



IDsw

InformatieDesk standaarden Water

Wijzigingsvoorstel Ontwerp en implementatie “Aquo-catalogus”

Auteur: IDsw
Datum: 18-01-2010
Versie: 0.4c
Kenmerk: W-0910-0010

Documentbeheer

Wijzigingshistorie

Datum	Versie	Auteur	Wijziging
28-09-2009	0.1	H-J. Lekkerkerk	Eerste versie nav Functioneel ontwerp
08-10-2009	0.2	H-J. Lekkerkerk	Opmerkingen verwerkt
08-01-2010	0.3	H-J. Lekkerkerk	Definitief concept opstellen obv rapportage van Alterra
21-01-2010	1.0	H-J. Lekkerkerk	Opmerkingen verwerkt

Review

Datum	Versie	Reviewer	Functie
4-10-2009	0.1	Hinne Reitsma	Projectleider Standaarden
15-01-2010	0.3	Hinne Reitsma	Projectleider Standaarden
11-01-2010	0.3	Wilbert Vos	Specialist Standaarden

Controle en vrijgave

Datum	Versie	Controleur	Functie

Literatuurbronnen

- Concept eindrapport Objectencatalogus, Alterra, December 2009
- Richtlijn voor het opstellen van een wijzigingsvoorstel op de uitwisselmodellen, IDSW, maart 2006
- Aquo standaard dd juli 2009
- NTA 8611:2003
- Ontwerp BRON thesaurus
- Plan van eisen Domeintabellen services

Inhoudsopgave

1. Motivatie	5
1.1 Aanleiding	5
1.1.1 Achtergrond	5
1.1.2 Doel	6
1.2 Business Case	7
1.2.1 Voordelen	7
1.2.2 Impact	7
1.2.3 Afbakening	7
1.2.4 Beheer	8
2. Huidige Aquo standaard in relatie tot een ‘Objecten catalogus’	11
2.1 Classificatie van de huidige produkten	11
2.1.1 Aquo-lex	11
2.1.2 IMWA	11
2.1.3 UM Aquo / LM Aquo	11
2.2 Relaties tussen de huidige onderdelen van de standaard	13
3. Voorbeeld en werking Objectencatalogus	16
3.1 Introductie	16
3.2 Voorbeeld op hoofdlijnen	16
3.3 Detailvoorbeelden	18
3.3.1 Entiteit versus Concept en subklassen	18
3.4 Van objectencatalogus naar een informatiemodel	21
3.4.1 Voorbeeld: Kunstwerk - afsluitemiddel / wijze	21
3.4.2 Aanzet tot afleidingsregels voor de objecten catalogus	24
4. Het model van de objectencatalogus	25
4.1.1 Objectklassen in de objectencatalogus	25
4.1.2 Properties van de objectencatalogus	27
4.1.3 Meertaligheid	28
4.1.4 Properties per objectklasse	29
Bijlage A Theorie en concepten achter de objectencatalogus	33

1. Motivatie

1.1 Aanleiding

1.1.1 Achtergrond

De Aquo-standaard bestaat uit aan elkaar gerelateerde onderdelen die gericht zijn op definities van gegevens, het uitwisselen van gegevens, de opslag van gegevens en de inwinning, presentatie en verwerking van gegevens.

Voor elke standaard zijn er - naast sec de standaard - aanvullende documenten beschikbaar om de toepassing van de standaard te ondersteunen (praktijkrichtlijnen, voorbeeldbestanden etc.). In het kader van de objecten catalogus zijn de volgende onderdelen van de Aquo-standaard relevant.

Functie	Onderdelen Aquo-standaard:	Komt voort uit 'bruidsschat'
Definitie van gegevens	Aquo-lex; Aquo domeintabellen.	Adventus gegevenswoordenboek, CIW gegevenswoordenboek, Omega woordenboek
Uitwisseling van gegevens	InformatieModel Water - IMWA; UitwisselModel Aquo - UM Aquo;	IMWA
Opslag van gegevens	Logisch Model Aquo – LM Aquo.	Logisch Model Adventus

Eén van de belangrijkste opdrachten die IDSW bij zijn oprichting meekreeg was het integreren van de 'bruidsschat', bestaande uit de CIW gegevensstandaard, het Adventus stelsel, IMWA en Omega Woordenboek tot een enkele, integrale standaard.

Aan de zijde van de gegevensdefinitie is dit traject inmiddels nagenoeg afgerond met de opschoning van de domeintabellen en de publicatie van een enkel, geïntegreerd Aquo-lex woordenboek.

Qua standaard zijn zowel de domeintabellen en Aquo-lex inmiddels zo goed als af, maar op de beschikbaarstelling en het gebruiksgemak valt nog wel wat af te dingen. De ontsluiting als 'klassiek' woordenboek / termenlijst van Aquo-lex of de domeintabellen is redelijk tot goed geregeld. De metadata (waarom, door wie, wanneer etc.) zijn aanzienlijk minder goed geregeld. Het wordt problematisch als het gaat om verbanden tussen domeinwaarden en definities onderling en tussen elkaar. Deze relaties zijn momenteel niet of nauwelijks voor de gebruiker terug te vinden.

