

# Specificatie attributen gegenereerd voor associaties

Versie	Datum	Auteur	Aanpassingen
1.0	6 sep. 19	Martin Vanbrabant	-
1.1	12 sep. 19	Martin Vanbrabant	Extra gevallen Gestructureerde indeling
1.2.alfa1	27 sep. 19	Martin Vanbrabant	<b>Aanpassingen na meeting 2019-09-25:</b> Herformulering inleiding Extra hoofdstuk Algemene principes Extra specificatie: name(att) Extra specificatie: ap-label-nl(att) Extra specificatie: definition-nl(att); bij attributen in associatieklassen Extra specificatie: uri(att); gevallen met uri op betrokken associatie/rol/klasse Aanpassingen voor cross-package gevallen Aanpassingen voor afdalingen i.c.m. associatieklassen Bevat nog TODO regels – vandaar alfa1
1.2.alfa2	2 okt. 19	Martin Vanbrabant	<b>Aanpassingen:</b> Inleiding: toelichting bij de voorbeelden Algemene principes: toevoeging paragraaf Ongeoorloofd uitbreiden van extern vocabulary Correctie formules die gebruik maken van name-source-class en name-target-class Toevoeging hulp-tags uri-source-attribute en uri-target-attribute aan associatieklassen Aanpassingen voor cross-package gevallen uit vorige versie teniet gedaan (zie nu nieuwe paragraaf Ongeoorloofd uitbreiden van extern vocabulary) Wegwerken TODO regels Testresultaten aangeduid (groen=correct; rood=fout of nog te implementeren) Bevat nog rood – vandaar alfa2

## Inleiding

Dit document specificeert te genereren attributen in de output van de OSLO toolchain, in de gevallen waar in de UML associaties voorkomen, al dan niet met bijhorende associatieklassen.

De verschillende gevallen worden onderscheiden en per geval wordt gedefinieerd welke attributen worden gegenereerd en welk eigenschappen deze attributen krijgen.

Om het proza te beperken, worden de eigenschappen in de vorm van 'formules' gespecificeerd. Deze formules zijn een soort pseudo code, waarvan de syntax of notaties hieronder wordt uitgelegd.

Bij dit document hoort een input UML specificatie, in de vorm van enkele Enterprise Architect .EAP bestanden, die als testinput gebruikt worden. De diagrammen en de voorbeelden die onder elke formule worden gegeven zijn ontleend aan deze .EAP bestanden.

## Syntax in de formules

+	string concatenatie operator
UMLname (elem)	naam gegeven in UML aan elem. Niet te verwarren met de name tag!
effectiveName (elem)	waarde van de name tag van elem of indien deze niet bestaat, UMLname (elem).
baseURI (elem)	de base URI toegekend aan het package van elem.
xyz (elem)	waarde van tag xyz voor element aangeduid met elem.
tolower1 (tekst)	tekst, maar met eerste karakter omgezet naar kleine letter.
disamb (a, b)	indien nodig omdat de associatie een self-associatie is, voeg ter disambiguatie suffix a toe aan source kant, suffix b aan target kant.

In bovenstaande is elem een van:

- assoc: associatie
- att: attribuut (gegenereerd attribuut in kwestie)
- att\_source: attribuut in source klasse of attribuut in associatieklasse, betreffende source klasse
- att\_target: attribuut in target klasse of attribuut in associatieklasse, betreffende target klasse
- class: klasse (deelnemende klasse die het betreffende attribuut bevat)
- class\_assoc: associatieklasse (die het betreffende attribuut bevat)
- class\_source: klasse, aan source kant van associatie
- class\_target: klasse, aan target kant van associatie
- rol: rol geplaatst op uiteinde van associatie

Een voorbeeld:

```
label-nl(att) = tolower1(label-nl(assoc))
```

*"De label-nl tag van het attribuut is gelijk aan de label-nl tag van de associatie, waarvan de eerste letter werd omgezet naar kleine letter."*

## Over de voorbeelden

De voorbeelddiagrammen tonen van elk aanwezig element slechts de UML naam ervan (zie ook de functie UMLname (elem) hierboven). Om eenvoudig voorbeelden te kunnen maken van andere eigenschappen van een aanwezig element, werden volgende afspraken gehanteerd. Deze zijn ook geïmplementeerd in de voorbeeld .EAP bestanden.

Tags:

- De waarde van de altijd aanwezige tag `label-nl` is aangemaakt op basis van de UML naam, maar bevat ter illustratie minstens één spatie. De eerste letter heeft hetzelfde voorkomen (kleine letter of grote letter) als de UML naam. Ter verduidelijking staan enkelvoudige aanhalingstekens voor en na de waarde.
- De waarde van de optionele tag `ap-label-nl` is gelijk aan de waarde van de tag `label-nl`, gevolgd door ' AP'.
- De waarde van de optionele tag `name` is gelijk aan de UML naam gevolgd door '\_naam'.
- De waarde van de optionele tag `uri` is gelijk aan de 'https://extern.domein#' gevolgd door de UML naam, waarvan in het geval van associaties de eerste letter werd omgevormd naar kleine letter.

Voorbeeld: associatieklasse Heeft06

UML naam	Heeft06
label-nl	'Heeft 06'
ap-label-nl	'Heeft 06 AP'
name	Heeft06_naam
uri	'https://extern.domein#Heeft06'

Voorbeeld: associatie Heeft06 (waarden gelijk aan deze hierboven, behalve):

uri	'https://extern.domein#heeft06'
-----	---------------------------------

Packages:

- Tenzij anders vermeld, bevinden alle elementen zich in een package, of is er een package tag aan toegekend. In beide gevallen heeft het package als baseURI `https://data.vlaanderen.be/ns/mijndomein#`.
- Een optioneel gebruikt ander package heeft als baseURI `https://data.vlaanderen.be/ns/mijnanderdomein#`.

## Algemene principes

### URI conflicten

In sommige gevallen kan voor meerdere gegenereerde attributen dezelfde URI worden gegenereerd. Een voorbeeld is geval 2, indien de associatie een tag `uri` heeft.

Dit kan aanleiding geven tot een URI conflict. Een URI conflict is het meer dan éénmaal een URI gebruiken, echter met verschillend domein en/of bereik.

In de gevallen waar de toolchain meer dan één keer eenzelfde URI genereert, zal ze een waarschuwing (warning) geven. Waarom geen foutmelding (error)? Omdat dit soms ook niet tot een URI conflict leidt. Dit leidt bijvoorbeeld niet tot een URI conflict in een applicatieprofiel, als het om een afgedaalde associatie zou gaan, die in het vocabularium zou gedefinieerd zijn op een gemeenschappelijke basisklasse van de deelnemende klassen. In geval dit tot een daadwerkelijke URI conflict leidt, kan de gebruiker als alternatief een associatie met rollen gebruiken.

