**MEMO**

|  |  |
| --- | --- |
| **Onderwerp** | Metamodel MIM/KKG – verzoek: zelf gedefinieerd datatype baseren op ander datatype |
| **Aan** | Paul Janssen, Ellen Debats, Arjan Kloosterboer, Arjan Loeffen |
| **Van** | Lennart van Bergen, Linda van den Brink, Peter Lentjes |
| **Datum** | 2018-06-12 |
| **Status** | Review 2e ronde verwerkt |

**Inleiding**

#29. <https://github.com/Geonovum/KKG-Metamodel/issues/29>

Op verzoek van Kadaster/IMKAD, Geonovum/BRO: hoe geef je aan dat een zelf gedefinieerd datatype, gebaseerd is op ander datatype?

Bijvoorbeeld:

* primitief datatype AN is gebaseerd op CharacterString (primitief datatype, extern aan MIM)
* NietNegatiefGetal is gebaseerd op Integer (primitief datatype, extern aan MIM)
* Vlak is gebaseerd op GM\_Surface (niet primitief datatype, extern aan MIM)

Bijna altijd is het onderliggende datatype een CharacterString, en dit wordt nu behandeld als de default, maar dit is niet altijd zo. Daarnaast is het netjes om het expliciet aan te geven.

In MIM staat nu:

*Het is ook mogelijk om in het eigen informatiemodel een eigen primitive datatype te definiëren in de vorm van een «Primitief datatype», met als UML metaclass de UML-Primitive datatype.*

***Voorbeelden*** *hiervan, die niet tot KKG behoren, maar ter illustratie zijn opgenomen, zijn:*

* *NietNegatieveInteger: een Integer die alleen de waarde 0 of groter mag hebben. Laat de naam van het primitieve type dan wel terugkomen in de naam (dus niet NietNegatiefGetal).*
* *Een beperking op een Real te specificeren door Decimal op te nemen (een gebroken getal, met (één of meer) cijfers voor de komma en cijfers achter de komma, conform ISO11404).*
* *AN. Deze is gebruikelijk bij een aantal basisregistraties. Datatype met een eigen naam, analoog aan CharacterString, maar met alleen ‘normale’ tekens. Dit zijn alle alfanumerieke tekens (dus inclusief diakrieten), de koppeltekens – en \_ en spaties. De minimale lengte is tenminste 1, de maximale lengte is onbepaald. De 1e positie mag géén spatie bevatten*

Deze uitwerking gaat primair over het specificeren van een zelf-gedefinieerd datatype dat gebaseerd is op een ander datatype, dat geen MIM stereotype heeft. Maar het is in principe ook toegestaan om structuur over te erven, zoals een specifieke <<referentielijst>>, die de structuur overerft van een generieke <<referentielijst>>, of bij een Bedrag en een Positief Bedrag. Het gaat dus niet alleen over primitieve datatypen, maar ook over gestructureerde datatypen.

**De insteek bij de uitwerking die gevolgd is …**

Een zelf-gedefinieerd datatype maken doe je alleen als je eigen datatype een verbijzondering/ aanscherping is van een ander datatype, waarbij dit andere datatype al dan niet extern is aan je eigen informatiemodel.

We modelleren dit als volgt:

* UML-Datatype, met een eigen naam (dit staat al zo in MIM).
* Deze heeft een UML-Generalisatie relatie van het zelf gedefinieerde type naar het datatype waar het een verbijzondering van is.

Merk op dat de definitie van de datatypes in dit geval het verschil benoemt met het generiekere datatype, eventueel in combinatie met een constraint (OCL of tagged value).

**Status van dit voorstel**

Wat betreft het voorstel zoals verderop in het document aangegeven:

* Standpunt Geonovum: Ok.
* Standpunt Kadaster: Ok.
* Standpunt VNG: Ok.   
  Oftewel, goedgekeurd. Het voorstel zal zoals uitgewerkt worden overgenomen naar MIM 1.1.