Waar het gaat om de informatiemodellen is de integratie van een ander niveau. Het nadeel van informatie modellen, of het nu gaat om modellen voor de procesmatige opslag (LM Aquo) of de uitwisseling (IMWA / UM Aquo), is dat deze voor een specifiek doel en/of organisatie worden ontwikkeld. Afhankelijk van het proces waarvoor de informatie nodig is zal voor een bepaalde modellering worden gekozen. Indien een model voor meerdere processen tegelijk wordt ontwikkeld zullen er ofwel dubbelingen ontstaan (vergelijkbare objecten / relaties) ofwel zullen er concessies aan het model gedaan moeten worden.

Indien naar de verschillende modellen gekeken wordt zal het opvallen dat de gebruikte begrippen / termen, attributen en domeintabellen zeer goed overeenkomen; het is mogelijk om verliesloos vanuit het LM Aquo naar IMWA / UM Aquo te vertalen. Andersom niet; dit is een gevolg van het ontbreken van bepaalde attributen in IMWA / UM Aquo (proceskeuze).

Kortom, feitelijk is de definitie van de informatie in de verschillende modellen identiek (entiteiten / klassen, attributen, definities en domeintabellen). Het enige verschil is de procesblik (relaties, exacte keuze attributen) waarmee het model is samengesteld.

1.1.2 Doel

Doel van dit wijzigingsvoorstel is het realiseren van een betere integratie van de onderdelen van de Aquo-standaard. Dit wordt gerealiseerd door enerzijds het toevoegen van semantiek aan Aquo-lex, anderzijds gaat het om het gezamenlijk deel van de informatiemodellen en Aquo-lex als zodanig te duiden en als apart product beschikbaar te stellen.

Een bijkomstig doel van dit voorstel is het verder beperken van de beheerslast en het zorgen voor een betere beschikbaarstelling van met name de inhoud van het huidige Aquo-lex.

1.2 Business Case

1.2.1 Voordelen

Door de ontwikkeling van de objecten catalogus ontstaat in feite een ingrediënten lijst voor de systeemontwikkelaar / gegevensbeheerder. Immers, de kern van de Aquo-standaard zal zich hiermee gaan beperken tot de objecten catalogus. Deze legt vast welke objecten met welke definitie en eigenschappen binnen het waterbeheer worden onderscheiden zonder de manier van modelleren van deze zaken vast te leggen.

Het voordeel van de ontwikkeling van de objectencatalogus voor de 'eindgebruiker' van de Aquo standaard (de beleidsmedewerker, beheerder etc) is het voordeel van de catalogus dat begrippen in hun onderlinge relaties ontsloten kunnen worden op een uniforme wijze. Daarnaast biedt de gekozen methode ook verdere integratie in het World Wide Web aan (Web 3.0) door de keuze van internationale standaarden bij de ontwikkeling. Een mogelijke toepassing hiervan is de ontwikkeling van services die het de gebruiker mogelijk maken termen en definities op 'afstand' op te vragen en toe te passen in publicaties. Ook het vertalen van termen via een geautomatiseerde service kan op basis van deze objectencatalogus eenvoudig bereikt worden.

1.2.2 Impact

Applicaties en informatie modellen die in de toekomst 'Aquo-proof' willen zijn zullen daarmee volledig conform moeten zijn met de Aquo-catalogus én de Aquo-domeintabellen (identieke naam, definitie en eigenschappen uit de lijst), m  r ze hoeven niet meer volledig conform een informatie model opgebouwd te zijn.

Natuurlijk zullen de informatie modellen wel een belangrijke rol blijven spelen in de interoperabiliteit tussen software producten. Het staat (een groep van) leveranciers echter vrij om, op basis van de Aquo catalogus, een eigen uitwisselformaat of database te ontwikkelen. Voor bepaalde trajecten die een enkele applicatie of (groep) waterbeheerder(s) overstijgen (bv KRW uitwisseling, Waterwet etc) neemt IDSW de rol in van leverancier in en zal daarvoor specifieke uitwisselmodellen beschikbaar stellen.

Maar ook voor de 'gewone' eindgebruiker zal informatie duidelijker te vinden zijn en wordt het makkelijker om verbanden tussen begrippen te leggen. Bij implementatie van de voorgestelde functionaliteit wordt het mogelijk voor waterbeheerders om begrippen op eigen (interne) websites direct te koppelen aan definities uit Aquo-lex. Hierdoor hoeft geen eigen beheer meer te worden gevoerd en is ook zo'n website direct Aquo-conform.

Eenzijds is de impact van de 'Aquo-catalogus' dan ook groot; het huidige Aquo-lex wordt vervangen door een nieuw, overkoepelend product waaraan iedere 'Aquo gebruiker' zal moeten voldoen om 'Aquo-proof' te zijn. Anderzijds zullen bestaande modellen en onderdelen qua inhoud volledig in stand blijven (maar in vorm veranderen).

1.2.3 Afbakening

Dit voorstel beperkt zich tot de volgende onderdelen van de standaard (en integreert deze tot de objectencatalogus):

- Aquo-lex: volledig

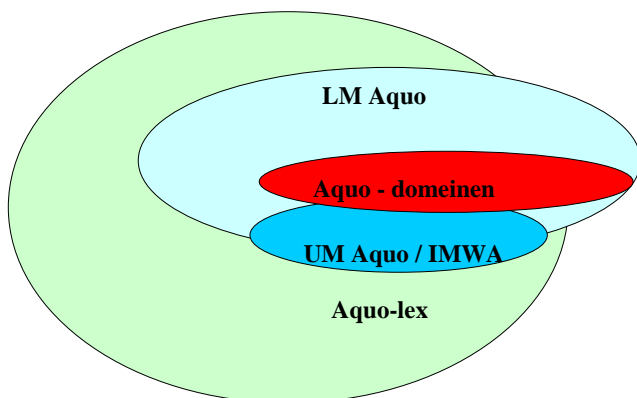
- IMWA: klassen / attributen en definities
- UM Aquo: klassen / attributen en definities
- LM Aquo: entiteiten / attributen, definities en visualisatie (beschrijving hoe een entiteit gevisualiseerd wordt).