### Niet gedefinieerde URI's in een applicatieprofiel

Indien men in een applicatieprofiel een associatie verlegt ("afdaalt") van een basisklasse naar een afgeleide, kan voor het betreffende attribuut in de deelnemende klasse een URI worden gegenereerd, die niet gedefinieerd werd in het vocabularium. Om dit te vermijden kan de gebruiker als alternatief een associatie met rollen gebruiken. Ook voor het linkend attribuut in de eventuele associatieklasse kan dit voorkomen. Om dit te vermijden kan de gebruiker de speciaal daarvoor voorziene tags `name-source-class` en `name-target-class` toekennen aan de associatieklasse. Deze worden beschreven verder in dit document.

### Ongeoorloofd uitbreiden van extern vocabularium

Het genereren van nieuwe attributen in klassen die deelnemen aan een associatie of in associatieklassen kan aanleiding geven tot het uitbreiden van een extern vocabularium, wat niet toegelaten is (zie ook modelleringsregels). Deze situatie is niet te herkennen door de toolchain. Er is telkens een alternatief voorzien, waarbij de gebruiker zelf een uri kan toekennen aan het betreffende te genereren attribuut.

### Kopiëren van tags

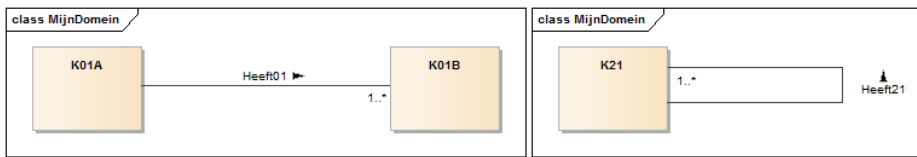
Attributen gegenereerd in deelnemende klassen krijgen naast de in dit document expliciet gedefinieerde tags ook een kopie van elke tag die toegekend was aan de associatie of de rol waarvoor ze werden gegenereerd, behalve de tags die expliciet worden vermeld in de formules.

Attributen gegenereerd in associatieklassen krijgen enkel de in dit document expliciet gedefinieerde tags.

## Geval 1: zonder associatieklasse, met richtingspijl, zonder rollen

*Dit is de historische situatie die verder ondersteund wordt omwille van achterwaartse compatibiliteit.  
Voor nieuwe modellen: voorkeur geven aan oplossingen zonder richtingspijl.*

### UML voorbeeld



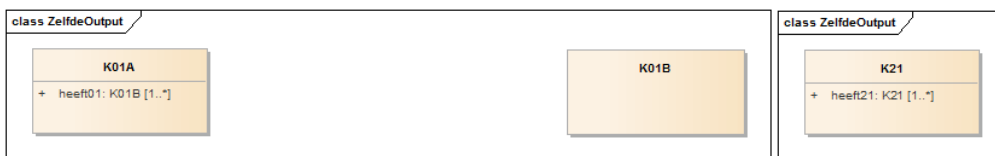
### Kenmerken

- Naam van de associatie: aanwezig
- Richtingspijl bij de naam van de associatie: aanwezig
- Rollen: afwezig

### Principes voor gegenereerde attributen

- Er wordt een attribuut gegenereerd in de klasse aan de kant van de associatie waaruit de richtingspijl vertrekt.

### UML met zelfde output als het voorbeeld



### Attributen in klassen

#### Formule 1.1.1

```
label-nl( att ) = tolower1( label-nl( assoc ) )
```

#### Voorbeelden

```
'heeft 01'  
'heeft 21'
```

#### Formule 1.1.2

*Enkel als assoc een tag ap-label-nl heeft.*

```
ap-label-nl( att ) = tolower1( ap-label-nl( assoc ) )
```

#### Voorbeelden

```
'heeft 01 AP'  
'heeft 21 AP'
```

#### Formule 1.1.3

(Input voor bepaling uri van dit attribuut).

```
name( att ) = tolower1( effectiveName( assoc ) )
```

#### Voorbeelden

```
heeft01, heeft01_naam  
heeft21, heeft21_naam
```

#### Formule 1.1.4

*Enkel als assoc een tag uri heeft.*

```
uri( att ) = uri( assoc )
```

#### Voorbeelden

```
https://extern.domein#heeft01
```

```
https://extern.domein#heeft21
```

#### Formule 1.1.5

***Enkel als*** assoc ***geen tag*** uri ***heeft.***

```
uri( att ) = baseURI( class ) + name( att )
```

#### Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#heeft01
```

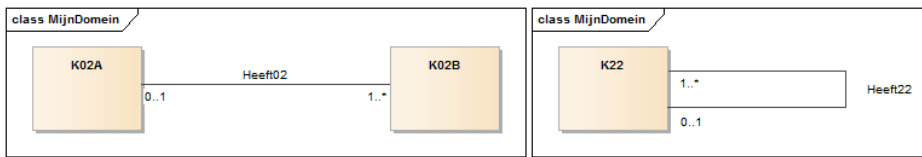
```
https://data.vlaanderen.be/ns/mijndomein#heeft01_naam
```

```
https://data.vlaanderen.be/ns/mijndomein#heeft21
```

```
https://data.vlaanderen.be/ns/mijndomein#heeft21_naam
```

## Geval 2: zonder associatieklasse, zonder richtingspijl, zonder rollen

### UML voorbeeld



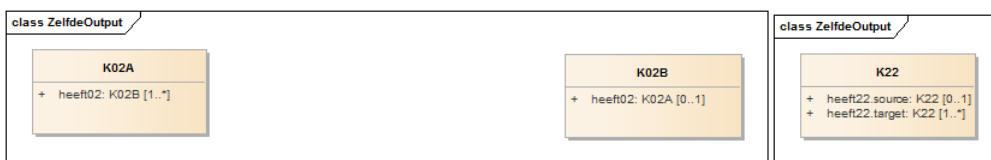
### Kenmerken

- Naam van de associatie: aanwezig
- Richtingspijl bij de naam van de associatie: afwezig
- Rollen: afwezig

### Principes voor gegenereerde attributen

- Er wordt een attribuut gegenereerd in beide betrokken klassen.
- Waar nodig, worden aan label-nl en URI een suffix toegevoegd ter disambiguatie.

### UML met zelfde output als het voorbeeld



### Attributen in klassen

#### Formule 2.1.1

```
label-nl(att) = tolower1(label-nl(assoc)) + disamb(' (source)', ' (target)')
```

##### Voorbeelden

```
'heeft 02'  
'heeft 22 (source)', 'heeft 22 (target)'
```

#### Formule 2.1.2

**Enkel als** assoc **een tag** ap-label-nl **heeft**.

