Hoe Rollen Properties dragen

|  |  |  |
| --- | --- | --- |
| Joop Ringelberg | 10-04-18 – 11-04-18 | Versie: 2 |

# Eigen properties

Een Rol wordt gedefinieerd in termen van zijn Properties. Zij worden in de definitie van de Rol opgenomen, in de sectie *rollen* van de Arc tekst. Zo’n Property heeft een *lokale naam* die gekwalificeerd wordt met de (gekwalificeerde) naam van de Rol (de Rolnaam dient als *namespace*).

We zeggen dat zo’n Rol deze properties *draagt*.

# Aspect properties

Maar een Rol draagt óók de properties van eraan gekoppelde AspectRollen die Properties bijdragen. Deze properties hebben gekwalificeerde namen in de namespace van hun Aspect.

## Run time representatie van eigen- en aspect properties

Instanties van een Rol worden gerepresenteerd met een Purescript Record waarin waarden vastgelegd zijn voor Properties van beide soorten herkomst. Om die reden slaan we deze waarden op onder de volledige, gekwalificeerde naam als sleutel. We noemen deze properties *lokaal gerepresenteerd*. ‘Lokale representatie’ is dus een *run time* begrip!

# Primitieve- en GebondenRollen

Tot nog toe hebben we Rollen op zichzelf bekeken. Maar een rol in de instantie van een Context kan een binding met een andere rolinstantie hebben. Dat kan zowel een RolInContext, als de BuitenRol van een Context zijn. Deze binding is van belang in verband met properties die gedragen worden door een Rol.

Een rolinstantie met een binding is een voorbeeld van een *gebonden rol*.

Kan een rolinstantie geen binding hebben, dan is het een voorbeeld van een *primitieve rol.*

Een Rol is dus een gebonden rol, of een primitieve rol. De ‘of’ duidt op een *Sum type*. Dat leidt tot het volgende pseudo-type[[1]](#footnote-1):

Rol = ‘gebonden rol’ of ‘primitieve rol’

## Primitieve rol

Een primitieve rol is de rol zoals we hem hierboven hebben besproken hebben. Zo’n rol heeft alleen eigen properties, of aspect properties.

Maar hoe zit het met de properties van een gebonden rol?

## Gebonden rol

Een gebonden rol is de combinatie van zowèl een Rol àls een ander Type. Dit soort combinatie, van onderdelen, heet een *product type*. Ik kies de naam gebondenAan om het product mee aan te geven, bijvoorbeeld zo: Rol gebondenAan Rol.

Hier binden we een Rol aan:

* een Rol, of
* een Context, of
* een Sum type van Rollen, of
* een Sum type van Contexten.

Opnieuw duidt het gebruik van ‘of’ op een Sum type. Als we ’gebonden rol’ in het pseudo type van Rol uitwerken, dan krijgen we:

Rol = ‘primitieve rol’

of (Rol gebondenAan Rol)

of (Rol gebondenAan Context)

of (Rol gebondenAan (Rol of Rol of …))

of (Rol gebondenAan (Context of Context of …))

In termen van dit pseudo-type van Rol kunnen we een Context beschrijven.

# Properties van Rol

Een Rol heeft dus vijf verschijningsvormen. We kunnen de gedragen properties per verschijningsvorm aangeven:

* de properties van een primitieve rol zijn zijn eigen- en aspect properties;
* de properties van een Rol gebonden aan een Rol is de **vereniging** van de properties van beiden;
* de properties van een Rol gebonden aan een Context is de **vereniging** van de properties van de Rol en de BuitenRol van de Context;
* de properties van een Rol gebonden aan een Sum van Rollen is de **doorsnede** van de properties van die Rollen;
* de properties van een Rol gebonden aan een Sum van Contexten is de **doorsnede** van de properties van hun BuitenRollen.

## Run time representatie van properties van een gebonden rol

De waarde van een Property van die niet lokaal gerepresenteerd is, is gerepresenteerd bij de instantie van de Rol waarvan die Property een eigen- of Aspect property is. Dit betekent dat we in run time de ketting van rolinstanties moeten aflopen tot de instantie die de Property lokaal representeert.

1. Het is geen ‘echt’ type in de zin van Purescript of Perspectives. [↑](#footnote-ref-1)