In de objectencatalogus worden geen relaties / associaties overgenomen uit de diverse modellen anders dan generalisatie / specialisatie (sub / superklasse). De afbakening in de modellering is weergegeven tabel 2 van hoofdstuk "4.1.4 Datamodel totaal", waarin staat vermeld wat mogelijk is en daarmee toegestaan. De objecten catalogus is gedefinieerd vanuit het objecten catalogus oogpunt en niet vanuit de bestaande standaarden.

Naast de objectencatalogus zullen specifieke informatiemodellen blijven bestaan voor specifieke doelen (daar waar gemeenschappelijke relaties ook van belang zijn).

Aquo-domeinen

In dit voorstel worden de Aquo-domeinen verder buiten beschouwing gelaten. Wel kunnen de waarden in Aquo-domeinen verwijzen naar onderdelen van de Aquo-catalogus waardoor waarden uit de Aquo-domeinen van een eenduidige definitie voorzien kunnen worden.



Figuur: Huidige overlap tussen de diverse onderdelen van de standaard qua begrippen / definities

In bovenstaande figuur is te zien dat er een klein deel van het LM Aquo niet is opgenomen in Aquo-lex. Dit is het gevolg van het ontbreken van 'fatsoenlijke' definitie zoals bij soort gemaal. Vaak gaat het daarbij om attributen met algemene eigenschappen (lengte / breedte etc.) of classificaties. Een belangrijk punt voor de objectencatalogus is dan ook de koppeling tussen domeintabellen en attributen. In veel gevallen vervangt een domeintabel met daarin 'typeringen' in feite een hiërarchie.

1.2.4 Beheer

Het beperken van het beheer wordt bereikt door het beheren van het huidige Aquo-lex verder te automatiseren. Momenteel wordt dit beheer gevoerd in een Excel spreadsheet. Daarnaast zal het beheer van IMWA, UM Aquo en LM Aquo wijzigen. Definities uit deze modellen en beschrijvingen van entiteiten / attributen worden met dit voorstel opgenomen in de Objecten catalogus en daarmee niet meer onderhouden in verschillende tools (Enterprise Architect en Oracle Designer).

Aangezien de relaties / associaties uit de bestaande informatiemodellen niet overgenomen worden in het voorstel voor de Objecten catalogus (vanwege de veelheid en vaak proces specifieke eigenschappen van deze relaties) blijft het beheer van de modellen IMWA / UM Aquo en LM Aquo noodzakelijk. Dit beheer blijft echter beperkt tot de diagrammen met daarin de relaties (en eventueel technische eigenschappen) tussen de diverse klassen / entiteiten. Voor de inhoud wordt verder integraal verwezen naar de objectencatalogus.

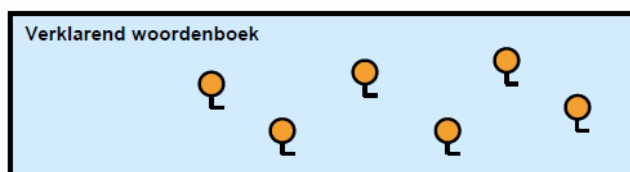
2. Huidige Aquo standaard in relatie tot een ‘Objecten catalogus’

2.1 Classificatie van de huidige producten

Om een goed beeld te schetsen van de huidige status van de Aquo standaard en de relatie daarvan met een te ontwikkelen ‘Objecten catalogus’ is het goed te kijken naar de invalshoek bij de huidige onderdelen van de Aquo-standaard. Hiervoor is gebruik gemaakt van de NTA 8611 - 2003 voor Objecten bibliotheken.

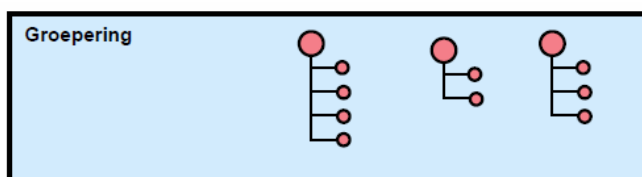
2.1.1 Aquo-lex

Gezien de opzet kan Aquo-lex (en de relaties met de overige delen van de standaard) op dit moment het beste gezien worden als een ‘verklarend woordenboek’. Volgens NTA 8611 - 2003 heeft deze de functie van het geven van *“de (lexicale) definitie van een object”*. Hier kan de vergelijking worden getrokken met een woordenboek als de ‘Van Dale’. Het verklarend woordenboek verklaart of beschrijft de objecten door een definitie.”