```
label-nl(att) = tolower1(ap-label-nl(assoc)) + disamb(' (source)', ' (target)')
```

##### Voorbeelden

```
'heeft 02 AP'  
'heeft 22 AP (source)', 'heeft 22 AP (target)'
```

#### Formule 2.1.3

(Input voor bepaling uri van dit attribuut).

```
name(att) = effectiveName(class) + '.' + tolower1(effectiveName(assoc)) +  
disamb('.source', '.target')
```

##### Voorbeelden

```
K02A.heeft02, K02A_naam.heeft02_naam  
K02B.heeft02, K02B_naam.heeft02_naam  
K22.heeft22.source, K22_naam.heeft22_naam.source [.source ontbreekt]  
K22.heeft22.target, K22_naam.heeft22_naam.target [.target ontbreekt]
```

#### Formule 2.1.4

**Enkel als** assoc **een tag** uri **heeft**.

```
uri(att) = uri(assoc)
```

## Voorbeelden

```
https://extern.domein#heeft02  
https://extern.domein#heeft22
```

## Formule 2.1.5

**Enkel als** assoc **geen tag** uri **heeft**.

<code>uri(att) = baseURI(class) + name(att)</code>
--

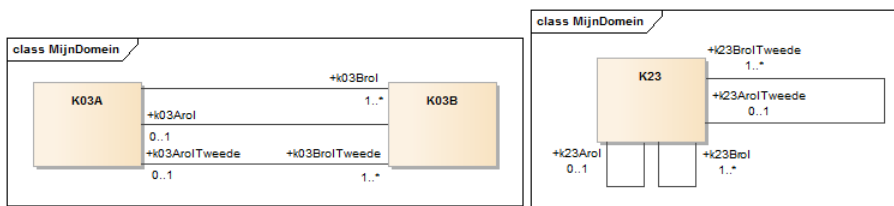
## Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#K02A.heeft02  
https://data.vlaanderen.be/ns/mijndomein#K02A_naam.heeft02_naam  
https://data.vlaanderen.be/ns/mijndomein#K02B.heeft02  
https://data.vlaanderen.be/ns/mijndomein#K02B_naam.heeft02_naam  
https://data.vlaanderen.be/ns/mijndomein#K22.heeft22.source [.source ontbreekt]  
https://data.vlaanderen.be/ns/mijndomein#K22_naam.heeft22_naam.source [idem]  
https://data.vlaanderen.be/ns/mijndomein#K22.heeft22.target [.target ontbreekt]  
https://data.vlaanderen.be/ns/mijndomein#K22_naam.heeft22_naam.target [idem]
```



## Geval 3: zonder associatieklasse, zonder richtingspijl, met rollen

### UML voorbeeld



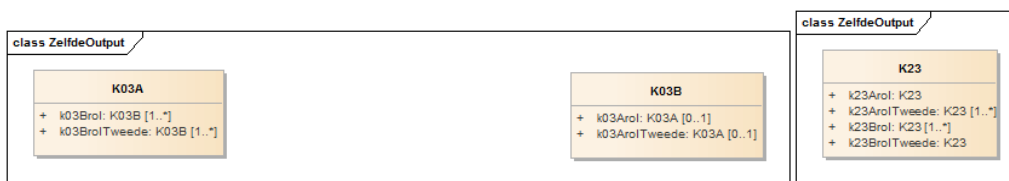
### Kenmerken

- Naam van de associatie: afwezig
- Richtingspijl bij de naam van de associatie: afwezig
- Rollen: aanwezig

### Principes voor gegenereerde attributen

- Enkel als aan een uiteinde van een associatie een rol is toegekend, wordt in de klasse aan het andere uiteinde van de associatie een attribuut gegenereerd.

### UML met zelfde output als het voorbeeld



### Attributen in klassen

#### Formule 3.1.1

$\text{label-nl}(\text{att}) = \text{label-nl}(\text{rol})$

#### Voorbeelden

```
'k 03A rol', 'k 03A rol Tweede'
'k 03B rol', 'k 03B rol Tweede'
'k 23A rol', 'k 23A rol Tweede' [leeg]
'k 23B rol', 'k 23B rol Tweede' [leeg]
```

#### Formule 3.1.2

*Enkel als rol een tag ap-label-nl heeft.*

$\text{ap-label-nl}(\text{att}) = \text{ap-label-nl}(\text{rol})$

#### Voorbeelden

```
'k 03A rol AP', 'k 03A rol Tweede AP'
'k 03B rol AP', 'k 03B rol Tweede AP'
'k 23A rol AP', 'k 23A rol Tweede AP' [leeg]
'k 23B rol AP', 'k 23B rol Tweede AP' [leeg]
```

#### Formule 3.1.3

(Input voor bepaling uri van dit attribuut).

$\text{name}(\text{att}) = \text{effectiveName}(\text{rol})$

#### Voorbeelden

```
k03Arol, k03Arol_naam, k03ArolTweede, k03ArolTweede_naam
k03Brol, k03Brol_naam, k03BrolTweede, k03BrolTweede_naam
```

k23Arol, k23Arol\_naam, k23ArolTweede, k23ArolTweede\_naam  
k23Brol, k23Brol\_naam, k23BrolTweede, k23BrolTweede\_naam

### Formule 3.1.4

*Enkel als rol een tag uri heeft.*

<code>uri(att) = uri(rol)</code>
----------------------------------

#### Voorbeelden

`https://extern.domein#k03Arol`  
`https://extern.domein#k03ArolTweede`  
`https://extern.domein#k03Brol`  
`https://extern.domein#k03BrolTweede`  
`https://extern.domein#k23Arol [uri tag niet gelezen]`  
`https://extern.domein#k23ArolTweede [uri tag niet gelezen]`  
`https://extern.domein#k23Brol [uri tag niet gelezen]`  
`https://extern.domein#k23BrolTweede [uri tag niet gelezen]`

