mismo nombre (`BasedOnIncompatibility`) y no a ella misma, pues es abstracta.

~~La subclase ha de escogerse de forma consistente a la incompatibilidad definida por la restricción que se desea caracterizar, según la multiplicidad de la última referencia en cada cadena de referencias.~~

También sería válido establecerla a cualquiera de sus superclases (en este caso únicamente la clase raíz *CVD*), asumiendo una descripción de incumplimiento más pobre de la que es posible, desaprovechando información contenida en la caracterización de la restricción.

- **Ejemplos en MAST-2.**

Sobre el meta-modelo MAST-2 se encuentran definidas numerosas restricciones de integridad basadas en incompatibilidad y que por tanto su caracterización podría ser perfectamente modelada mediante esta clase.

Algunos ejemplos son:

- o *i_3_5_a*

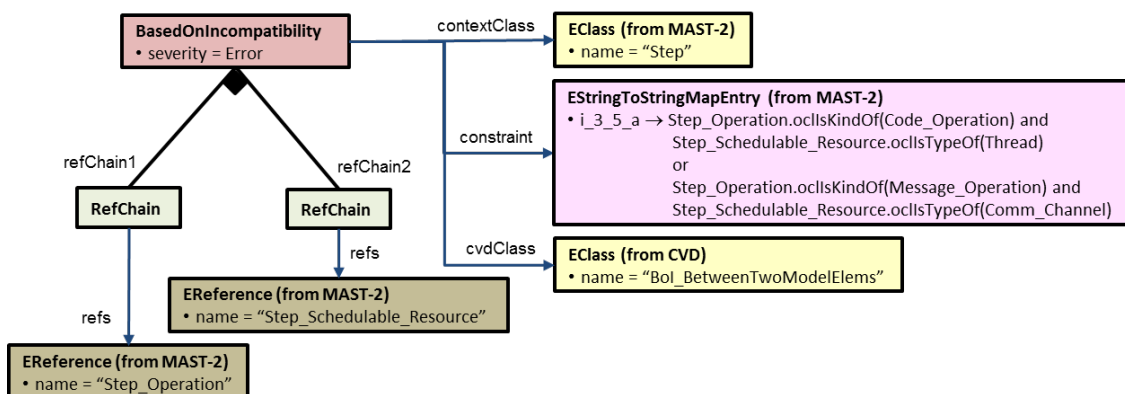


Figura 3.6

- o *i_3_3_a_l*

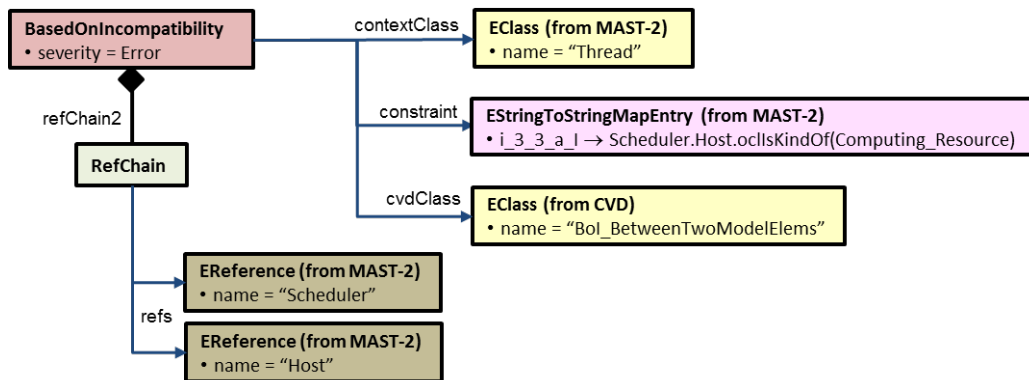


Figura 3.7

3.3 BasedOnContainmentRelation

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en establecer una relación de contención entre dos conjuntos de instancias de una clase.

Cada conjunto viene indicado por el extremo final de sendas cadenas de asociaciones que parten de la clase contexto y la enlazan con la clase que tipa a los elementos de los conjuntos.

- **Estructura.**

Junto a las posibilidades de modelado heredadas de la clase `ConstraintCharacterization`, esta clase define dos asociaciones `refChain1` y `refChain2` mediante las que especificar aquellas cadenas de referencias cuyo extremo final indica los conjuntos de elementos de modelo sobre los que se establece la relación de contención.