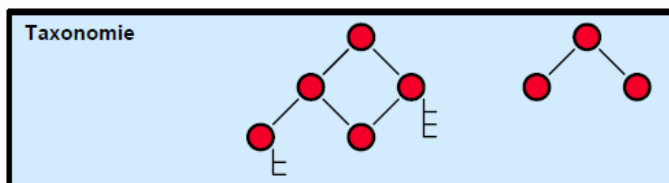
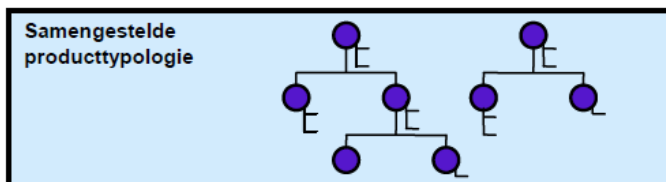
2.1.2 IMWA

Qua opzet komt IMWA nog het meest overeen met een ‘Groepering’ conform NTA 8611 - 2003. *“De laatste functionele invalshoek heeft vooral een ondersteunend, context classificerend karakter. De invalshoek bevat objecten die zijn bedoeld om objecten uit andere functionele invalshoeken te classificeren of groeperen. Zo kunnen objecten uit de vorige invalshoeken worden geclassificeerd naar de discipline waarin ze primair worden toegepast en kunnen kenmerken worden gerelateerd aan hun toepassingsdomein. Ook kunnen groepen van kenmerken van objecten worden gedefinieerd waarmee ‘views’ op delen van de informatie over objecten mogelijk wordt”*



2.1.3 UM Aquo / LM Aquo

Het UM Aquo / LM Aquo zijn lastig in te delen. Enerzijds vertonen ze kenmerken van de ‘groepering’ zoals in IMWA, anderzijds vertonen deze modellen veel overeenkomst met de ‘Samengestelde Produkttypologie’ en de Taxonomie conform NTA 8611 - 2003: *“Hierbij ligt de nadruk op de specifieke opbouw van objecten uit andere objecten. Deze functionele invalshoek kent veelal zijn toepassing binnen bedrijven en bij integratie van gegevens uit verschillende bronnen/invalshoeken, zoals integratie over een bedrijfskolom”*

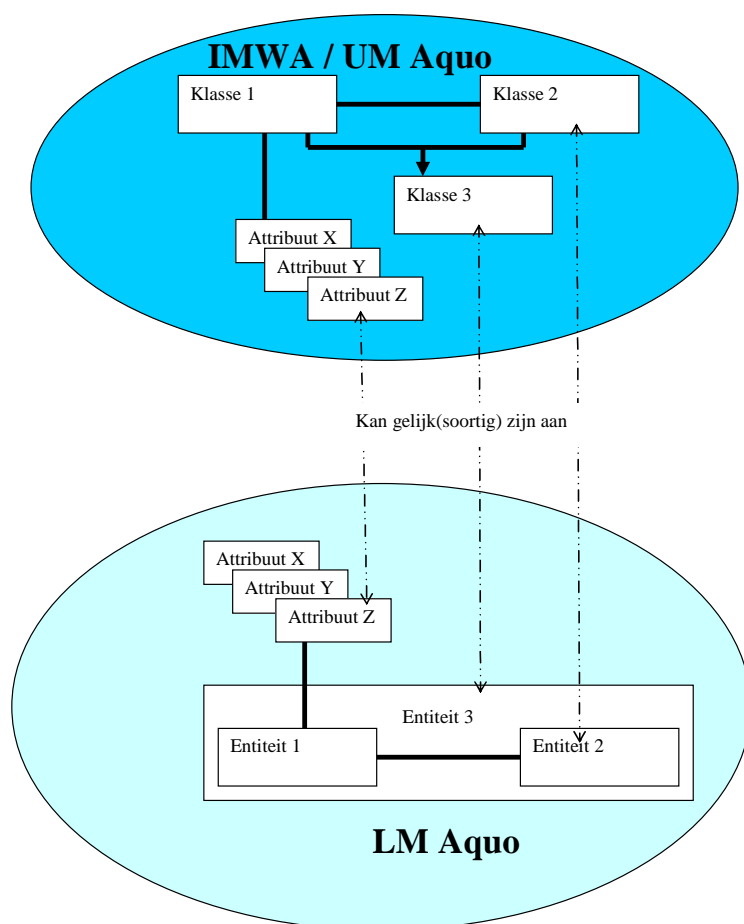


Zo kunnen zogenoemde kenmerken worden vastgelegd die een object typeren (zie harkjes in de figuur). Door het koppelen van deze kenmerken aan een object, weet de gebruiker wat dit object van de andere objecten onderscheidt. Zo heeft bijvoorbeeld een kogel een vormeigenschap met als waarde 'bolvormig'. Een nog belangrijker mechanisme voor het vastleggen van de betekenis betreft de verschillende relaties tussen de objecten (in het bijzonder specialisatierelaties). Deze relaties beschrijven hoe een bepaald object zich verhoudt tot een ander object.

