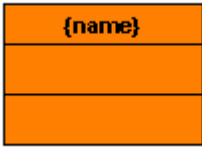
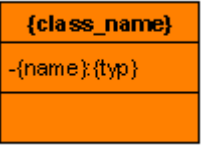
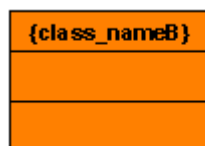
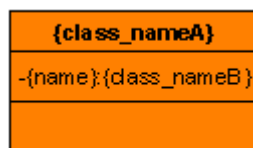
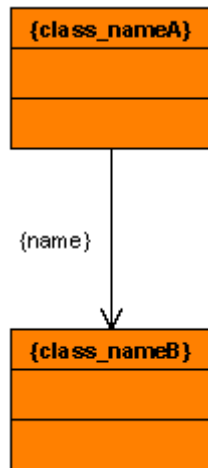


A Transformationsregeln für UML beschreibt OWL

Um mit der UML Ontologien in OWL zu modellieren, müssen die folgenden Transformationsregeln beachtet werden. In der Tabelle A.1 wird in der Spalte „OWL“ das zu modellierende OWL Konzept dargestellt, in Spalte „UML“ die grafische Umsetzung. In der Spalte „Besonderheiten/Erläuterung“ werden solche beschrieben. Ausdrücke in geschweiften Klammern, z.B. {name}, stellen einen Platzhalter dar. Der in UML angegebene Ausdruck in geschweiften Klammern erscheint bei einer Transformation in OWL an der entsprechenden Platzhalterstelle, die ebenfalls an den geschweiften Klammern zu erkennen ist.

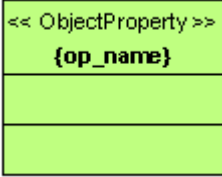
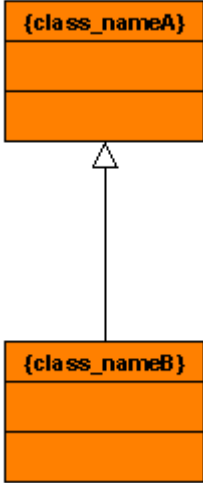
OWL	UML	Besonderheiten/Erläuterung
<pre>owl:Class mit rdf:ID="{name}"</pre>		/
<pre>owl:DatatypeProperty mit rdf:ID="{name}" zugeordnet zur Klasse {class_name} über owl:onProperty rdf:resource= "#{name}"</pre>		<p>Der Definitionsbereich ist eine Vereinigung aller Klassen, die das gleiche Attribut besitzen (Gleichheit von Namen und Typ), der Wertebereich wird anhand des angegebenen Typs {typ} ermittelt. Die unterstützten Datentypen in OWL sind in Tabelle A.2 aufgelistet.</p>

owl:ObjectProperty mit
 rdf:ID="{name}"
 zugeordnet zur Klasse
 {class_nameA} über
 owl:onProperty
 rdf:resource=
 "#{name}"



ObjectPropertys können durch gerichtete Assoziationen modelliert werden. Der Definitionsbereich ist eine Vereinigung aller Klassen, von denen eine gleich benannte Assoziation ausgeht. Der Wertebereich ist eine Vereinigung über alle Klassen, die am Assoziationsende einer solchen Assoziation stehen. Sollen der Defintions- und Wertebereich manuell festgelegt werden, so wird dies über Eigenschaftswerte erreicht. Siehe dazu Zeile „rdfs:domain und rdfs:range“.


Ebenfalls können sie durch Attribute mit einer Klasse als Typ modelliert werden. Der Definitionsbereich wird automatisch ermittelt und ist eine Vereinigung aller Klassen, die das gleiche Attribut (Name und Typ) aufweisen. Der Wertebereich ist die als Typ angegebene Klasse. Sollen der Defintions- und Wertebereich manuell festgelegt werden, so wird dies über Eigenschaftswerte erreicht. Siehe dazu Zeile „rdfs:domain und rdfs:range“.