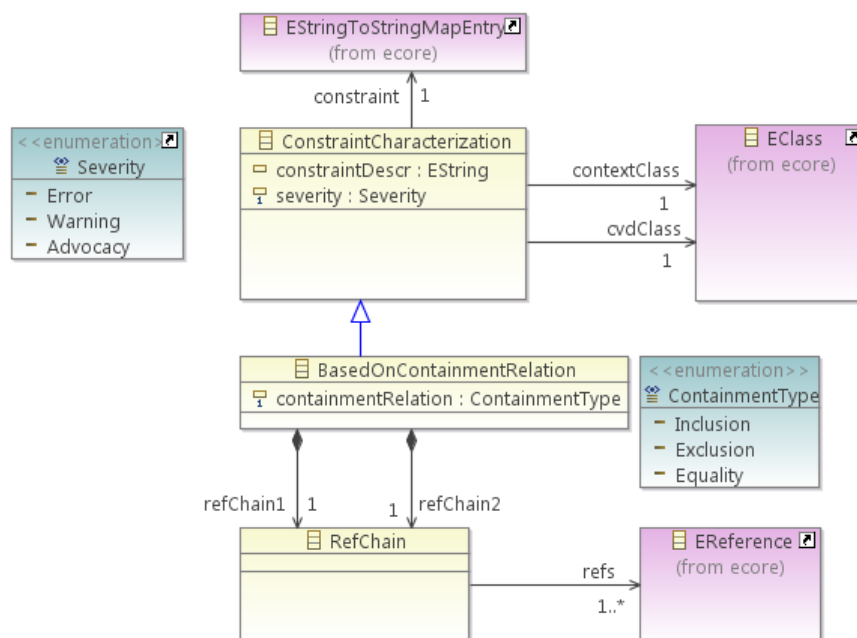


Figura 3.8

- **Restricciones de integridad.**

- La primera EReference de cada cadena ha de pertenecer a la clase contexto.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`BasedOnContainmentRelation`).

También sería válido establecerla a cualquiera de sus superclases (en este caso únicamente la clase raíz `CVD`), asumiendo una descripción de incumplimiento más pobre de la que es posible, desaprovechando información contenida en la caracterización de la restricción.

- **Ejemplos en MAST-2.**

Sobre el meta-modelo MAST-2 sólo se encuentra definida una restricción de integridad basada en contención y que por tanto su caracterización podría ser perfectamente modelada mediante esta clase.

o i_3_1_a

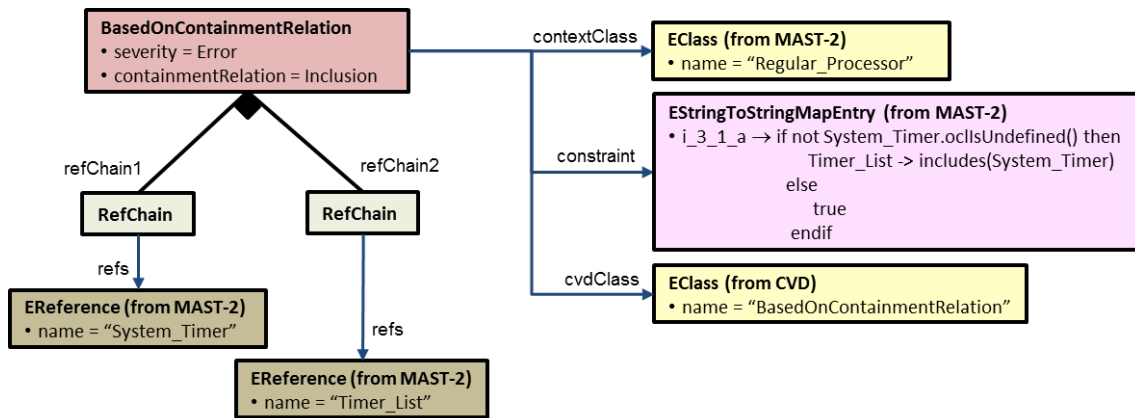


Figura 3.9

3.4 ScopeBased

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones cuya satisfacción depende no sólo del estado de un elemento de modelo, sino también del ámbito o población de elementos de modelo dentro del cual el primero se halla.

- **Estructura.**

Junto a las posibilidades de modelado heredadas de la clase `ConstraintCharacterization`, esta clase define una asociación-composición `scope` que contiene la descripción del ámbito o población de elementos de modelo en cuyo seno se **impone** la restricción.