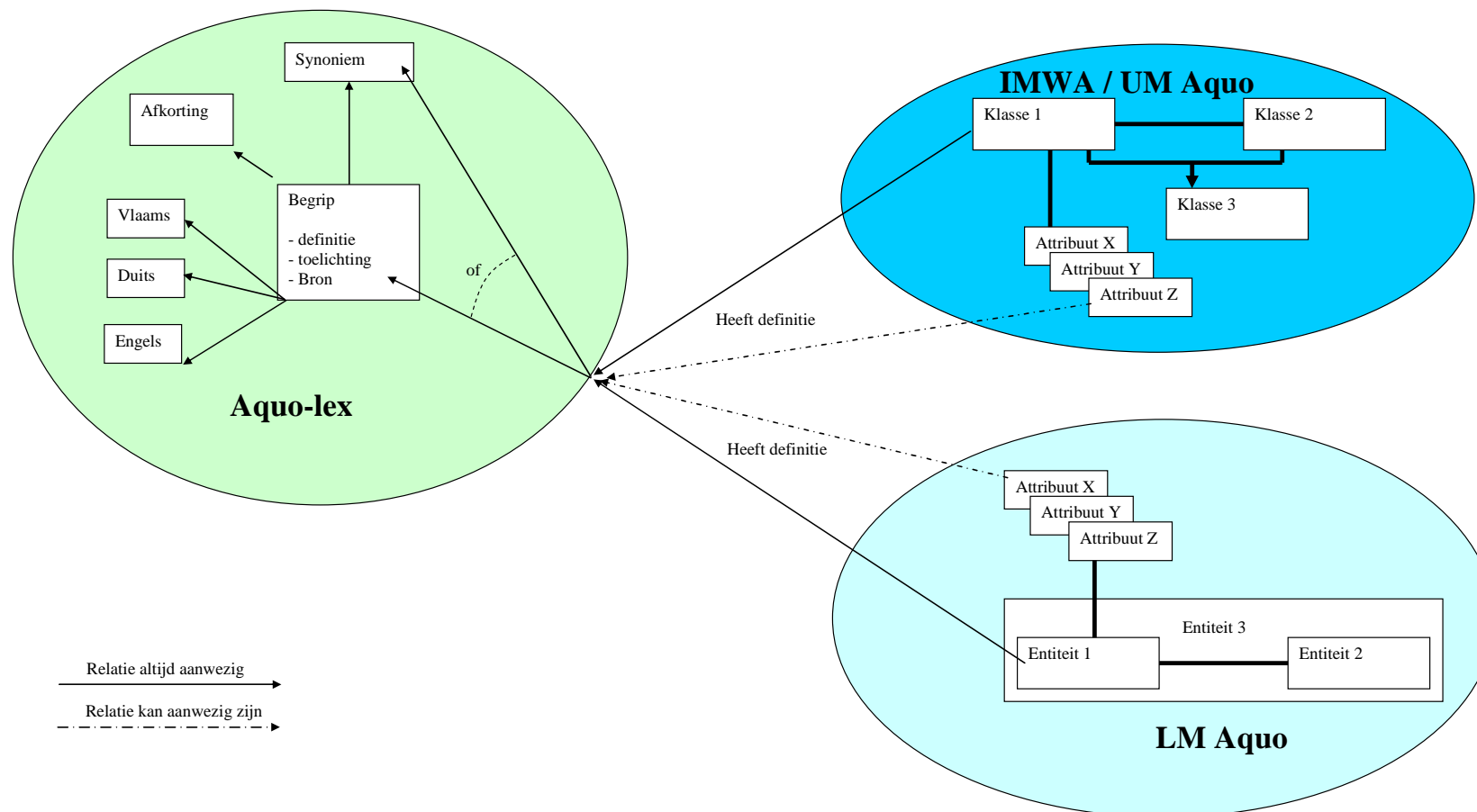
Functioneel gezien biedt vooral taxonomie de basis om tussen partijen te komen tot een gemeenschappelijke definitie van de betekenis van een object. Daarnaast is deze functionele invalshoek geschikt als zoekmechanisme (eventueel voor de geschetste innovatie).

2.2 Relaties tussen de huidige onderdelen van de standaard

De diverse onderdelen van de Aquo-standaard staan niet op zichzelf; begrippen uit Aquo-lex vormen de basis voor entiteiten in het LM Aquo en klassen in UM Aquo / IMWA. Tussen LM Aquo en IMWA / UM Aquo is er afstemming van klassen en entiteiten en worden dezelfde of vergelijkbare domeintabellen gebruikt.