<p>owl:ObjectProperty mit rdf:ID="{op_name}" ohne Zuordnung zu einer Klasse</p>		<p>Es ist auch möglich, über das Stereotyp «ObjectProperty» eine solche zu modellieren. Definitions- und Wertebereich müssen dabei über Eigenschaftswerte modelliert werden. Siehe dazu Zeile „rdf:domain und rdf:range“. Die Angabe von Eigenschaftswerten ist möglich.</p>
<p>rdfs:subClassOf</p>		<p>Über die Generalisierungsbeziehung wird die Klasse {class_nameB} als Unterklasse der Klasse {class_nameA} überführt.</p>
<p>rdfs:domain und rdf:range</p>	<p>Die Modellierung wird in der folgenden Grafik veranschaulicht. Sie zeigt einen Ausschnitt, wie Eigenschaftswerte in Poseidon modelliert werden.</p>	<p>Mit den Eigenschaftswerten „domain“ und „range“ können manuell der Definitions- und Wertebereich bei ObjectProperty festgelegt werden. „domain“ muss dabei stets als erster Eigenschaftswert stehen, „range“ als zweiter. Die beiden Eigenschaftswerte können entweder die Werte true, false oder beliebig viele Klassennamen getrennt durch einen Doppelpunkt annehmen. Der Wert true steht für die</p>

		<p>automatische Ermittlung, <code>false</code> für keine Ermittlung. Über die Klassennamen können die Bereiche manuell genau spezifiziert werden. <code>false</code> ist der default-Wert, wenn kein Eigenschaftswert angegeben wird.</p>
--	--	---

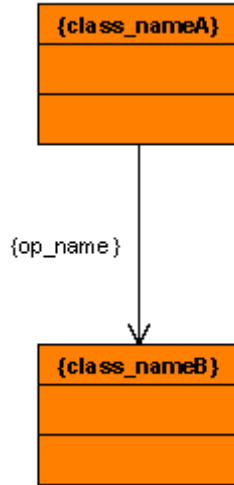
in	SourceCode	Einschränkungen	Eigenschaftswerte
Eigenschaft			Wert
domain			true
range			{class_nameA};{class_nameB}

<p>owl:inverseOf rdfs:subPropertyOf owl:equivalentProperty owl:equivalentClass owl:disjointWith owl:complementOf owl:transitiveProperty owl:symmetricProperty owl:functionalProperty owl:inverseFunctionalProperty</p>	<p>Die Modellierung des owl:inverseOf Konstruktes wird in der folgenden Grafik veranschaulicht. Es zeigt die Modellierung von Einschränkungen mit Poseidon.</p>	<p>Alle hier angegebenen Konzepte werden auf die gleiche Art und Weise über Einschränkungen modelliert. Der Name der Einschränkung ist gleich dem Namen des Konstruktes ohne das Präfix (owl:). Als Ausdruck muss die entsprechende Klasse oder Property angegeben werden. Bei den speziellen Propertys darf als Ausdruck nur „true“ oder „false“ stehen.</p>
---	---	---

Eigenschaften	Stil	Dokumentation	SourceCode	Einschränkungen	Eigenschaftswerte
					
Name	Sprache	Stereotypen	Ausdruck		
inverseOf			{inverse_ObjectProperty}		

owl:someValuesFrom
 owl:allValuesFrom
 owl:hasValue

und Zuordnung der
 ObjectProperty {op_name} zu
 der Klasse {class_nameA}, mit
 der Restriktion, dass nur Werte
 von {class_nameB}
 vorkommen dürfen, wenn – wie
 hier im Beispiel dargestellt –
 „allValuesFrom“ als
 Einschränkung modelliert
 wurde.