### Formule 3.1.5

*Enkel als rol geen tag uri heeft.*

<code>uri(att) = baseURI(class) + name(att)</code>
--

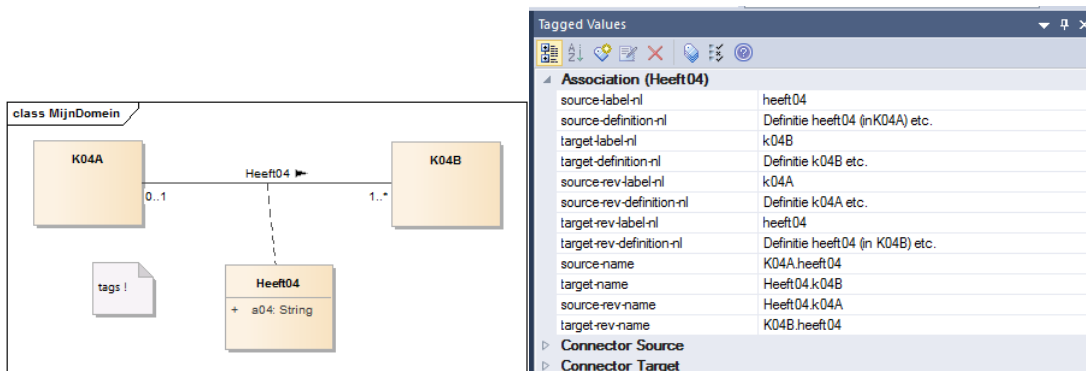
#### Voorbeelden

`https://data.vlaanderen.be/ns/mijndomein#k03Arol`  
`https://data.vlaanderen.be/ns/mijndomein#k03Arol_naam`  
`https://data.vlaanderen.be/ns/mijndomein#k03ArolTweede`  
`https://data.vlaanderen.be/ns/mijndomein#k03ArolTweede_naam`  
`https://data.vlaanderen.be/ns/mijndomein#k03Brol`  
`https://data.vlaanderen.be/ns/mijndomein#k03Brol_naam`  
`https://data.vlaanderen.be/ns/mijndomein#k03BrolTweede`  
`https://data.vlaanderen.be/ns/mijndomein#k03BrolTweede_naam`  
`https://data.vlaanderen.be/ns/mijndomein#k23Arol`  
`https://data.vlaanderen.be/ns/mijndomein#k23Arol_naam [name tag niet gelezen]`  
`https://data.vlaanderen.be/ns/mijndomein#k23ArolTweede`  
`https://data.vlaanderen.be/ns/mijndomein#k23ArolTweede_naam [name tag niet gelezen]`  
`https://data.vlaanderen.be/ns/mijndomein#k23Brol`  
`https://data.vlaanderen.be/ns/mijndomein#k23Brol_naam [name tag niet gelezen]`  
`https://data.vlaanderen.be/ns/mijndomein#k23BrolTweede`  
`https://data.vlaanderen.be/ns/mijndomein#k23BrolTweede_naam [name tag niet gelezen]`

## Geval 4: met associatieklasse, specifieke tags op associatieklasse

*Dit is de historische situatie die verder ondersteund wordt omwille van achterwaartse compatibiliteit.  
Voor nieuwe modellen: voorkeur geven aan oplossingen zonder specifieke tags en zonder richtingspijl.*

### UML voorbeeld



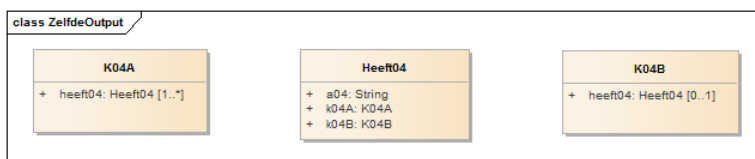
### Kenmerken

- Specifieke tags op associatieklasse: aanwezig
- Naam van de associatie: aanwezig
- Richtingspijl bij de naam van de associatie: aanwezig
- Rollen: afwezig

### Principes voor gegenereerde attributen

- Het genereren van attributen in bevattende klassen en associatieklassen wordt gestuurd door de tags **source-...**, **source-rev-...**, **target-...** en **target-rev-...** in de in de associatie.
- Het genereren van een attribuut kan worden onderdrukt door een respectievelijke tag **source-ignore**, **source-rev-ignore**, **target-ignore** of **target-rev-ignore** et waarde **true** toe te voegen.
- **Bemerk het ontbreken van een attribuut in de ene betrokken klasse dat verwijst naar de andere betrokken klasse: er wordt bijvoorbeeld in klasse K04A geen attribuut k04B gegenereerd.**

### UML met zelfde output als het voorbeeld



### Attributen in klassen (behalve associatieklasse)

#### Voorbeelden

heeft04

<https://data.vlaanderen.be/ns/mijndomein#K04A.heeft04>

<https://data.vlaanderen.be/ns/mijndomein#K04B.heeft04>

### Attributen in associatieklasse

#### Voorbeelden

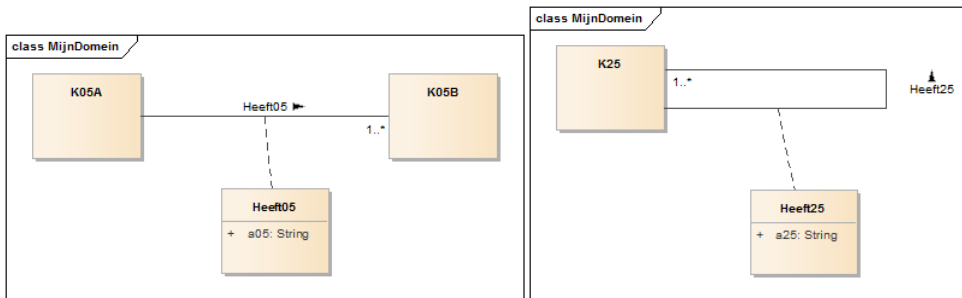
k04A, k04B

<https://data.vlaanderen.be/ns/mijndomein#Heeft04.k04A>  
<https://data.vlaanderen.be/ns/mijndomein#Heeft04.k04B>

## Geval 5: met associatieklasse, met richtingspijl, zonder rollen

**Voor nieuwe modellen: voorkeur geven aan oplossingen zonder richtingspijl.**

### UML voorbeeld



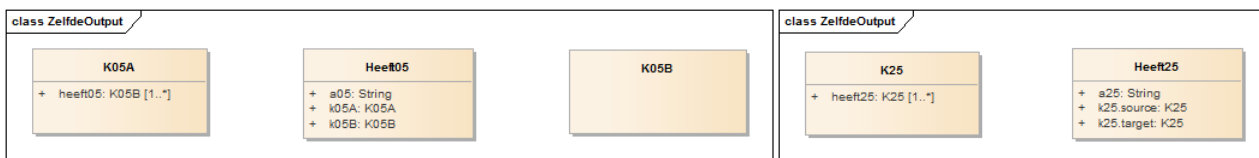
### Kenmerken

- Specifieke tags op associatieklasse: afwezig
- Naam van de associatie: aanwezig
- Richtingspijl bij de naam van de associatie: aanwezig
- Rollen: afwezig

### Principes voor gegenereerde attributen

- Er wordt een attribuut gegenereerd in de klasse aan de kant van de associatie waaruit de richtingspijl vertrekt.
- Er worden twee attributen gegenereerd in de associatieklasse, elk verwijzend naar een van de betrokken klassen.
- Waar nodig, worden aan label-nl en URI een suffix toegevoegd ter disambiguatie.