**Uitwerking: Generalisatie van datatype naar datatype.**

Voor «Primitief datatype», «Gestructureerd datatype», «Referentielijst» .

Niet voor «Codelist», «Enumeratie », «Union».

Definitie

**MIM Hoofdstuk 2**

**Paragraaf 2.2.2**

**5. Generalisatie:** De UML-representatie van een specialisatie, uitgedrukt in een UML-generalization (metaclass).

*Toevoegen:* Een generalisatie kan worden gelegd tussen objecttypes of tussen datatypes. In beide gevallen spreken we van een generalisatie, maar de definitie verschilt:

(bestaande tekst, te handhaven):

***Definitie Generalisatie tussen objecttypes***

*De typering van het hiërarchische verband tussen een meer generiek object van een objecttype en een meer specifiek object van een ander objecttype waarbij het laatstgenoemde object eigenschappen van het eerstgenoemde object overerft.*

(toe te voegen nieuwe tekst):

**Definitie Generalisatie tussen datatypes**

De typering van het hiërarchische verband tussen een meer generieke structuur van data in de vorm van een datatype, en een meer specifieke structuur van data in de vorm van een ander datatype, waarbij het laatstgenoemde datatype de eigenschappen van het eerstgenoemde datatype overerft, én een verbijzondering hierin aanbrengt in de vorm van een meer restrictieve definitie, of een meer restrictief patroon/formeel patroon.

Toelichting:

* het andere datatype is bijvoorbeeld een CharacterString, Integer, GM Surface of DMO en dient als basis voor een zelf te definiëren datatype (zie 3.1.2.).
* deze generalisatie is van toepassing op de volgende datatypes: «Primitief datatype», «Gestructureerd datatype», «Referentielijst», «Codelist».

**Paragraaf 2.1**

Generalisatie toevoegen aan 2.1.2.

**Paragraaf 2.3**

**Specificatie voor «Generalisatie» tussen objecttypes**

*Tekst handhaven*

**H3**

**Specificatie voor «Generalisatie» tussen datatypes**

De generalisaties worden naar het volgende aspect gespecificeerd:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Aspect** | **Kardi**  **naliteit** | **Toelichting** | **In UML 2.5** | **In EA** |
| **Naam** | 0..1 | De naam van de generalisatie.  Standaard ‘is specialisatie van’. | *name van de metaclass Named element* | *Name* |
| **Datatype** | 1 | Het datatype dat een specialisatie is van een (ander) datatype. | */source: related*  *Element bij Relationship* *Element* | *Source* |
| **Gerelateerd datatype** | 1 | Het datatype dat de generalisatie is van een (ander) datatype. | */target: related*  *Element bij Relationship* *Element* | *Target* |

**3.1.2. Primitief datatype zelf definiëren** 🡪 herformuleren naar **3.1.2. Datatype zelf definiëren**

Het is ook mogelijk om in het eigen informatiemodel een eigen datatype te definiëren in de vorm van een «Primitief datatype», «Gestructureerd datatype», <<Codelist>>, of «Referentielijst». Zelf gedefinieerde datatypes hebben altijd een eigen definitie en optioneel een eigen patroon of formeel patroon.

**V*oorbeelden*** hiervan, die niet tot MIM behoren, maar ter illustratie zijn opgenomen, zijn:

* NietNegatieveInteger: een Integer die alleen de waarde 0 of groter mag hebben. Laat de naam van het primitieve type dan wel terugkomen in de naam (dus niet NietNegatiefGetal).
* Een beperking op een Real te specificeren door Decimal op te nemen (een gebroken getal, met (één of meer) cijfers voor de komma en cijfers achter de komma, conform ISO11404).
* AN. Deze is gebruikelijk bij een aantal basisregistraties. Datatype met een eigen naam, analoog aan CharacterString, maar met alleen ‘normale’ tekens. Dit zijn alle alfanumerieke tekens (dus inclusief diakrieten), de koppeltekens – en \_ en spaties. De minimale lengte is tenminste 1, de maximale lengte is onbepaald. De 1e positie mag géén spatie bevatten.
* Een Vlak: een verbijzondering van een GM Surface, met een eigen definitie, die bijvoorbeeld aangeeft dat het om een 2 dimensionale geometrie gaat.