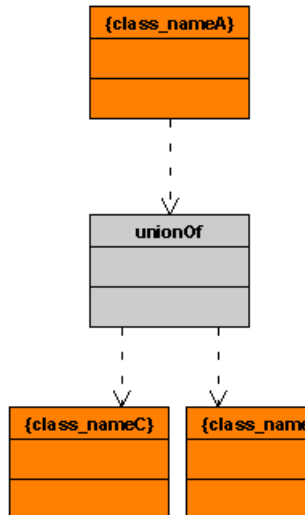
Die Modellierung der
 Einschränkung ist in
 der Grafik unten
 dargestellt.

Diese OWL Konzepte
 werden über
 Einschränkungen
 modelliert. Die
 Einschränkung wird für die
 ObjectProperty {op_name}
 definiert und muss den
 Namen des Konzeptes ohne
 das Präfix (owl :) erhalten.
 Der Ausdruck ist Name der
 Klasse, für die das Konzept
 gelten soll, also in der Regel
 die Klasse, auf die die
 ObjectProperty zeigt.

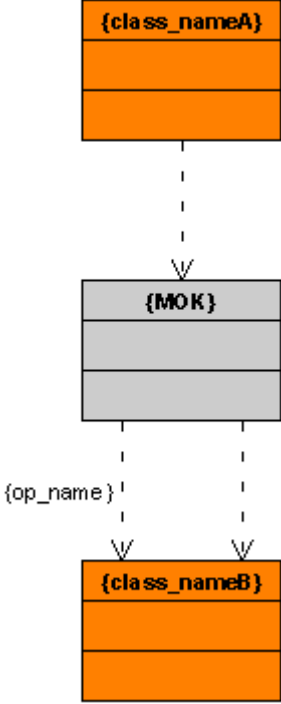
Für komplexe
 Mengenoperationen können
 diese Konzepte auch
 verwendet werden. Siehe
 dazu Zeile „komplexe
 Mengenoperationen“.

Eigenschaften		Stil	Dokumentation	SourceCode	Einschränkungen	Eigenschaftswerte
Name	Sprache	Stereotypen	Ausdruck			
allValuesFrom			{class_nameB}			