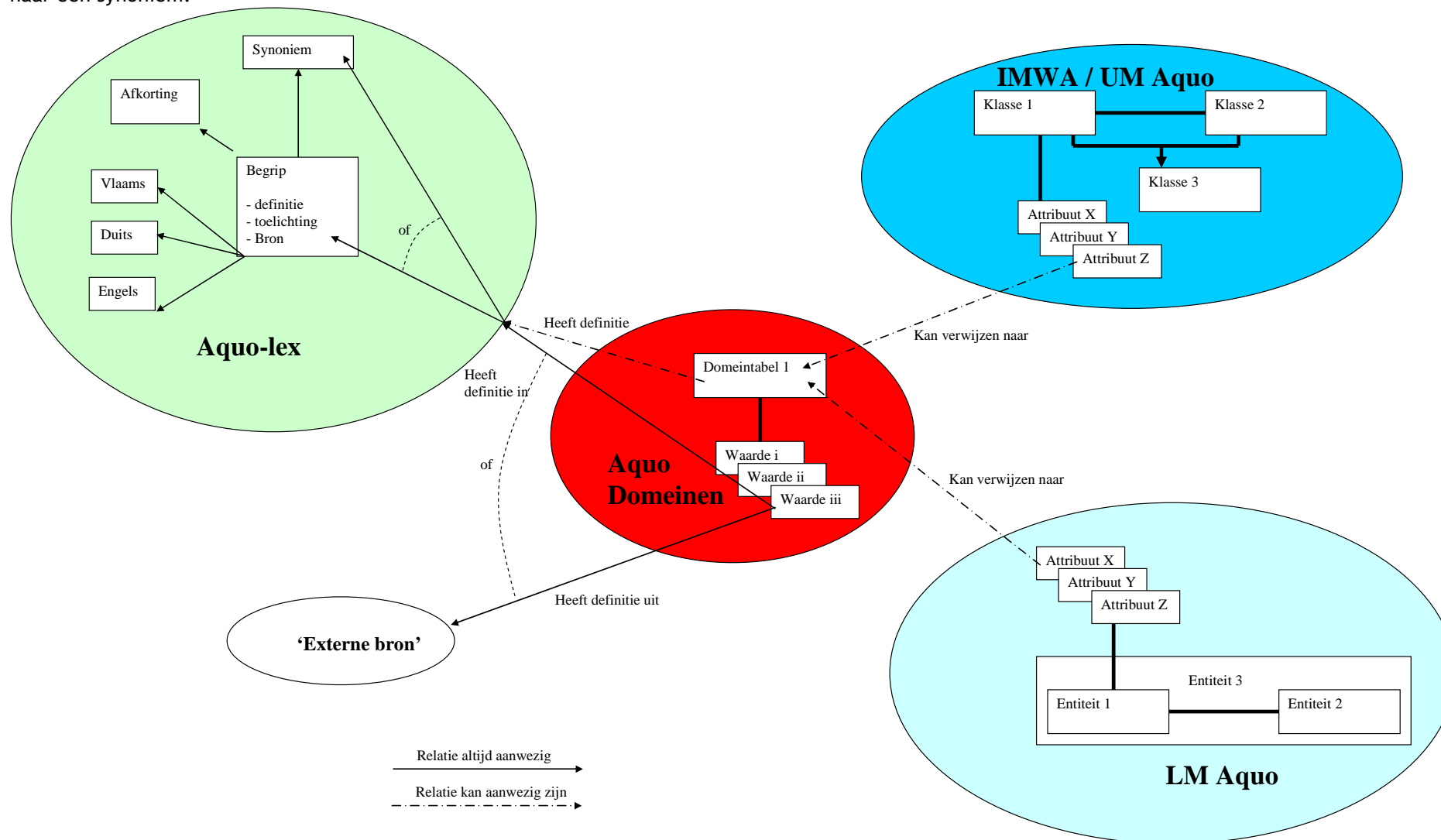
Figuur: relaties tussen IMWA / UM Aquo en het LM Aquo.



Figuur: Relatie tussen de verschillende onderdelen via attributen / klassen / entiteiten

Alle klassen en entiteiten hebben een definitie die is vastgelegd in Aquo-lex. Dit kan via het ‘voorkeursbegrip’ zijn, maar ook via een synoniem. De meeste attributen hebben ook een definitie; uitzondering hier zijn de ‘classificaties’ (soort x, type y etc) en algemene begrippen (lengte, breedte etc). Attributen uit zowel het LM Aquo als IMWA / UM Aquo kunnen verwijzen naar een specifieke domeintabel. De - ontsluiting van - de domeintabellen vallen buiten de scope van de objectencatalogus behalve voor wat betreft de definitie van de domeintabel en de definities van domeinwaarden. Deze definities zijn nu ofwel in Aquo-lex ofwel in een ‘externe bron’ (bv CAS nummers, literatuur bij TWN etc) vastgelegd. De te ontwikkelen domeintabellen service zal daarvoor

verwijzen naar de objectencatalogus. In de meeste gevallen verwijst een domeinwaarde / tabel naar een ‘voorkeursbegrip’, maar in sommige gevallen ook naar een synoniem.



Figuur: Relatie tussen definitie / begrippen via domeinwaarden en domeintabellen

3. Voorbeeld en werking Objectencatalogus

3.1 Introductie

In dit hoofdstuk zijn een aantal concrete uitwerkingen van het voorstel van de objectencatalogus opgenomen. In deze voorbeelden wordt gebruik gemaakt van een datamodel pas later wordt toegelicht. Voor meer detail informatie omtrent het gebruikt datamodel, zie hoofdstuk 4. Voor meer informatie over de gebruikte technologie, zie Bijlage A.

3.2 Voorbeeld op hoofdlijnen

Op hoofdlijnen kan de werking / vertaling vanuit de verschillende (bestaande) modellen en onderdelen van de Aquo-standaard naar de Objectencatalogus volgens onderstaand voorbeeld plaats vinden.

In IMWA en in het LM Aquo is de klasse 'Kunstwerk' gedefinieerd. In IMWA is er, door het doel van het model, voor gekozen om via een domeintabel 'type kunstwerk' aan te geven om welke type kunstwerk het precies gaat. Daarin zijn waarden als 'gemaal'; 'brug' en 'sluis' opgenomen. Sommige kunstwerken kennen, in de domeintabel, weer een onderverdeling naar een nader kunstwerktype.