### UML met zelfde output als het voorbeeld



### Attributen in klassen (behalve associatieklasse)

**Formules 5.1.1-5.1.5 zijn dezelfde als 1.1.1-1.1.5 (de voorbeelden verschillen).**

#### Formule 5.1.1

```
label-nl(att) = tolower1(label-nl(assoc))
```

#### Voorbeelden

```
'heeft 05'  
'heeft 25'
```

#### Formule 5.1.2

**Enkel als** *assoc een tag* `ap-label-nl heeft`.

```
ap-label-nl(att) = tolower1(ap-label-nl(assoc))
```

#### Voorbeelden

```
'heeft 05 AP'  
'heeft 25 AP'
```

#### Formule 5.1.3

(Input voor bepaling uri van dit attribuut).

```
name(att) = tolower1(effectiveName(assoc))
```

Voorbeelden

```
heeft05, heeft05_naam  
heeft25, heeft25_naam
```

#### Formule 5.1.4

**Enkel als** assoc **een tag** uri **heeft**.

```
uri(att) = uri(assoc)
```

Voorbeelden

```
https://extern.domein#heeft05  
https://extern.domein#heeft25
```

#### Formule 5.1.5

```
uri(att) = baseURI(class) + name(att)
```

Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#heeft05  
https://data.vlaanderen.be/ns/mijndomein#heeft05_naam  
https://data.vlaanderen.be/ns/mijndomein#heeft25  
https://data.vlaanderen.be/ns/mijndomein#heeft25_naam
```

### Attributen in associatieklasse

#### Formule 5.2.1.a

```
label-nl(att_source) = tolower1(label-nl(class_source)) +  
disamb(' (source)', ' (target)')
```

Voorbeelden

```
'k 05A'  
'k 25 (source)'
```

#### Formule 5.2.1.b

```
label-nl(att_target) = tolower1(label-nl(class_target)) +  
disamb(' (source)', ' (target)')
```

Voorbeelden

```
'k 05B'  
'k 25 (target)'
```

#### Formule 5.2.2.a

**Enkel als** class\_source **een tag** ap-label-nl **heeft**.

```
ap-label-nl(att_source) = tolower1(ap-label-nl(class_source)) +  
disamb(' (source)', ' (target)')
```

Voorbeelden

```
'k 05A AP'  
'k 25 AP (source)'
```

#### Formule 5.2.2.b

**Enkel als** class\_target **een tag** ap-label-nl **heeft**.

```
ap-label-nl(att_target) = tolower1(ap-label-nl(class_target)) +  
disamb(' (source)', ' (target)')
```

Voorbeelden

```
'k 05B AP'  
'k 25 AP (target)'
```

#### Formule 5.2.3.a

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **een tag** name-source-class **heeft**.

```
name(att_source) = tolower1(name-source-class(class_assoc)) +
```

```
disamb('.source', '.target')
```

Voorbeelden

k05A [nog te implementeren]

k25.source [nog te implementeren]

### Formule 5.2.3.b

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **een tag** name-target-class **heeft**.

```
name(att_target) = tolower1(name-target-class(class_assoc)) +  
disamb('.source', '.target')
```

Voorbeelden

k05A [nog te implementeren]

k25.target [nog te implementeren]

### Formule 5.2.4.a

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **geen tag** name-source-class **heeft**.

```
name(att_source) = tolower1(effectiveName(class_source)) + disamb('.source', '.target')
```

Voorbeelden

k05A, k05A\_naam

k25.source, k25\_naam.source

### Formule 5.2.4.b

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **geen tag** name-target-class **heeft**.

```
name(att_target) = tolower1(effectiveName(class_target)) + disamb('.source', '.target')
```

Voorbeelden

k05B, k05B\_naam

k25.target, k25\_naam.target

### Formule 5.2.5.a

**Enkel als** class\_assoc **een tag** uri-source-attribute **heeft**.

```
uri(att_source) = uri-source-attribute(class_assoc)
```

Voorbeelden

https://extern.domein#Heeft05.k05A [nog te implementeren]

https://extern.domein#Heeft25.k25.source [nog te implementeren]

### Formule 5.2.5.b

**Enkel als** class\_assoc **een tag** uri-target-attribute **heeft**.

```
uri(att_target) = uri-target-attribute(class_assoc)
```

Voorbeelden

https://extern.domein#Heeft05.k05B [nog te implementeren]

https://extern.domein#Heeft25.k25.target [nog te implementeren]

### Formule 5.2.6.a

**Enkel als** class\_assoc **geen tag** uri-source-attribute **heeft**.

```
uri(att_source) = baseURI(class_assoc) + effectiveName(class_assoc) +  
'.' + name(att_source)
```

Voorbeelden

https://data.vlaanderen.be/ns/mijndomein#Heeft05.k05A

https://data.vlaanderen.be/ns/mijndomein#Heeft05\_naam.k05A\_naam

https://data.vlaanderen.be/ns/mijndomein#Heeft25.k25.source

https://data.vlaanderen.be/ns/mijndomein#Heeft25\_naam.k25\_naam.source

### Formule 5.2.6.b

**Enkel als** class\_assoc **geen tag** uri-target-attribute **heeft**.

```
uri(att_target) = baseURI(class_assoc) + effectiveName(class_assoc) +  
'.' + name(att_target)
```

### Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#Heeft05.k05B  
https://data.vlaanderen.be/ns/mijndomein#Heeft05_naam.k05B_naam  
https://data.vlaanderen.be/ns/mijndomein#Heeft25.k25.target  
https://data.vlaanderen.be/ns/mijndomein#Heeft25_naam.k25_naam.target
```

### Formule 5.2.7.a

```
definition-nl(att_source) = 'Referentie naar verbonden klasse.'
```

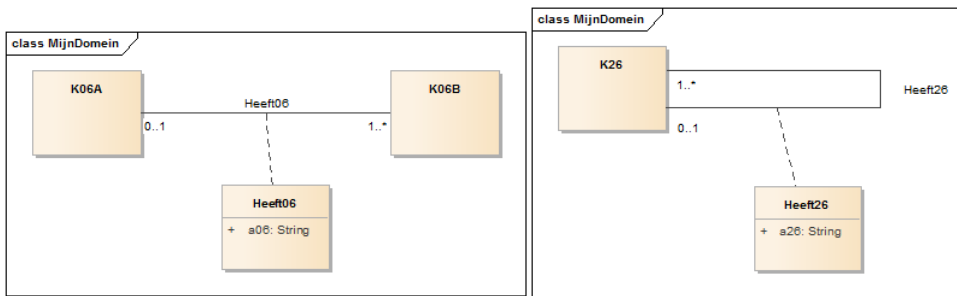
### Formule 5.2.7.b

```
definition-nl(att_target) = 'Referentie naar verbonden klasse.'
```



