Prioritering van Property Formats

Ontwerp van een algoritme voor het prioriteren van Fresnel Property Formats .

|  |
| --- |
| **Alex Mekkering** |
| Revisiehistorie |

|  |  |
| --- | --- |
| **Datum** | **Beschrijving** |
| 2015-01-25 | Initiële versie |
| 2015-02-01 | CSS concatenatie + owf:delimiter verwerkt |
| 2015-02-20 | Revisiehistorie bijgewerkt |
| 2015-03-31 | Overige OWF properties toegevoegd |
| 2015-05-12 | Uiteindelijke versie |

# Doel

Het doel van dit document is om een algoritme te ontwerpen voor het prioriteren van Fresnel Property Formats.

# Inleiding

Bij het toepassen van Fresnel Formats op Properties welke geselecteerd zijn door Fresnel Lenses kan het zijn dat er meerdere Fresnel Formats geldig zijn voor een betreffende Property voor die Lens. Deze Fresnel Formats kunnen echter tegenstrijdige presentatie-informatie bevatten waardoor er een noodzaak is tot het eenduidig selecteren van presentatie-informatie.

Om de presentatie-informatie eenduidig te kunnen bepalen uit een collectie van geldige formaten, wordt hiervoor in dit document een algoritme ontworpen.

In dit document wordt, na een korte uitleg over verschillende onderdelen van de Fresnel ontologie, een algoritme bepaald welke uit een collectie van geldige formaten, de presentatie-informatie zo goed als mogelijk eenduidig bepaalt.

# Achtergrond

## Properties op een Lens

De selectie van Properties door een Lens wordt bepaald door de fresnel:showProperties en fresnel:hideProperties predicaten. Het fresnel:showProperties predicaat bepaalt welke properties geselecteerd worden en het fresnel:hideProperties predicaat bepaalt welke properties ge-deselecteerd worden. Hierbij geldt dat fresnel:hideProperties voorrang heeft op fresnel:showProperties; properties welke zowel in fresnel:showProperties als in fresnel:hideProperties vermeld zijn, zijn niet geselecteerd voor die lens.

## Aanduiding van properties

Het aanduiden van properties bij fresnel:showProperties en fresnel:hideProperties kan op twee manieren gebeuren:

1. Een property wordt, middels zijn IRI, rechtstreeks vermeld in één van de lijsten.
2. Er wordt een blank node toegevoegd welke de property beschrijft. Deze blank node bevat minstens de statements ‘rdf:type fresnel:propertyDescription’ en ‘fresnel:property <property IRI>’, waarbij <property IRI> de IRI van de betreffende property is. Daarnaast wordt expliciet een Format gekoppeld aan de betreffende property door een ‘fresnel:use <Format IRI>’ statement, waarbij <Format IRI> de IRI van de betreffende Format is.

## Presentatie-informatie

De presentatie-informatie voor een property, zoals die nu gebruikt wordt binnen de Fresnel Forms Protégé Plugin, kan bestaan uit statements met de volgende predicaten:

* fresnel:label – het label van een property. Kan tekst zijn welke het label definieert, fresnel:none om het label niet te tonen of fresnel:show om het label wel te tonen (default).
* fresnel:value – manier om aan te geven hoe alle waarden van een property getoond moeten worden. Kan één van de volgende waarden hebben:
  + fresnel:image – waarde moet getoond worden als plaatje.
  + fresnel.externalLink – waarde moet getoond worden als klikbare link.
  + fresnel.uri – waarde moet getoond worden als URI.
* fresnel:propertyStyle – de (CSS) stijl van het gebied waarin de property wordt weergegeven.
* fresnel:labelStyle – de (CSS) stijl van het gebied waarin het label van de property wordt weergegeven.
* fresnel:valueStyle – de (CSS) stijl van het gebied waarin de waarde van de property wordt weergegeven.
* owf:delimiter – het scheidingsteken welke gebruikt moet worden om verschillende waarden voor die property in een lijst van elkaar te scheiden.
* owf: autocompleteFromClass – welke klasse(n) gebruikt moet(en) worden voor een object property door een autocompletion functie.
* owf:datatype – welk datatype voor een datatype property geldt.
* owf:isList – of een property meerdere waarden (objecten) kan bevatten voor hetzelfde subject, of niet.
* owf:isMandatory – of de invulling van een property voor een subject verplicht is, of niet.

Een lens kan, naast de CSS stijlen hierboven welke dan als default gelden voor alle properties op die lens, ook nog een fresnel:resourceProperty statement hebben welke de (CSS) stijl bepaalt van het gebied waar de lens in weergegeven wordt. Alle CSS stijlen voor een Lens bevinden zich in een fresnel:Group.