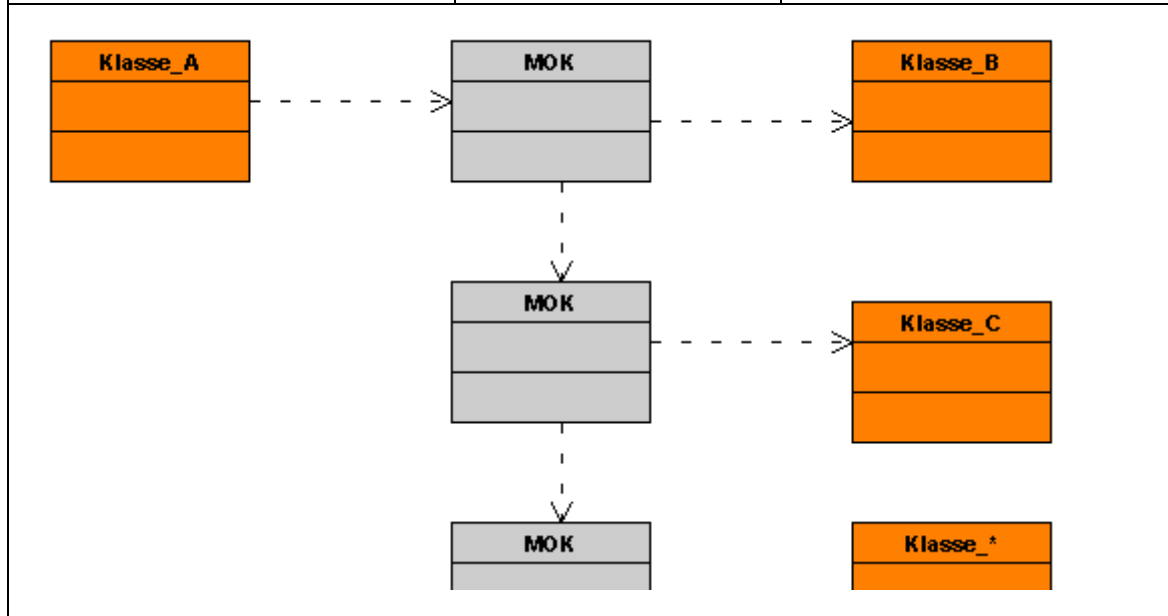
owl:unionOf
 owl:intersectionOf



Die beiden Konzepte für
 Mengenoperationen werden
 über eine eigene Klasse mit
 dem Namen der
 Mengenoperation
 modelliert. Zwischen dieser
 Mengenoperationsklasse
 und den beteiligten Klassen
 dürfen nur
 Abhängigkeitsbeziehungen
 verwendet werden.

<p>Mengenoperationen mit anonymen Klassen</p> <p>Vereinigung oder Schnittmenge zwischen:</p> <pre>owl:Class rdf:about="#{class_nameB}"</pre> <p>und einer anonymen Klasse:</p> <pre><owl:Restriction> <owl:onProperty rdf:resource= "#{op_name}"/> </owl:Restriction></pre>		<p>Eine Vereinigung oder eine Schnittmenge zwischen einer Klasse und einer anonymen Klasse, der eine bestimmte Property zugewiesen ist, wird nicht explizit modelliert. Es wird mit der Modellierung einer gerichteten, gestrichelten Kante, die den Namen der Property {op_name} trägt, eine solche anonyme Klasse erzeugt. {MOK} steht für Mengenoperationsklasse und dient als Platzhalter für „unionOf“ oder „intersectionOf“.</p> <p>Propertyrestriktionen können, wie in der Zeile „Komplexe Mengenoperationen“ beschrieben, hinzugefügt werden.</p>
<p>Komplexe Mengenoperationen</p>	<p>Die grundsätzliche Modellierung von komplexen Mengen wird in der folgenden Grafik veranschaulicht. Es können beliebig viele geschachtelte Mengenoperationen vorgenommen werden.</p>	<p>Sollen Propertyrestriktionen bei komplexen Mengenoperationen genutzt werden, dann werden diese als Einschränkung an die modellierte Abhängigkeit geschrieben. Der Name der Einschränkung entspricht dem OWL Konstrukt ohne das Präfix (owl :), der Ausdruck muss den Namen der Klasse enthalten, auf die die Abhängigkeit zeigt. Ist dies eine Mengenoperationsklasse</p>

(MOK), so muss der Ausdruck den Namen dieser Klasse enthalten, also „unionOf“ oder „intersectionOf“. Zyklen sind verboten.



Instanzen

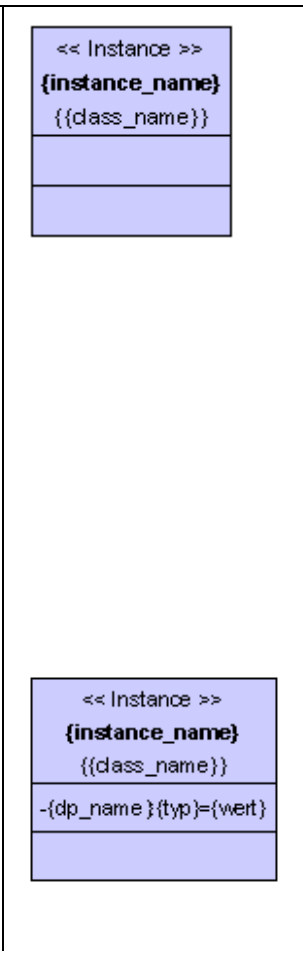
```
<owl:Thing rdf:about=
"#{instance_name}">

<rdf:type rdf:resource=
"#{class_name}"/>

</owl:Thing>
```

- einer DatatypeProperty

```
<{dp_name}
rdf:datatype="http://www.w3
.org/2001/XMLSchema#{typ}">
{wert}
```