## Geval 6: met associatieklasse, zonder richtingspijl, zonder rollen

### UML voorbeeld



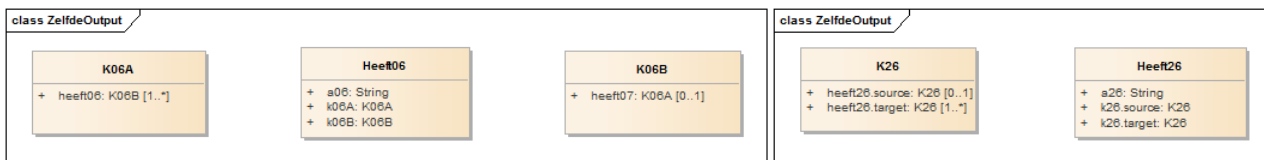
### Kenmerken

- Specifieke tags op associatieklasse: afwezig
- Naam van de associatie: aanwezig
- Richtingspijl bij de naam van de associatie: afwezig
- Rollen: afwezig

### Principes voor gegenereerde attributen

- Er wordt een attribuut gegenereerd in beide betrokken klassen, telkens verwijzend naar de andere betrokken klasse.
- Er worden twee attributen gegenereerd in de associatieklasse, elk verwijzend naar een van de betrokken klassen.
- Waar nodig, wordt aan eigenschappen een suffix toegevoegd ter disambiguatie.

### UML met zelfde output als het voorbeeld



### Attributen in klassen (behalve associatieklasse)

*Formules 6.1.1-6.1.5 zijn dezelfde als 2.1.1-2.1.5 (de voorbeelden verschillen).*

#### Formule 6.1.1

```
label-nl(att) = tolower1(label-nl(assoc)) + disamb(' (source)', ' (target)')
```

Voorbeelden

'heeft 06'

'heeft 26 (source)', 'heeft 26 (target)'

#### Formule 6.1.2

*Enkel als assoc een tag ap-label-nl heeft.*

```
label-nl(att) = tolower1(ap-label-nl(assoc)) + disamb(' (source)', ' (target)')
```

Voorbeelden

'heeft 06 AP'

'heeft 26 AP (source)', 'heeft 26 AP (target)'

#### Formule 6.1.3

(Input voor bepaling uri van dit attribuut).

```
name(att) = effectiveName(class) + '.' + tolower1(effectiveName(assoc)) +  
disamb('.source', '.target')
```

#### Voorbeelden

```
K06A.heeft06, K06A_naam.heeft06_naam,  
K06B.heeft06, K06B_naam.heeft06_naam,  
K26.heeft26.source, K26_naam.heeft26_naam.source,  
K26.heeft26.target, K26_naam.heeft26_naam.target
```

#### Formule 6.1.4

**Enkel als** assoc *een tag* uri *heeft*.

```
uri(att) = uri(assoc)
```

#### Voorbeelden

```
https://extern.domein#heeft06  
https://extern.domein#heeft26
```

#### Formule 6.1.5

**Enkel als** assoc *geen tag* uri *heeft*.

```
uri(att) = baseURI(class) + name(att)
```

#### Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#K06A.heeft06  
https://data.vlaanderen.be/ns/mijndomein#K06A_naam.heeft06_naam  
https://data.vlaanderen.be/ns/mijndomein#K06B.heeft06  
https://data.vlaanderen.be/ns/mijndomein#K06B_naam.heeft06_naam  
https://data.vlaanderen.be/ns/mijndomein#K26.heeft26.source  
https://data.vlaanderen.be/ns/mijndomein#K26_naam.heeft26_naam.source  
https://data.vlaanderen.be/ns/mijndomein#K26.heeft26.target  
https://data.vlaanderen.be/ns/mijndomein#K26_naam.heeft26_naam.target
```

### Attributen in associatieklasse

**Formules 6.2.1-6.2.7 zijn dezelfde als 5.2.1-5.2.7 (de voorbeelden verschillen).**

#### Formule 6.2.1.a

```
label-nl(att_source) = tolower1(label-nl(class_source)) +  
disamb(' (source)', ' (target)')
```

#### Voorbeelden

```
'k 06A'  
'k 26 (source)'
```

#### Formule 6.2.1.b

```
label-nl(att_target) = tolower1(label-nl(class_target)) +  
disamb(' (source)', ' (target)')
```

#### Voorbeelden

```
'k 06B'  
'k 26 (target)'
```

#### Formule 6.2.2.a

**Enkel als** class\_source *een tag* ap-label-nl *heeft*.

```
ap-label-nl(att_source) = tolower1(ap-label-nl(class_source)) +  
disamb(' (source)', ' (target)')
```

#### Voorbeelden

```
'k 06A AP'  
'k 26 AP (source)'
```

#### Formule 6.2.2.b

**Enkel als** class\_target *een tag* ap-label-nl *heeft*.

```
ap-label-nl(att_target) = tolower1(ap-label-nl(class_target)) +  
disamb(' (source)', ' (target)')
```

## Voorbeelden

```
'k 06B AP'  
'k 26 AP (target)'
```

### Formule 6.2.3.a

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **een tag** name-source-class **heeft**.

```
name(att_source) = tolower1(name-source-class(class_assoc)) +  
disamb('source', 'target')
```

## Voorbeelden

```
k06A [nog te implementeren]  
k26.source [nog te implementeren]
```

### Formule 6.2.3.b

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **een tag** name-target-class **heeft**.

```
name(att_target) = tolower1(name-target-class(class_assoc)) +  
disamb('source', 'target')
```

## Voorbeelden

```
k06B [nog te implementeren]  
k26.target [nog te implementeren]
```

### Formule 6.2.4.a

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **geen tag** name-source-class **heeft**.

```
name(att_source) = tolower1(effectiveName(class_source)) + disamb('source', 'target')
```

## Voorbeelden

```
k06A, k06A_naam  
k26.source, k26_naam.source
```

### Formule 6.2.4.b

(Input voor bepaling uri van dit attribuut).

**Enkel als** class\_assoc **geen tag** name-target-class **heeft**.