### Gelaagdheid

De presentatie-informatie van een property op een lens kan op verschillende niveaus gedefinieerd zijn. Opgesomd in toenemende mate van specificiteit zijn deze niveaus:

1. Er kunnen generieke Formats gedefinieerd zijn met als fresnel:propertyFormatDomain de betreffende property. Dit generieke Format heeft dan geen fresnel:Group gedefinieerd.
2. Wanneer de betreffende Lens onderdeel is van een fresnel:Group, kan voor die Group bepaald zijn wat de presentatie-eigenschappen zijn voor alle Resources en properties welke door de Lens geselecteerd worden. Een fresnel:Group kan echter slechts stijlinformatie voor properties definiëren, dus geen fresnel:label en fresnel:value.
3. Er kunnen Formats gedefinieerd zijn met als fresnel:propertyFormatDomain de betreffende property én welke onderdeel zijn van één of meer van de groepen waar ook de Lens onderdeel van is. Deze laag kan bestaan uit meerdere sub-lagen: hoe meer fresnel:Group objecten overeenkomen tussen het betreffende Format en de betreffende Lens, hoe specifieker het Format is voor die property en die Lens.
4. Een Format kan expliciet gekoppeld zijn aan de selectie van een property voor een Lens. Dit vindt dan altijd plaats door een fresnel:propertyDescription aanduiding, dus met een Blank Node.

Voor het bepalen van de uiteindelijk presentatie-eigenschappen van een Property voor een Lens is het belangrijk om alle lagen van minst specifiek naar meest specifiek te inventariseren. In de minst specifieke laag kan immers bepaald zijn dat de property een fresnel:propertyStyle en fresnel:valueStyle definitie heeft, terwijl op een specifieker niveau de fresnel:propertyStyle nogmaals gedefinieerd is. De uiteindelijke presentatie-informatie die dan geldt, is de fresnel:valueStyle uit het minst specifieke Format en de fresnel:propertyStyle uit het specifiekere Format.

### Ambiguïteit

Binnen een laag zoals hierboven opgesomd kunnen Formats gedefinieerd zijn die tegenstrijdig zijn. Er kunnen immers best twee Formats zijn die een property op een generieke manier beschrijven. Dit levert echter een niet-consistente Fresnel ontologie op. Volgens (Bizer, Lee, & Pietriga, 2015) dient dan één van de formaten gekozen te worden.

### Meerdere definities binnen hetzelfde Format

Voor een Format kunnen ook meerdere statements voor dezelfde property gedefinieerd zijn.

Als voorbeeld kan de property owf:autocompleteFromClass meerdere keren gedefinieerd zijn voor een Format. Dit houdt dan in dat de autocompletion functionaliteit een keuze moet bieden uit de vereniging van alle instanties van de opgegeven klassen.

Het algoritme kan hier generiek ondersteuning voor bieden door voor elke laag, per property, een lijst van mogelijke waarden terug te geven.

### CSS

De waarden van de CSS stijl properties (fresnel:\*Style) bestaan uit CSS welke meerdere CSS attributen kan bevatten. Zo kan in bijvoorbeeld voor fresnel:propertyStyle de waarde “color:purple; font-weight:bold;” gedefinieerd zijn, wat inhoud dat de tekst binnen een property de kleur paars moet krijgen en vet gedrukt moet zijn.

Dezelfde styling property, in verschillende lagen gedefinieerd, dus met verschillende specificiteit, kan dus CSS attributen bevatten welke elkaar niet of slechts gedeeltelijk overlappen. Om hiervoor ondersteuning te kunnen bieden dienen dezelfde styling properties van verschillende specificiteit zodanig samengevoegd te worden dat alle informatie erin behouden blijft, evenals de specificiteit ervan.

CSS (Bos, Håkon, Lilley, & Jacobs, 2015) biedt hier ondersteuning voor doordat attributen die later zijn gedefinieerd (achteraan in de string staan) per definitie meer specifiek zijn dan attributen die eerder zijn gedefinieerd. Zo geldt bij de CSS waarde “color:purple; color:yellow;” dat de kleur van het element geel moet worden.