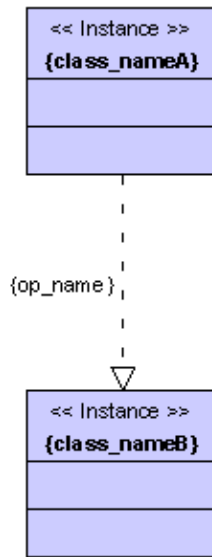
Bei einer Instanz muss die Einschränkung „instanceOf“ angegeben werden, die als Ausdruck den Klassennamen {class_name} besitzt von der die Instanz eine Ausprägung ist. Es wird empfohlen, die „instanceOf“ Einschränkung stets als erste Einschränkung zu modellieren.

Die Ausprägung einer DatatypeProperty und damit die Zuweisung einer solchen Property zu einer Instanz wird über die Modellierung eines

```
</{dp_name}>
```

- einer ObjectProperty

```
<owl:Thing
rdf:about="#{class_nameA}">
<{op_name} rdf:resource=
"#{class_nameB}"/>
</owl:Thing>
```



Attributes und das Setzen eines Startwertes erreicht.

Zwischen zwei Instanzen kann über das UML Element Realisierung eine Instanz einer ObjectProperty modelliert werden.

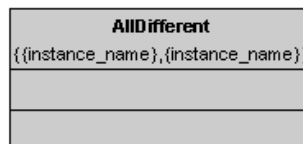
```
owl:sameAs
owl:differentFrom
```

In der folgenden Grafik wird die Verwendung von owl:differentFrom durch die Modellierung einer entsprechenden Einschränkung in Poseidon dargestellt.

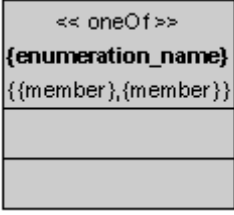
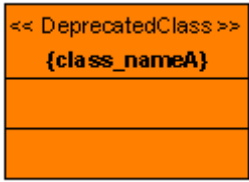
Die beiden Konstrukte können über Einschränkungen modelliert werden. Der Name der Einschränkung muss dabei dem Konstruktnamen ohne das Präfix (owl:) entsprechen, als Ausdruck muss ein Instanzname {instance_name} angegeben werden.

Eigenschaften	Stil	Dokumentation	SourceCode	Einschränkungen	Eigenschaftswerte
✖					
Name	Sprache	Stereotypen	Ausdruck		
differentFrom			{instance_name}		

```
owl:AllDifferent
```



Die Klasse mit dem Namen „AllDifferent“ ermöglicht die Modellierung des OWL Konzeptes. Als Einschränkungen müssen alle sich jeweils unterscheidenden

		<p>Instanzen angegeben werden. Als Name der Einschränkung muss der Klassenname {class_name}, als Ausdruck der Instanzname {instance_name} verwendet werden.</p>
<p>owl:oneOf</p>	 <pre> classDiagram class EnumerationName["{enumeration_name}"] { <<oneOf>> {member} {member} } </pre>	<p>Das Stereotyp «oneOf» – verwendet bei einer Klasse – ermöglicht die Modellierung einer Enumeration mit dem Namen {enumeration_name}. Als Einschränkungsname kann entweder der spezielle definierte Typ {enumeration_name} oder die allgemeine Klasse Thing verwendet werden. Als Ausdruck werden die Mitglieder der Enumeration angegeben {member}.</p>
<p>owl:DeprecatedClass owl:DeprecatedProperty</p>	 <pre> classDiagram class ClassNameA["{class_nameA}"] { <<DeprecatedClass>> } </pre>	<p>Beide Konzepte können über ein eigenes Stereotyp, entweder «DeprecatedClass» oder «DeprecatedProperty», modelliert werden. «DeprecatedProperty» kann nicht bei ObjectProperty verwendet werden, wenn diese durch eine Klasse mit dem Stereotyp «ObjectProperty»</p>