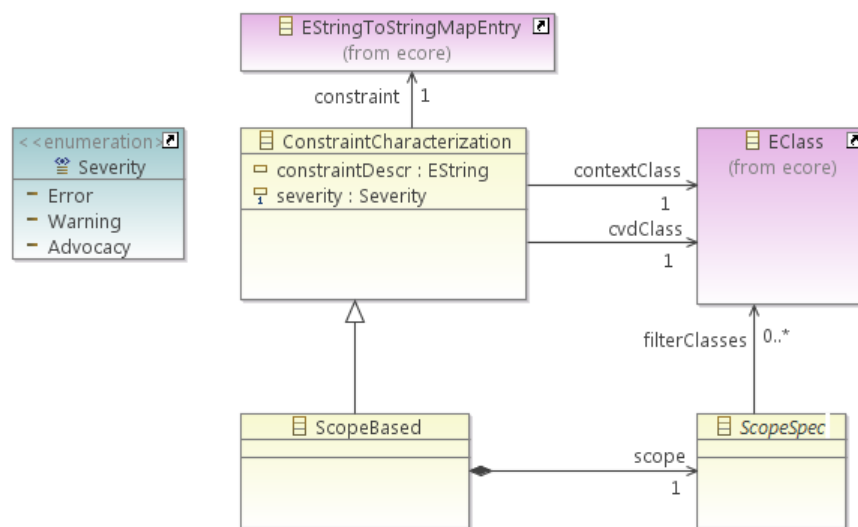


Figura 3.10

En el caso más general, el ámbito puede ser el modelo entero, lo cual se representa mediante la clase `GlobalScope`, mientras que la clase `ReachableScope` sirve para modelar ámbitos más específicos, esto es, poblaciones de elementos de modelo alojados en el extremo de una asociación con cardinalidad múltiple. Esto se consigue indicando un elemento de modelo (`scopeModelElem`) y una referencia de cardinalidad múltiple (`scopeRef`) perteneciente a la clase del `scopeModelElem`. Si el tipo del extremo viene definido por una clase que posee subclases, es posible restringir la población a aquellos elementos que sean exclusivamente instancias de una o más de esas subclases, lo cual se consigue especificando a través de la referencia `filterClasses` qué subclases de filtrado se aplican. Por último, la clase `CombinedScope` representa la unión de dos o más ámbitos básicos.

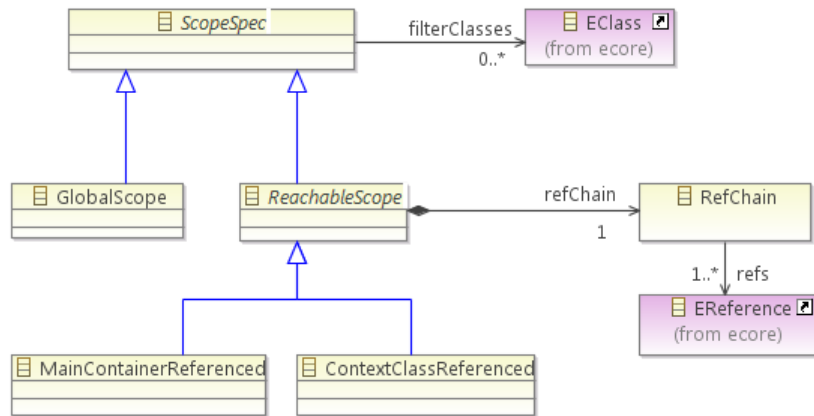


Figura 3.11

- **Restricciones de integridad.**

- .

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`ScopeBased`).

También sería válido establecerla a cualquiera de sus superclases (en este caso únicamente la clase raíz `CVD`), asumiendo una descripción de incumplimiento más pobre de la que es posible, desaprovechando información contenida en la caracterización de la restricción.

- **Ejemplos en MAST-2.**

4 Subclases de BasedOnPropertiesOfClass

El meta-modelo CC define varias subclases de `BasedOnPropertiesOfClass` buscando proporcionar formas de modelar aún en mayor detalle la caracterización de restricciones dentro de la variedad correspondiente a aquellas especificadas sobre propiedades de la clase contexto. La Fig. 4.1 muestra tales subclases, que se exponen a continuación.

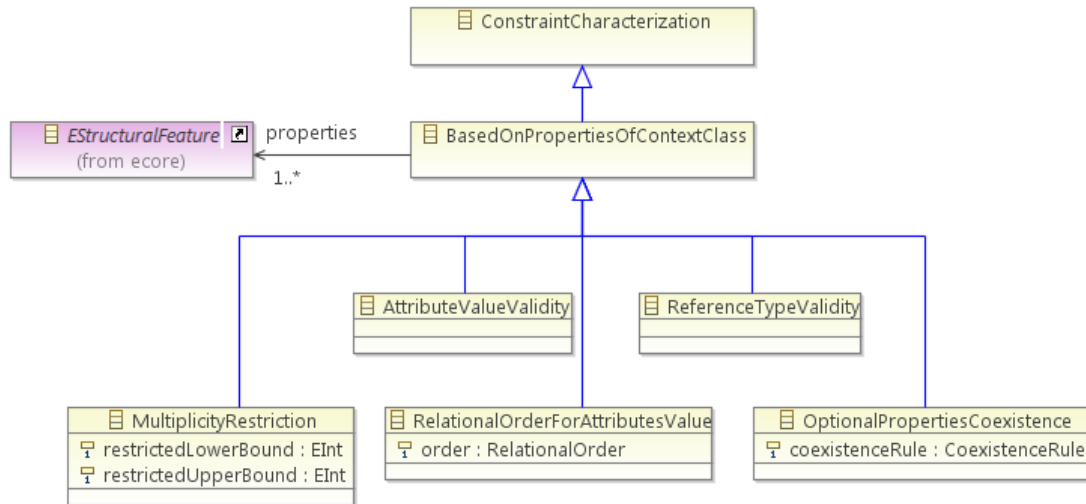


Figura 4.1

4.1 MultiplicityRestriction

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en restringir la multiplicidad de una propiedad de la clase contexto.