# Oplossing

Om de meest specifieke presentatie-informatie voor een Property voor een Lens te bepalen is het volgende bedacht:

* De gelaagdheid, zoals hierboven opgesomd, wordt geïmplementeerd door Formats en Groups prioriteiten toe te kennen waarbij 0 het minst specifiek is en een hoger nummer een steeds specifieker Format of Group betreft,
* Het aantal prioriteiten is gelijk aan 3 (de 1e, 2e en 4e laag) + het aantal fresnel:Group statements voor de betreffende Lens. Omdat de doorsnede van twee verzamelingen niet meer elementen kan hebben dan het minimum van het aantal elementen van de verzamelingen kan een Format niet meer overeenkomstige fresnel:Group statements hebben met de Lens als dat de Lens zelf fresnel:Group statements heeft.
* Voor elke prioriteit wordt een lijst voor Formats en Groups gedefinieerd. Dit om de eigenschappen van Formats die in principe tegenstrijdig zijn, maar andere presentatie eigenschappen beschrijven (dus eigenlijk toch niet tegenstrijdig zijn), toch correct mee te kunnen nemen. Formats en Groups worden aan één van deze lijsten toegevoegd, op basis van hun prioriteit.

De prioriteit van Formats en Groups wordt als volgt bepaald:

* Een generiek Format met als fresnel:propertyFormatDomain <property IRI> en geen fresnel:Group krijgt prioriteit 0.
* Een fresnel:Group waarvan de Lens onderdeel is, krijgt prioriteit 1.
* Een Format welke een fresnel:propertyFormatDomain <property IRI> heeft en fresnel:Group statements heeft die overeenkomen met die van de betreffende Lens, krijgt als prioriteit 1 + (het aantal fresnel:Group statements welke overeenkomen met die van de Lens).
* Een Format welke expliciet gekoppeld is aan een Property (middels fresnel:propertyDescription) krijgt de hoogste prioriteit (3 + (het aantal fresnel:Group statements welke overeenkomen met die van de Lens) -1).

Per property voor een Lens worden de presentatiegegevens als volgt bepaald:

1. Het algoritme wordt geïnitialiseerd:
   1. Er wordt gekeken of er voor de betreffende property een propertybeschrijving aanwezig is voor de Lens (blank node met fresnel:propertyDescription).
   2. Het aantal prioriteiten wordt bepaald op basis van het aantal fresnel:Groups waarvan de Lens onderdeel is.
   3. Er wordt een geprioriteerde Lijst van lege verzamelingen opgebouwd waarbij de verzameling met de laagste prioriteit (0, minst specifiek) als eerst gedefinieerd is.
2. Alle fresnel:Groups worden in de verzameling met prioriteit 1 geplaatst.
3. Voor alle Formats wordt nu het volgende gedaan:
   1. De prioriteit wordt bepaald.
   2. Indien de prioriteit bepaald kon worden (Het Format was van toepassing op de betreffende Property op de Lens), wordt het Format toegevoegd aan de verzameling met de juiste prioriteit.
4. Vervolgens wordt de geprioriteerde lijst van verzamelingen in toenemende prioriteit doorlopen om de hierin verzamelde meest-specifieke presentatie-gegevens voor de Property op de Lens te bepalen.
   1. Voor styling properties worden de waarden (CSS) achteraan toegevoegd aan eventueel eerder gevonden waarden
   2. Voor overige properties geldt dat gegevens welke in een eerdere verzameling (met lagere prioriteit) aanwezig waren worden overschreven. Hierbij worden de complete lijsten van property waarden overschreven met de property waarden welke voor het betreffende Format gelden.
   3. Voor de verzameling met Groups (prioriteit 1) worden de niet-stijl properties niet meegenomen.

Alle meest specifiek presentatiegegevens worden verzameld in een instantie van een klasse welke is afgeleid van een HashMap welke als sleutel een property heeft en als waarde een lijst van propertywaarden. Er wordt gebruik gemaakt van een afgeleide klasse van HashMap om een ‘pick’ methode te kunnen definiëren welke van de lijst van waarden één arbitraire waarde teruggeeft. Naast de standaard interface van een HashMap, wordt hiermee een API geboden waarmee voor eenvoudige properties, waarvan verwacht wordt dat er niet meer dan één waarde gedefinieerd is, de waarde eenvoudiger bepaald kan worden.

# Bibliografie

Bizer, C., Lee, R., & Pietriga, E. (2015, 01 25). *Fresnel - Display Vocabulary for RDF*. Opgehaald van W3C: http://www.w3.org/2005/04/fresnel-info/manual/

Bos, B., Håkon, W., Lilley, C., & Jacobs, I. (2015, 02 01). *Cascading Style Sheets, level 2 CSS2 Specification*. Opgehaald van W3C: http://www.w3.org/TR/REC-CSS2/