```
name(att_target) = tolower1(effectiveName(class_target)) + disamb('source', 'target')
```

## Voorbeelden

```
k06B, k06B_naam  
k26.target, k26_naam.target
```

### Formule 6.2.5.a

**Enkel als** class\_assoc **een tag** uri-source-attribute **heeft**.

```
uri(att_source) = uri-source-attribute(class_assoc)
```

## Voorbeelden

```
https://extern.domein#Heeft06.k06A [nog te implementeren]  
https://extern.domein#Heeft26.k26.source [nog te implementeren]
```

### Formule 6.2.5.b

**Enkel als** class\_assoc **een tag** uri-target-attribute **heeft**.

```
uri(att_target) = uri-target-attribute(class_assoc)
```

## Voorbeelden

```
https://extern.domein#Heeft06.k06B [nog te implementeren]  
https://extern.domein#Heeft26.k26.target [nog te implementeren]
```

### Formule 6.2.6.a

**Enkel als** class\_assoc **geen tag** uri-source-attribute **heeft**.

```
uri(att_source) = baseURI(class_assoc) + effectiveName(class_assoc) +  
'.' + name(att_source)
```

### Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#Heeft06.k06A
https://data.vlaanderen.be/ns/mijndomein#Heeft06_naam.k06A_naam
https://data.vlaanderen.be/ns/mijndomein#Heeft26.k26.source
https://data.vlaanderen.be/ns/mijndomein#Heeft26_naam.k26_naam.source
```

### Formule 6.2.6.b

*Enkel als* class\_assoc *geen tag* uri-target-attribute *heeft*.

```
uri(att_target) = baseURI(class_assoc) + effectiveName(class_assoc) +
'.' + name(att_target)
```

### Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#Heeft06.k06B
https://data.vlaanderen.be/ns/mijndomein#Heeft06_naam.k06B_naam
https://data.vlaanderen.be/ns/mijndomein#Heeft26.k26.target
https://data.vlaanderen.be/ns/mijndomein#Heeft26_naam.k26_naam.target
```

### Formule 6.2.7.a

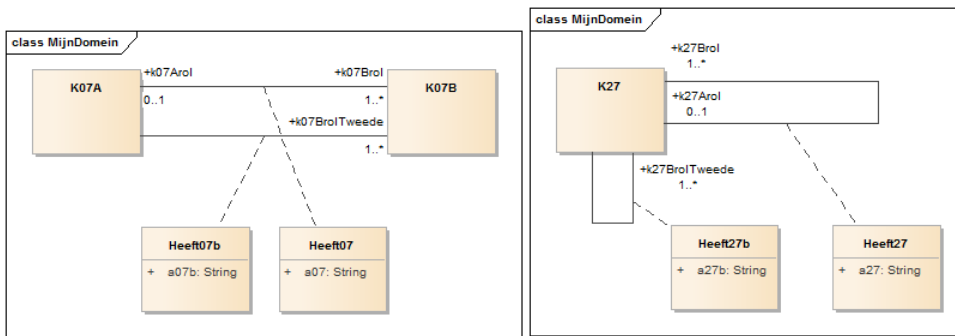
```
definition-nl(att_source) = 'Referentie naar verbonden klasse.'
```

### Formule 6.2.7.b

```
definition-nl(att_target) = 'Referentie naar verbonden klasse.'
```

## Geval 7: met associatieklasse, zonder richtingspijl, met rollen

### UML voorbeeld



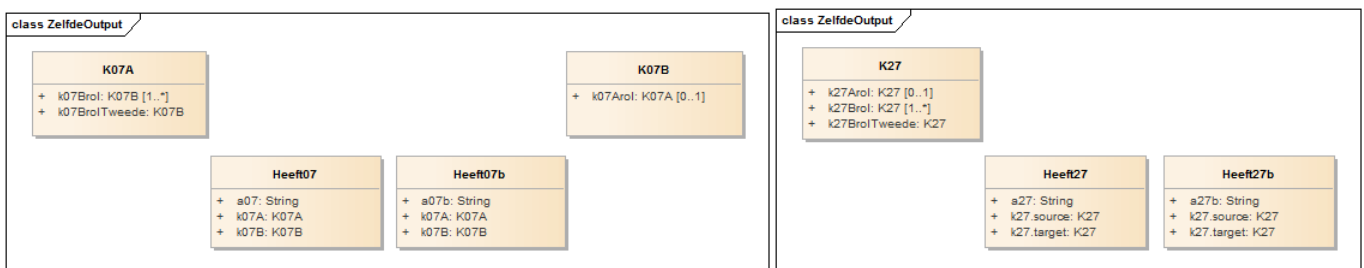
### Kenmerken

- Specifieke tags op associatieklasse: afwezig
- Naam van de associatie: afwezig
- Richtingspijl bij de naam van de associatie: afwezig
- Rollen: aanwezig

### Principes voor gegenereerde attributen

- Enkel als aan een uiteinde van een associatie een rol is toegekend, wordt in de klasse aan het andere uiteinde van de associatie een attribuut gegenereerd.
- Er worden twee attributen gegenereerd in de associatieklasse, elk verwijzend naar een van de betrokken klassen.
- Waar nodig, wordt aan eigenschappen een suffix toegevoegd ter disambiguatie.

### UML met zelfde output als het voorbeeld



### Attributen in klassen (behalve associatieklasse)

**Formules 7.1.1-7.1.5 zijn dezelfde als 3.1.1-3.1.5 (de voorbeelden verschillen).**

#### Formule 7.1.1

```
label-nl(att) = label-nl(rol)
```

#### Voorbeelden

```
'k 07A rol', 'k 07B rol'
'k 07B rol Tweede'
'k 27A rol', 'k 27B rol' [leeg]
'k 27B rol Tweede' [leeg]
```

#### Formule 7.1.2

**Enkel als rol een tag** ap-label-nl **heeft.**

```
ap-label-nl(att) = ap-label-nl(rol)
```

## Voorbeelden

```
'k 07A rol AP', 'k 07B rol AP'
'k 07B rol Tweede AP',
'k 27A rol AP', 'k 27B rol AP' [leeg]
'k 27B rol Tweede AP' [leeg]
```

### Formule 7.1.3

(Input voor bepaling `uri` van dit attribuut).