- **Estructura.**

Junto a las posibilidades de modelado heredadas de `BasedOnPropertiesOfContextClass`, esta clase define dos atributos de tipo entero (`restrictedLowerBound` y `restrictedUpperBound`) que representan los límites de multiplicidad restringida.

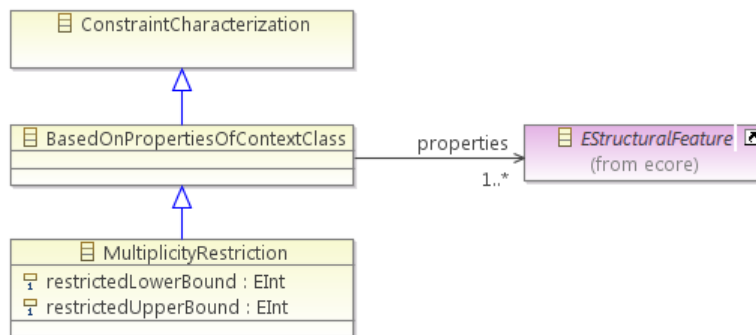


Figura 4.2

- **Restricciones de integridad.**

- Los límites de multiplicidad especificados han de estar en orden y contenidos dentro de la multiplicidad original.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`MultiplicityRestriction`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

-

4.2 AttributeValueValidity

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en restringir el abanico de valores que puede tomar un atributo de una clase.

Típicamente, el atributo es de un tipo primitivo numérico (int, float, etc.), pero también podría tratarse de atributos de tipo no numérico (un tipo enumerado, char, etc.).

- **Estructura.**

Junto a las capacidades de modelado heredadas de `BasedOnPropertiesOfContextClass`, esta clase define una asociación-composición `validityRange` que contiene la descripción del intervalo de validez, indicando sus extremos y su tipo { abierto (... , ...), cerrado [... , ...], semiabierto cerrado por la derecha (... , ...] semiabierto cerrado por la izquierda [... , ...) }.

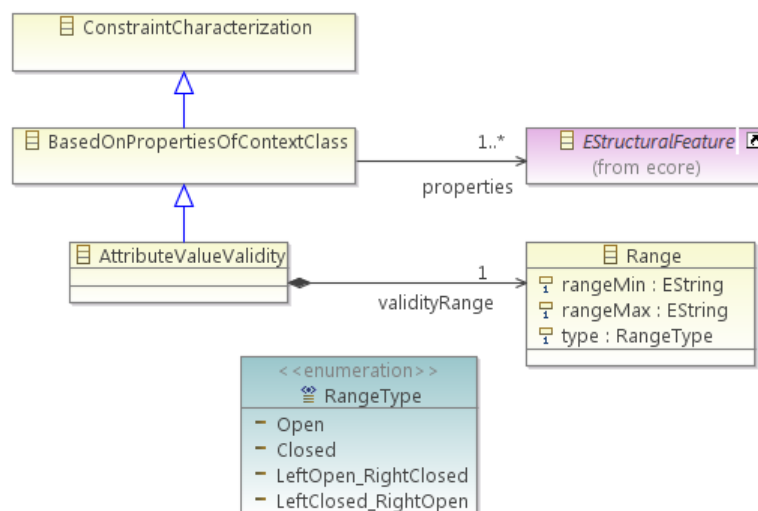


Figura 4.3

- **Restricciones de integridad.**

- A través de `properties` se ha de referenciar a un `EAttribute`.
- La expresión textual de los extremos especificados para el intervalo de validez ha de ser convertible al tipo del `EAttribute` referenciado.
- Éstos han de estar en orden.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`AttributeValueValidity`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

- `i_1_2_a`

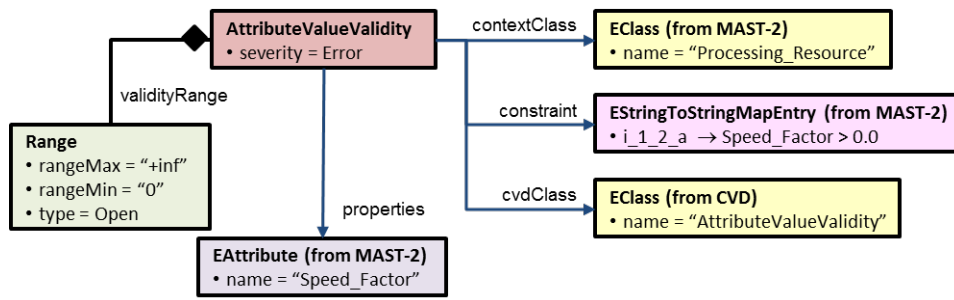


Figura 4.4

4.3 ReferenceTypeValidity

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en restringir el abanico de clases que pueden tipar a una referencia.

