Typen

|  |  |  |
| --- | --- | --- |
| Joop Ringelberg | 27-11-17 | Versie: 1 |

# Introductie

In de tekst *MetaTypen* heb ik beschreven hoe ContextType, RolInContextType, PropertyType en PropertyPerspectief in termen van rollen en contexten beschreven kunnen worden. Ik heb beschreven hoe het pragma-attribuut door het systeem gebruikt kan worden om de modelleur te ondersteunen als hij concrete typen gaat maken. In deze tekst beschrijf ik hoe zulke typen opgeschreven kunnen worden in de Perspect modelleertaal en hoe we zulke typen representeren in termen van contexten en rollen.

Ik doe dat aan de hand van het voorbeeld van Aangifte, met één rol, namelijk Aangever, een private property ‘aantekeningen’ en één publieke property: ‘urgentie’. De Aangever voorzien we van een property ‘betrouwbaarheid’. Op Aangever construeren we een propertyperspectief.

# Definities in tekst

Het ContextType Aangifte heeft de volgende tekstuele definitie:

:ContextType :Aangifte

private :aantekeningen

:isFunctional = true

:isVerplicht = false

:range = String

public :urgentie

:isFunctional = true

:isVerplicht = true

:range = Number

:RolInContextType :Aangever

isFunctional = false

isVerplicht = true

:MogelijkeBinding => :Gebruiker

:PropertyType :betrouwbaarheid

:isFunctional = true

:isVerplicht = true

:range = Number

:PropertyPerspectief :aangeverBetrouwbaarheid

:volgnummer = 1

:PropertyReferentie => :betrouwbaarheid

Merk op dat alle overige definities in context van :Aangifte staan.

## Aangifte gerepresenteerd

We representeren Aangifte met een context. De structurele onderdelen moeten dus rollen zijn; simpele aspecten van Aangifte zouden we met properties kunnen representeren. Maar zulke aspecten zijn er niet, wat direct blijkt uit de definitie van ContextType zelf; daar zien we vier rollen, elk met het pragma typeDefinition. Deze rollen zijn:

* RolInContextType (TD): in dit voorbeeld is dat Aangever.
* BinnenRolType (TD): hier komt de private property :aantekeningen terecht.
* BuitenRolType (TD): hier komt de public property :urgentie.
* BerekendeRolType (TD): niet aanwezig in ons voorbeeld.

Merk op dat we van de drie aanwezige componenten (twee properties, één rol) hebben aangegeven of ze functioneel en/of verplicht zijn. Dit kan, omdat in de definitie van RolInContextType óók de rollen isFunctioneel en isVerplicht zijn opgenomen (met pragma propertyAssignment). In diagram ziet de context voor Aangifte er dan als volgt uit:

RolInContextType

Aangever

Aantekeningen

Urgentie

Aangifte

1-n

PA

0-1

PA

1

PA

range=String

range=Number

privatePropertyType

publicPropertyType

## Aangever

Aangever is natuurlijk een rol in een concrete aangifte, maar in de definitie representeren we het als een context. In de definitie van RolInContextType zien we zes structurele onderdelen:

* PropertyType (TD): in dit voorbeeld is dat :betrouwbaarheid
* PropertyPerspectief (TD): in dit voorbeeld is dat :aangeverBetrouwbaarheid
* MogelijkeBinding (RB): in het voorbeeld binden we aan :Gebruiker
* isVerplicht (PA): in het voorbeeld true
* isFunctioneel (PA): in het voorbeeld false.
* pragma: niet genoemd in het voorbeeld.

We krijgen dus een context met twee roldefinities, één rolbinding en twee property-assignments. Merk op dat de roldefinities en de rolbinding alle drie rollen zijn, maar dat hun binding anders uitwerkt. Voor het propertytype en het perspectief creëren we nieuwe onderdelen, voor de rolbinding verwijzen we naar een al bestaande context (die :Gebruiker representeert).

Betrouwbaarheid

AangeverBetrouwbaarheid

Aangever

1

PA

0-1

TD

Gebruiker

propertyType

propertyPerspectief

mogelijkeBinding

Volgnummer=1

propertyReferentie

Bovendien krijgen we twee properties