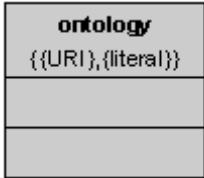
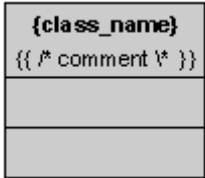
		modelliert wurde. In der Grafik ist zur Verdeutlichung nur die Verwendung von «DeprecatedClass» abgebildet.
<pre>owl:priorVersion owl:incompatibleWith owl:backwardCompatibleWith owl:versionInfo</pre>		Für die Versionierung kann eine eigene Klasse mit dem Namen „ontology“ modelliert werden. Die Versionierungskonstrukte werden als Einschränkungen modelliert. Als Ausdruck muss ein URI bzw. bei „versionInfo“ ein Literal angegeben werden.
<pre>rdfs:comment</pre>		Kommentare werden über eine Einschränkung mit dem Namen „comment“ modelliert. Als Ausdruck steht der Kommentar. Es wird empfohlen, Kommentare zusätzlich zwischen Schrägstrich und Sternchen („/* - /*“) zu schreiben.

Tabelle A.1 – Transformationsregeln für UML beschreibt OWL

Im der Tabelle A.2 sind alle definierten Stereotype aufgelistet, die bei einer Modellierung von Ontologien mit der UML verwendet werden können. In der Spalte „Semantik“ ist die Bedeutung, und wie dieses Stereotyp bei einer Transformation beachtet wird, erläutert.

Stereotyp aus dem UfO Profil	Anwendbar auf	Semantik
ObjectProperty	Klassenelement	Das Klassenelement wird dann in eine owl:ObjectProperty transformiert.
Instance	Klassenelement	Kann bei einer modellierten Klasse verwendet werden, um eine Instanz zu modellieren. Das Klassenelement wird dann in eine Instanz überführt.
oneOf	Klassenelement	Kann bei einer modellierten Klasse verwendet werden, um eine Enumeration zu modellieren. Das Klassenelement wird dann – mit entsprechend modellierten Einschränkungen – in eine Enumeration transformiert.
DeprecatedClass	Klassenelement	Kann bei einer modellierten Klasse verwendet werden, um die Klasse mit dieser Eigenschaft zu belegen. Bei einer Transformation wird das OWL Konstrukt der Klasse hinzugefügt.
DeprecatedProperty	Assoziationselement Attributelement	Kann bei einer modellierten ObjectProperty oder DatatypeProperty verwendet werden, um die Property mit dieser Eigenschaft zu belegen. Bei einer Transformation wird das OWL Konstrukt der Property hinzugefügt.

Tabelle A.2 – Erlaubte Stereotype bei der Modellierung von OWL Ontologien mit der UML

In der Tabelle A.3 werden alle möglichen Einschränkungen aufgelistet. Ebenso stehen in der Tabelle ihre Semantik, sowie die UML Elemente, bei denen die Einschränkungen gemacht werden dürfen.

Einschränkung	Anwendbar auf	Semantik
inverseOf	Assoziationselement (Attributelement) Klassenelement mit Stereotyp «ObjectProperty»	Beschreibt die inverse einer ObjectProperty. Anwendbar auf ein Attributelement, wenn der Wert ein Datentyp ist und eine OWL Full Ontologie entwickelt werden soll.
disjointWith	Klassenelement	Beschreibt die Disjunktion zwischen zwei Klassen.
complementOf	Klassenelement	Beschreibt das Komplement einer Klasse.
subPropertyOf	Assoziationselement Attributelement Klassenelement mit Stereotyp «ObjectProperty»	Definiert die Property als Unterproperty.
TransitiveProperty	Assoziationselement Klassenelement mit Stereotyp «ObjectProperty»	Beschreibt eine transitive ObjectProperty.
SymmetricProperty	Assoziationselement Klassenelement mit Stereotyp «ObjectProperty»	Beschreibt eine symmetrische ObjectProperty.
FunctionalProperty	Assoziationselement Klassenelement mit Stereotyp «ObjectProperty»	Beschreibt eine funktionale ObjectProperty.