Típicamente, esta situación se da cuando en el meta-modelo se ha establecido una clase abstracta como tipo de una referencia pero en determinadas circunstancias sólo se consideran admisibles ciertas subclases de ella.

- **Estructura.**

Junto a las capacidades de modelado heredadas de `BasedOnPropertiesOfContextClass`, esta clase define una asociación `validTypes` mediante la que especificar las clases consideradas admisibles.

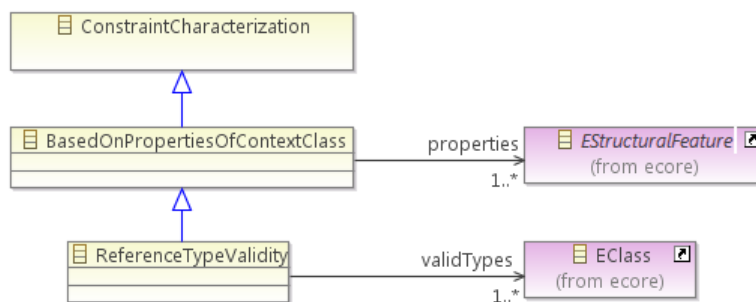


Figura 4.5

- **Restricciones de integridad.**

- A través de `properties` se ha de referenciar a una `EReference`.
- Las `EClass` referenciadas a través de `validTypes` han de ser subclases del tipo de la `EReference` especificada.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`ReferenceTypeValidity`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

-

4.4 AutoRefSymmetry

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en...

- **Estructura.**

Junto a las capacidades de modelado heredadas de `BasedOnPropertiesOfContextClass`, esta clase define un atributo `brokenSymmetry` que representa el tipo de simetría incumplida (simetría, asimetría o antisimetría).

- **Restricciones de integridad.**
- **Relación con el meta-modelo CVD.**
- **Ejemplos en MAST-2.**

4.5 OptionalPropertiesCoexistence

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en imponer sobre un conjunto de propiedades opcionales en una clase la condición de que sean mutuamente excluyentes o de que hayan de estar presentes todas en bloque.

- **Estructura.**

Junto a las capacidades de modelado heredadas de `BasedOnPropertiesOfContextClass`, esta clase define un atributo `coexistenceRule` que representa el tipo de coexistencia impuesta (exclusión mútua o siempre aparecer juntas).

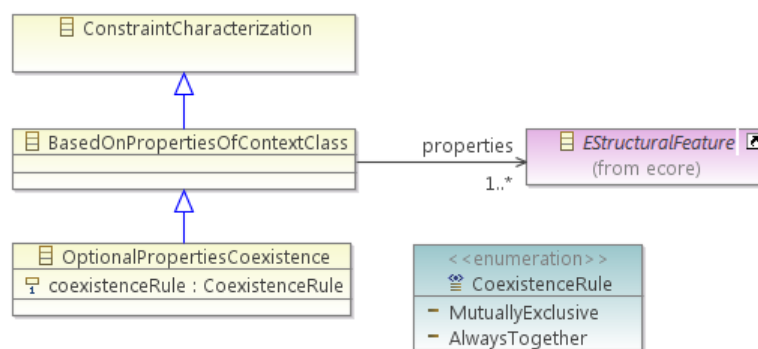


Figura 4.6

- **Restricciones de integridad.**

- A través de `properties` se han de referenciar a `EStructuralFeature` que sean opcionales.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`OptionalPropertiesCoexistence`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

Sobre el meta-modelo MAST-2 sólo se encuentra definida una (desglosada en dos) restricción de integridad basada en coexistencia de propiedades opcionales y que por tanto su caracterización podría ser perfectamente modelada mediante esta clase.