De gele datatypes zijn extern aan het model.

Het type modelelement (stereotype) verandert niet door de generalisatie. Een zelf gedefinieerd primitief datatype zal een generalisatie hebben met een ander primitief datatype. Een zelf gedefinieerd gestructureerd datatype zal een generalisatie hebben met een ander gestructureerd datatype.

Het komt voor dat het zelf gedefinieerde datatype een generalisatie heeft naar een extern gedefinieerd datatype, waarvan het modelelement (stereotype) niet is gespecificeerd. Maak dan zelf een inschatting.

Let hierbij op bij een «Gestructureerd datatype». Deze hoort altijd twee of meer data elementen te hebben.

Verdere kanttekeningen een aandachtspunten (niet om op te nemen in MIM, tenzij behoefte):

* Op pagina 28 staat:   
  Wanneer een Primitief datatype wordt gespecificeerd, dan heeft deze standaard als primitive datatype een CharacterString.  
    
  Dat mag zo blijven. Het is dus alleen nodig in een UML model om een generalisatie te specificeren als de target hiervan niet een CharacterString is. Dat mag natuurlijk wel, maar het hoeft niet.
* Als je een datatype uit wilt breiden, met een extra data element, wat doen je dan.
  + Creëer je dan een Gestructureerd datatype met data element vlak:GM\_Surface en oppervlakte: Integer? Oftewel, je mag géén data elementen toevoegen nadat je Generalisatie hebt toegepast. Nee.
  + Creëer je dan een Gestructureerd datatype met twee attributen, die je laat overerven van een primitief datatype? Nee.
  + Dit soort items opnemen in de standaard? N.t.b. gaat om toelichting in H3, dus is klein.
* Als je externe datatype een GML Surface is, die heel complex in elkaar zit, zeggen we dan dat dit een primitief datatype is of een gestructureerd datatype? Antwoord: valt onder, maak zelf een inschatting. Maar geven we hier nog een nadere richtlijn voor? N.t.b. gaat om toelichting in H3, dus is klein.
* Als je bv. wilt aangeven dat het 1e character niet een spatie of een whitespace mag zijn. Dat is een nadere specificatie van CharacterString. Dit kan je bv. opnemen in een datatype AN, die je laat extenden van CharacterString. Is dit een goed voorbeeld om op te nemen? Ja.
* Moeten we een apart package creëren om Integer of CharacterString op te nemen? Want die horen niet bij GML package.

**MIM Hoofdstuk 3**

**Bijlage: standaard werkwijze afhandelen punten**

1. Bepalen wel of niet in scope bepalen   
   Ja.
2. Bepalen of het hoort bij conceptueel en/of logisch, of in een extensie  
   Conceptueel en logisch, MIM zelf.
3. Richting/alternatieven bespreken  
   Eerste stap gezet, zie hierboven. Inventariseren ideeën.
4. Actiehouder bepalen  
   Lennart en Linda.
5. Voorstel uitwerken door actiehouder  
   Lennart.
6. Ter review rondsturen door actiehouder en review verwerken.   
   Linda.
7. Vaststellen door kernteam: zie pagina 2. Er is een OK.   
   Als niet OK: n.v.t.
   1. discussie overleg inplannen
   2. beslissen, indien mogelijk
   3. bijzonder overleg, indien nodig
8. Voorstel delen met een aantal mensen uit de doelgroep in de nabije periferie.   
   TODO.
9. Bij besluit: verwerken in standaard – document én profiel, met een achteraf controle of het nu goed is.   
   TODO.
10. Vaststellen nieuwe versie van de standaard.   
    TODO.