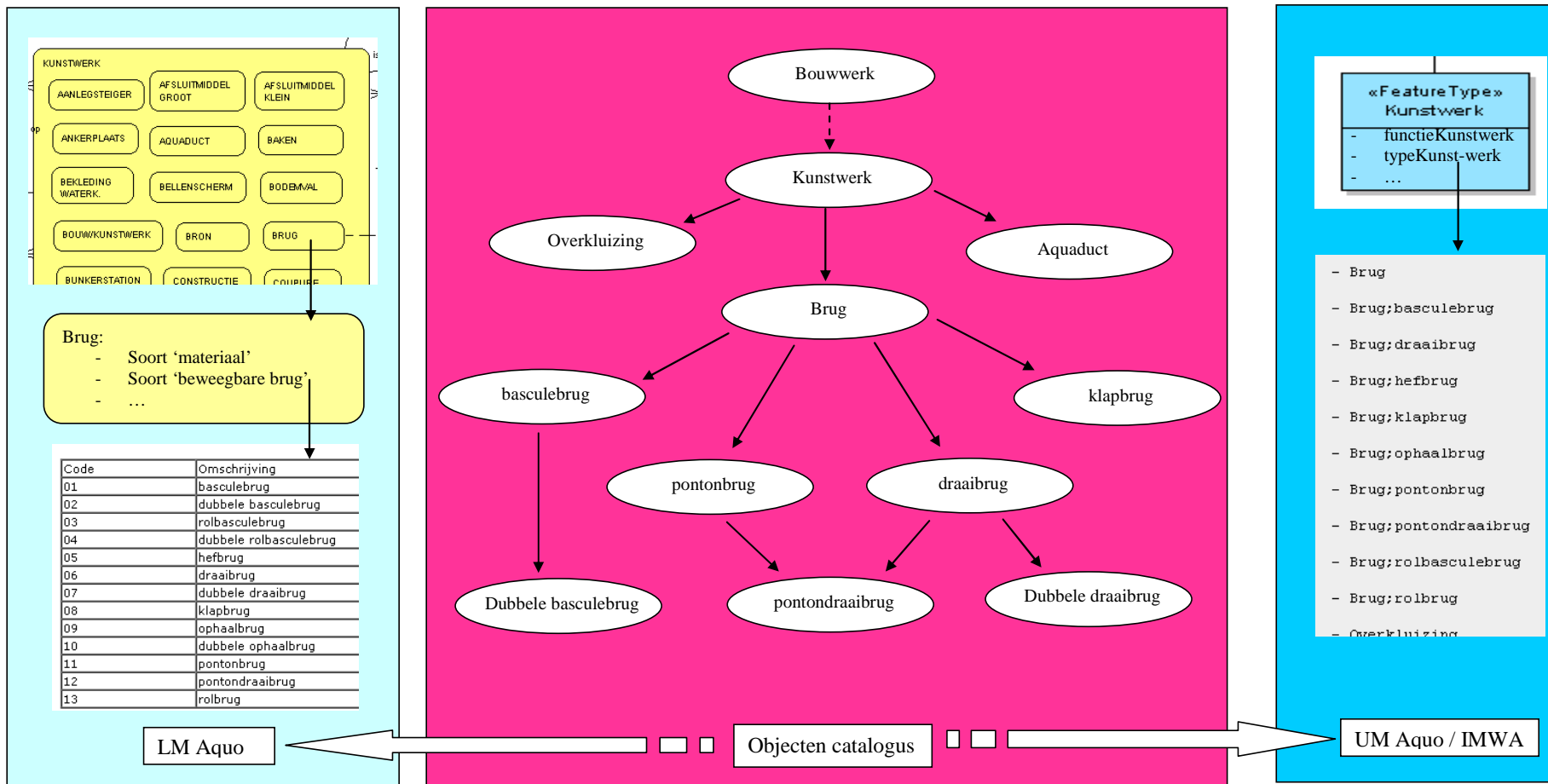
Dit soort modellering is prima als men niet verder wil gaan dan het classificeren van verschillende soorten kunstwerken zonder daar aanvullende informatie bij vast te leggen per (sub)type kunstwerk (bijvoorbeeld voor het afbeelden op een kaart).

In het LM Aquo is de entiteit 'kunstwerk' de superentiteit (generalisatie) van o.a. de entiteiten 'gemaal', 'brug' en 'sluis'. De entiteit 'brug' heeft een eigen, specifieke, set eigenschappen (attributen) waaronder een attribuut 'soort beweegbare brug' waarin de diverse typen beweegbare bruggen verder zijn gedetailleerd.

Dit soort modellering is prima als het gaat om het vastleggen van specifieke eigenschappen per (sub)type voor bijvoorbeeld beheerprocessen.

Lexicaal (voor bijvoorbeeld een objectencatalogus) zijn beide opties gelijkwaardig; de manier van modelleren (subentiteiten / klassen vs domeintabel) wordt in dit verband alleen bepaald door het verdere gebruik van het model (wel attributen per subtype = subklasse of niet = domeintabel).

In een lexicaal model maakt het niet meer uit welke modelleer optie gekozen wordt; ieder begrip is hier een eigen object met een eigen definitie (en mogelijk ook eigen eigenschappen). Door deze manier van vastleggen worden 'kromme' situaties die het gevolg zijn van een modelleertaal zoveel mogelijk voorkomen. Als voorbeeld de domeinwaarde 'pontondraaibrug' - *"Brug bestaande uit op het water drijvende pontons, welke bij opening om een verticale as worden verplaatst."* die nu als specifieke waarde is opgenomen, maar feitelijk een combinatie is van een pontonbrug - *"Brug bestaande uit op het water drijvende pontons"*. en draaibrug - *"Brug draaibaar om één verticale as."*



figuur: Modelling van brug in het LM Aquo (links) en in IMWA / UM Aquo (rechts). Daartussen een (mogelijk) lexicaal model (iedere pijl stelt een lagere orde begrip voor).