- `i_3_10_a`

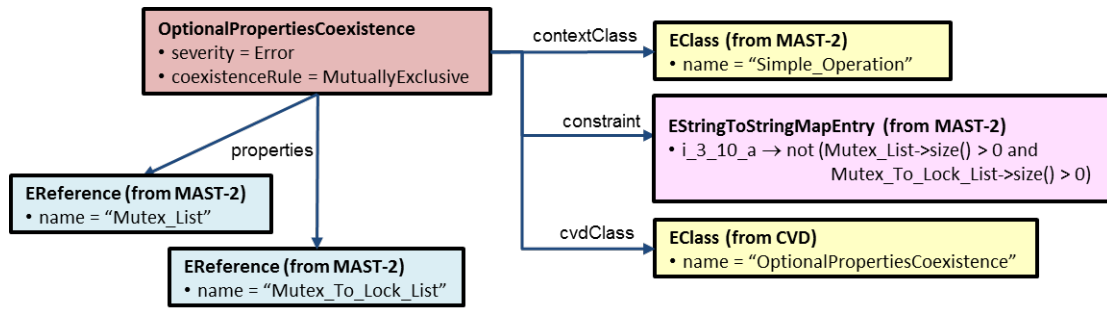


Figura 4.7

4.6 RelationalOrderForAttributesValue

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en, dado un subconjunto dentro de los atributos de una clase, imponer un determinado orden relacional entre los valores que adoptan los atributos pertenecientes a tal subconjunto.

Típicamente, los atributos sobre los que se especifica la restricción poseerán una semántica íntimamente relacionada.

Al igual que en el caso de `AttributeValueValidity`, el tipo de los atributos no ha de ser obligatoriamente numérico, sino que en general podría ser cualquier tipo para el que se haya establecido un orden bien definido, como por ejemplo 'char' o un tipo enumerado.

- **Estructura.**

Junto a las capacidades de modelado heredadas de `BasedOnPropertiesOfContextClass`, esta clase define un atributo `order` que representa el tipo de orden relacional impuesto {<, >, ≤, ≥}.

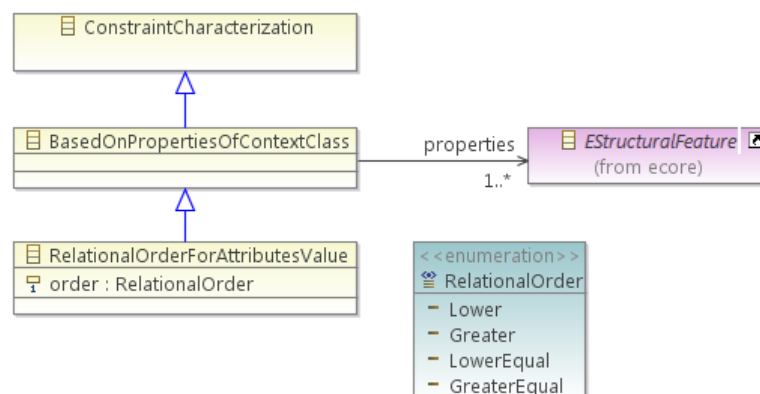


Figura 4.8

- **Restricciones de integridad.**

- A través de `properties` se han de referenciar `EAttribute`.
- Los `EAttribute` referenciados han de ser del mismo tipo.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`RelationalOrderForAttributesValue`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

Sobre el meta-modelo MAST-2 se encuentran definidas numerosas restricciones de integridad basadas únicamente en propiedades de la clase contexto, y que por tanto su caracterización podría ser perfectamente modelada mediante esta clase.

Algunos ejemplos son:

- `i_1_3_a`

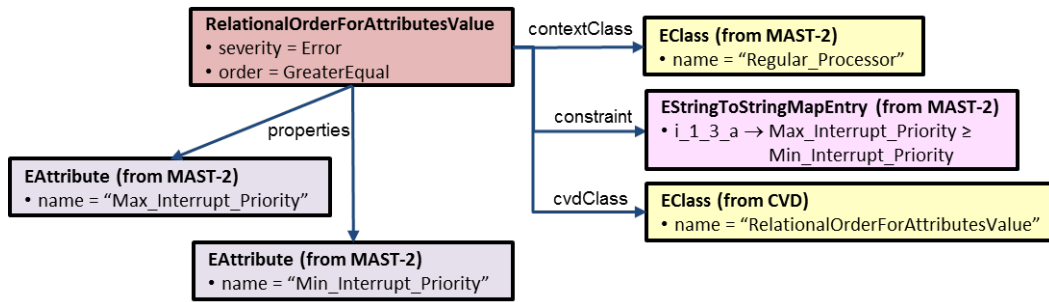


Figura 4.9

5 Subclases de ScopedBased

El meta-modelo CC define varias subclases de `ScopeBased` buscando proporcionar formas de modelar aún en mayor detalle la caracterización de restricciones dentro de la variedad correspondiente a aquellas cuya satisfacción depende no sólo del estado de un elemento de modelo, sino también del ámbito dentro del cual éste se halla. La Fig. 6.1 muestra tales subclases, que se exponen a continuación.