InverseFunctionalProperty	Assoziations- element Klassenelement mit Stereotyp «ObjectProperty»	Beschreibt eine inverse funktionale ObjectProperty.
equivalentClass	Klassenelement	Beschreibt die Äquivalenz zwischen zwei Klassen.
equivalentProperty	Assoziations- element Klassenelement mit Stereotyp «ObjectProperty»	Beschreibt die Äquivalenz zwischen zwei ObjectProperty.
allValuesFrom someValuesFrom hasValue	Assoziations- element Abhängigkeitselement	Beschreiben eine Propertyrestriktion. Können im Zusammenhang mit komplexen Mengenoperationen auch für ein Abhängigkeitselement angewandt werden.
instanceOf	Klassenelement mit Stereotyp «Instance»	Beschreibt die Zugehörigkeit einer Instanz zu einer Klasse. Sollte als erste Einschränkung einer Instanz modelliert werden.
sameAs differentFrom	Klassenelement mit Stereotyp «Instance»	Beschreibt die Gleichheit bzw. Unterschiedlichkeit zweier Instanzen.
Thing	Klassenelement mit Stereotyp «oneOf»	Wird bei einer Enumeration verwendet, um die Mitglieder einer Aufzählung vom Typ Thing zu definieren.
{spezifischer Typ}	Klassenelement mit Stereotyp «oneOf»	Wird bei einer Enumeration verwendet, um die Mitglieder einer Aufzählung von diesem spezifischen Typ zu definieren. Der Name des spezifischen

		Typs ist der Name der Enumeration.
priorVersion backwardCompatibleWith incompatibleWith imports versionInfo	Klassenelement mit Stereotyp «ontology»	Beschreiben Versionsinformationen für die Ontologie. Als Ausdruck müssen URIs angegeben werden, mit Ausnahme bei „versionInfo“. Dort muss ein Literal angegeben werden.
comment	Klassenelement Assoziationselement Attributelement	Dient dem Hinzufügen eines Kommentares. Der Ausdruck ist der Kommentar.

Tabelle A.3 – Übersicht aller möglichen Einschränkungen mit dem UfO Profil

In der folgenden Tabelle A.4 stehen die möglichen Eigenschaftswerte, die bei der Modellierung verwendet werden dürfen. Ebenso stehen in der Tabelle ihre Semantik und die möglichen UML Elemente, bei denen die Eigenschaftswerte angewendet werden dürfen.

Eigenschaftswert	Anwendbar auf	Semantik
domain	Assoziationselement	Schränkt den Definitionsbereich ein. Mit dem Wert <code>true</code> wird dieser automatisch aus dem Diagramm ermittelt. Mit dem Wert <code>false</code> (gleichzeitig default-Wert) wird kein Definitionsbereich erzeugt. Mit der Angabe von Klassennamen, getrennt durch einen Doppelpunkt, ist die Angabe eines spezifischen Definitionsbereichs möglich. Der Eigenschaftswert „domain“ muss an erster Stelle stehen.
range	Assoziationselement	Schränkt den Wertebereich ein. Mit dem Wert <code>true</code> wird dieser automatisch aus dem Diagramm ermittelt. Mit dem Wert <code>false</code> (gleichzeitig default-Wert) wird kein Wertebereich erzeugt. Mit der Angabe von Klassennamen, getrennt durch einen Doppelpunkt, ist die Angabe eines spezifischen Wertebereichs möglich. Der Eigenschaftswert „range“ muss an zweiter Stelle stehen.

Tabelle A.4 – Übersicht der möglichen Eigenschaftswerte mit dem UfO Profil

(c) copyright by Dipl.-Wirt.-Inf. Sebastian Leinhos

<http://diplom.ooyoo.de>