```
name(att) = effectiveName(rol)
```

## Voorbeelden

```
k07Arol, k07Arol_naam
k07Brol, k07Brol_naam, k07BrolTweede, k07BrolTweede_naam
k27Arol, k27Arol_naam
k27Brol, k27Brol_naam, k27BrolTweede, k27BrolTweede_naam
```

### Formule 7.1.4

*Enkel als rol een tag uri heeft.*

```
uri(att) = uri(rol)
```

## Voorbeelden

```
https://extern.domein#k07Arol
https://extern.domein#k07Brol
https://extern.domein#k07BrolTweede
https://extern.domein#k27Arol [uri tag niet gelezen]
https://extern.domein#k27Brol [uri tag niet gelezen]
https://extern.domein#k27BrolTweede [uri tag niet gelezen]
```

### Formule 7.1.5

*Enkel als rol geen tag uri heeft.*

```
uri(att) = baseURI(class) + name(att)
```

## Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#k07Arol
https://data.vlaanderen.be/ns/mijndomein#k07Arol_naam
https://data.vlaanderen.be/ns/mijndomein#k07Brol
https://data.vlaanderen.be/ns/mijndomein#k07Brol_naam
https://data.vlaanderen.be/ns/mijndomein#k07BrolTweede
https://data.vlaanderen.be/ns/mijndomein#k07BrolTweede_naam
https://data.vlaanderen.be/ns/mijndomein#k27Arol
https://data.vlaanderen.be/ns/mijndomein#k27Arol_naam [name tag niet gelezen]
https://data.vlaanderen.be/ns/mijndomein#k27Brol
https://data.vlaanderen.be/ns/mijndomein#k27Brol_naam [name tag niet gelezen]
https://data.vlaanderen.be/ns/mijndomein#k27BrolTweede
https://data.vlaanderen.be/ns/mijndomein#k27BrolTweede_naam [name tag niet gelezen]
```

## Attributen in associatieklasse

*Formules 7.2.1-7.2.7 zijn dezelfde als 5.2.1-5.2.7 (de voorbeelden verschillen).*

### Formule 7.2.1.a

```
label-nl(att_source) = tolower1(label-nl(class_source)) +
disamb(' (source)', ' (target)')
```

## Voorbeelden

```
'k 07A'
'k 27 (source)'
```

### Formule 7.2.1.b

```
label-nl(att_target) = tolower1(label-nl(class_target)) +
disamb(' (source)', ' (target)')
```

## Voorbeelden

```
'k 07B'
'k 27 (target)'
```

### Formule 7.2.2.a

**Enkel als** `class_source` **een tag** `ap-label-nl` **heeft.**

```
ap-label-nl(att_source) = tolower1(ap-label-nl(class_source)) +  
disamb(' (source)', ' (target)')
```

Voorbeelden

```
'k 07A AP',  
'k 27 AP (source)'
```

### Formule 7.2.2.b

**Enkel als** `class_target` **een tag** `ap-label-nl` **heeft.**

```
ap-label-nl(att_target) = tolower1(ap-label-nl(class_target)) +  
disamb(' (source)', ' (target)')
```

Voorbeelden

```
'k 07B AP'  
'k 27 AP (target)'
```

### Formule 7.2.3.a

(Input voor bepaling uri van dit attribuut).

**Enkel als** `class_assoc` **een tag** `name-source-class` **heeft.**

```
name(att_source) = name-source-class(class_assoc) + disamb('.source', '.target')
```

Voorbeelden

```
k07A [nog te implementeren]  
k27.source [nog te implementeren]
```

### Formule 7.2.3.b

(Input voor bepaling uri van dit attribuut).

**Enkel als** `class_assoc` **een tag** `name-target-class` **heeft.**

```
name(att_target) = name-target-class(class_assoc) + disamb('.source', '.target')
```

Voorbeelden

```
k07B [nog te implementeren]  
k27.target [nog te implementeren]
```

### Formule 7.2.4.a

(Input voor bepaling uri van dit attribuut).

**Enkel als** `class_assoc` **geen tag** `name-source-class` **heeft.**

```
name(att_source) = tolower1(effectiveName(class_source)) + disamb('.source', '.target')
```

Voorbeelden

```
k07A, k07A_naam  
k27.source, k27_naam.source
```

### Formule 7.2.4.b

(Input voor bepaling uri van dit attribuut).

**Enkel als** `class_assoc` **geen tag** `name-target-class` **heeft.**

```
name(att_target) = tolower1(effectiveName(class_target)) + disamb('.source', '.target')
```

Voorbeelden

```
k07B, k07B_naam  
k27.target, k27_naam.target
```

### Formule 7.2.5.a

**Enkel als** `class_assoc` **een tag** `uri-source-attribute` **heeft.**

```
uri(att_source) = uri-source-attribute(class_assoc)
```

Voorbeelden

```
https://extern.domein#Heeft07.k07A [nog te implementeren]  
https://extern.domein#Heeft07b.k07A [nog te implementeren]  
https://extern.domein#Heeft27.k27.source [nog te implementeren]  
https://extern.domein#Heeft27b.k27.source [nog te implementeren]
```

### Formule 7.2.5.b

**Enkel als** `class_assoc` *een tag* `uri-target-attribute` *heeft*.

```
uri(att_target) = uri-target-attribute(class_assoc)
```

#### Voorbeelden

```
https://extern.domein#Heeft07.k07B [nog te implementeren]
https://extern.domein#Heeft07b.k07B [nog te implementeren]
https://extern.domein#Heeft27.k27.target [nog te implementeren]
https://extern.domein#Heeft27b.k27.target [nog te implementeren]
```

### Formule 7.2.6.a

**Enkel als** `class_assoc` *geen tag* `uri-source-attribute` *heeft*.

```
uri(att_source) = baseURI(class_assoc) + effectiveName(class_assoc) +
'.' + name(att_source)
```

#### Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#Heeft07.k07A
https://data.vlaanderen.be/ns/mijndomein#Heeft07_naam.k07A_naam
https://data.vlaanderen.be/ns/mijndomein#Heeft27.k27.source
https://data.vlaanderen.be/ns/mijndomein#Heeft27_naam.k27_naam.source
```

### Formule 7.2.6.b

**Enkel als** `class_assoc` *geen tag* `uri-target-attribute` *heeft*.

```
uri(att_target) = baseURI(class_assoc) + effectiveName(class_assoc) +
'.' + name(att_target)
```

#### Voorbeelden

```
https://data.vlaanderen.be/ns/mijndomein#Heeft07.k07B
https://data.vlaanderen.be/ns/mijndomein#Heeft07_naam.k07B_naam
https://data.vlaanderen.be/ns/mijndomein#Heeft27.k27.target
https://data.vlaanderen.be/ns/mijndomein#Heeft27_naam.k27_naam.target
```

### Formule 7.2.7.a

```
definition-nl(att_source) = 'Referentie naar verbonden klasse.'
```

### Formule 7.2.7.b

```
definition-nl(att_target) = 'Referentie naar verbonden klasse.'
```