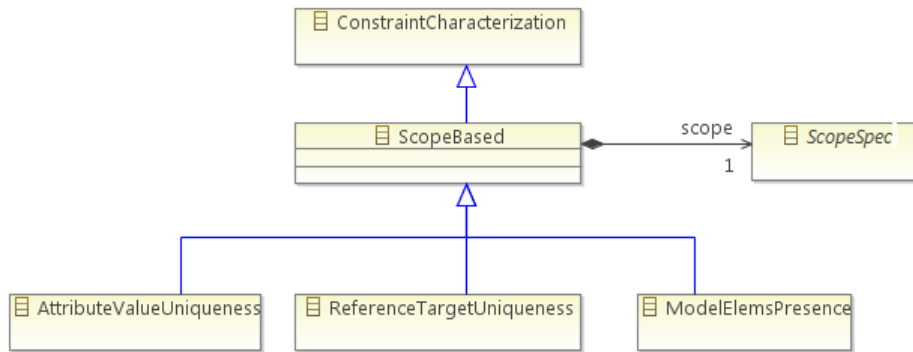


Figura 5.1

5.1 AttributeValueUniqueness

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en establecer que, en un ámbito o población de elementos de modelo, el valor de un atributo dado ha de ser único para cada elemento de modelo en ese ámbito.

- **Estructura.**

Junto a las capacidades de modelado heredadas de `ScopeBased`, esta clase define una referencia `attr` mediante la que especificar cuál es el atributo cuyo valor no puede hallarse repetido entre los elementos de modelo del ámbito.

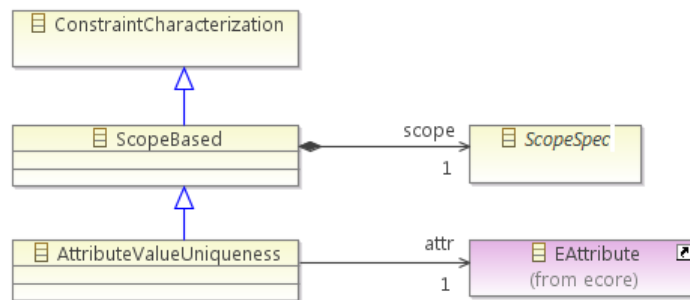


Figura 5.2

- **Restricciones de integridad.**

- El `EAttribute` especificado ha de pertenecer a la clase de los elementos de modelo que conforman el ámbito.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase CVD del mismo nombre (`AttributeValueUniqueness`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

- `i_2_2_a`

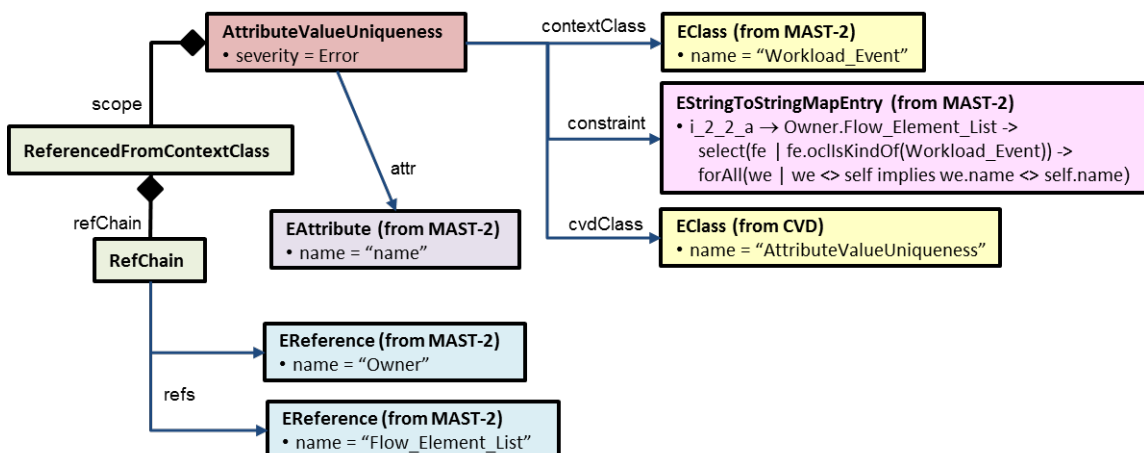


Figura 5.3

5.2 ReferenceTargetUniqueness

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en establecer que, en un ámbito o población de elementos de modelo, el *target* de una referencia dada ha de ser único para cada elemento de modelo en ese ámbito.

- **Estructura.**

Junto a las capacidades de modelado heredadas de `ScopeBased`, esta clase define una referencia `ref` mediante la que especificar cuál es la referencia cuyo *target* no puede hallarse repetido entre los elementos de modelo del ámbito.

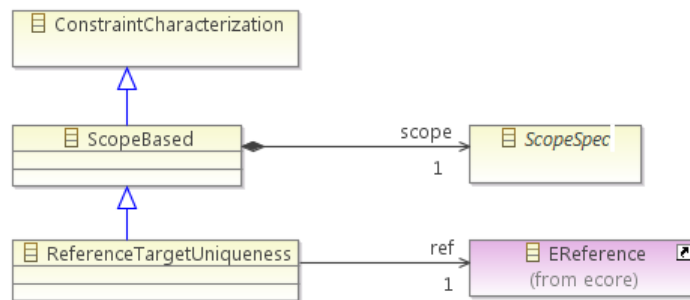


Figura 5.4

- **Restricciones de integridad.**

- La `EReference` especificada ha de pertenecer a la clase de los elementos de modelo que conforman el ámbito.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase `CVD` del mismo nombre (`ReferenceTargetUniqueness`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

5.3 ModelElemsPresence

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en acotar el número de elementos de modelo *de un determinado tipo* dentro de un ámbito.

La acotación puede significar establecer un número fijo y obligatorio de elementos o bien una cota superior / inferior.

- **Estructura.**

Junto a las capacidades de modelado heredadas de `ScopeBased`, esta clase define dos atributos `rule` y `ruleNum` que representan conjuntamente la regla de presencia impuesta.

El atributo `rule` indica si la restricción impuesta establece un número obligatorio de elementos de modelo, una cota inferior o una cota superior mientras que el atributo `ruleNum` especifica el valor de ese número o cota.

Además, la clase define una referencia `modelElemClass` mediante la que especificar el tipo de aquellos elementos de modelo dentro del ámbito sobre los que se establece la regla de presencia.

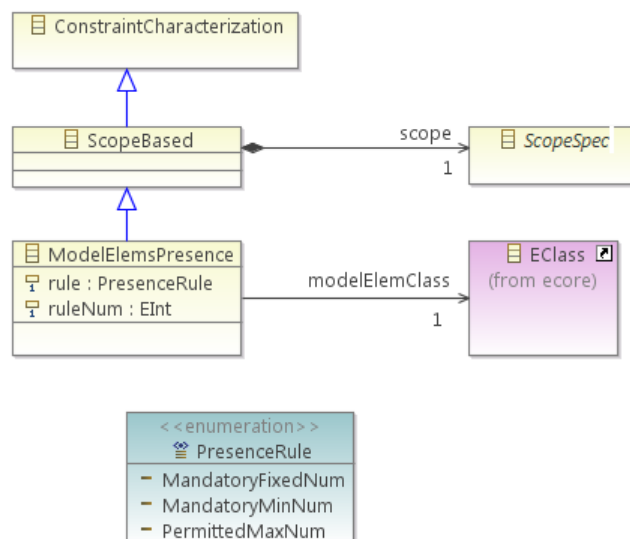


Figura 5.5

- **Restricciones de integridad.**

- La `EClass` referenciada por medio de `modelElemClass` ha de ser subclase del tipo de los elementos de modelo que conforman el ámbito especificado.

- **Relación con el meta-modelo CVD.**

Lo natural es que la referencia `cvdClass` quede establecida a la clase `CVD` del mismo nombre (`ModelElemsPresence`).

También sería válido establecerla a cualquiera de sus superclases.

- **Ejemplos en MAST-2.**

- `i_2_4_a`

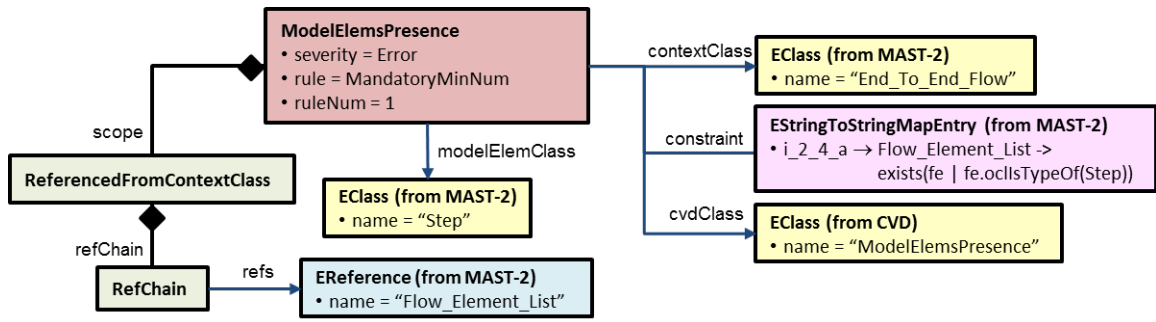


Figura 5.6

5.4 ModelElemsCoexistence

- **Naturaleza de las restricciones a ser caracterizadas.**

Esta clase es adecuada exclusivamente para modelar la caracterización de restricciones consistentes en

- **Estructura.**

Junto a las capacidades de modelado heredadas de `ScopeBased`, esta clase define un atributo `rule` que representa ... y una referencia `classes` que representa

- **Restricciones de integridad.**

-

- **Relación con el meta-modelo CVD.**

- **Ejemplos en MAST-2.**