3.3 Detailvoorbeelden

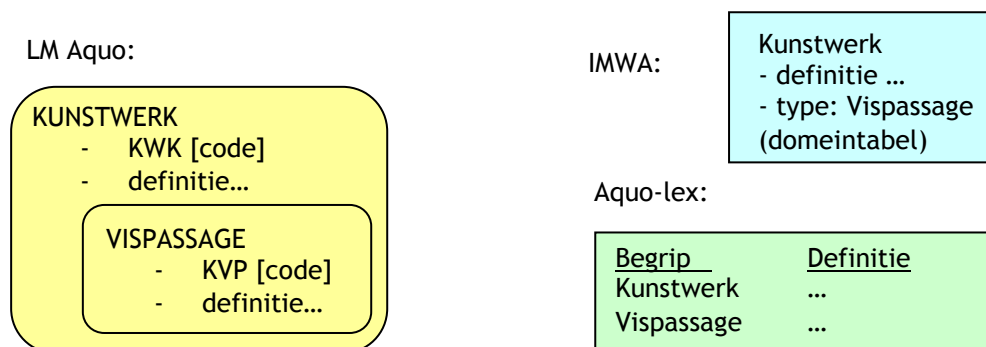
De in deze paragraaf beschreven voorbeelden (en afbeeldingen) geven enkele uitwerkingen van hoe objecten van de domeintabellen en LM Aquo en IMWA / UM Aquo worden gekoppeld aan Aquo-lex begrippen in de objectencatalogus.

Bij deze voorbeelden is uitgegaan van het ontwerp uit het vorige hoofdstuk.

3.3.1 Entiteit versus Concept en subklassen

In dit voorbeeld wordt getoond hoe een (deel) van de huidige modellering omgezet kan worden van de huidige modellen en onderdelen van de standaard naar een geïntegreerde objectencatalogus.

Het gaat hierbij om de volgende onderdelen van de standaard:



In het LM Aquo is Vispassage gedefinieerd als sub-entiteit van de entiteit Kunstwerk. Beide hebben een eigen code, definitie (en attributen).

In IMWA is de vispassage een type kunstwerk (via een domeintabel - typeKunstwerk) en heeft de klasse kunstwerk waar deze onder valt een eigen definitie (en attributen).

Tot slot zijn in Aquo-lex zowel de begrippen 'Vispassage' als 'Kunstwerk' opgenomen met ieder een eigen definitie, toelichting etc.

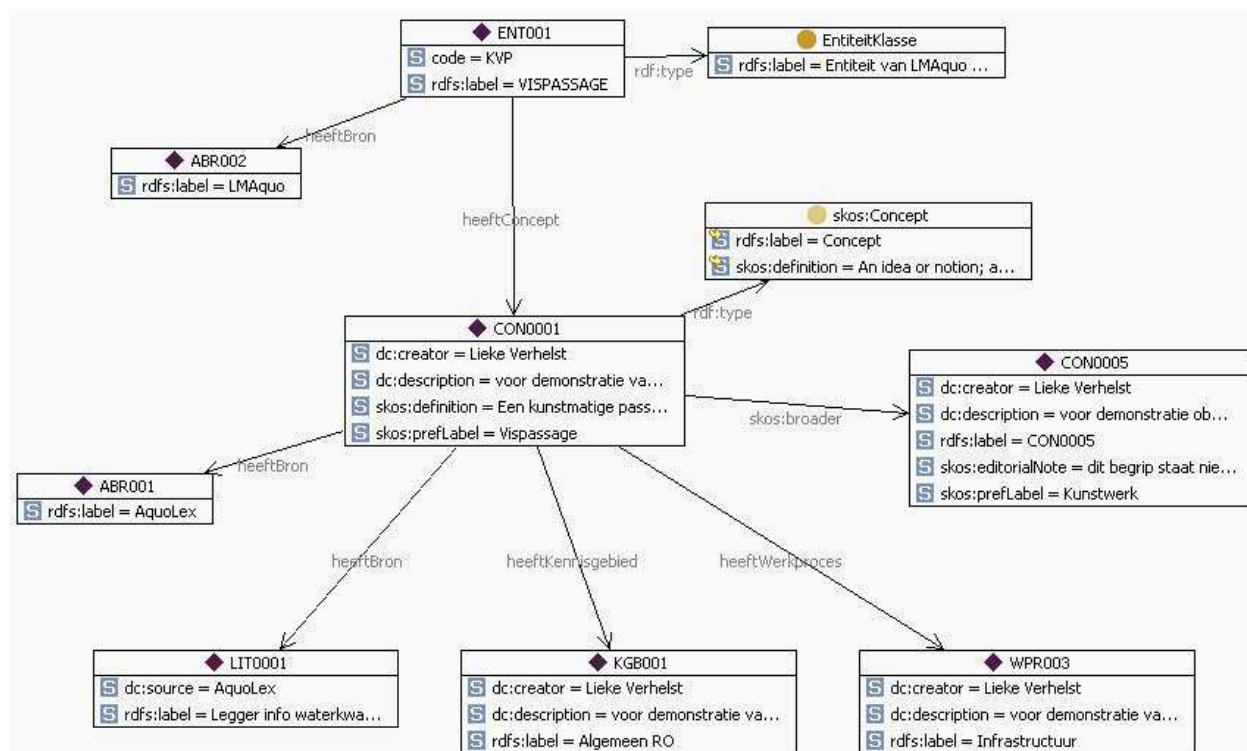
In onderstaande figuren zijn de verschillende stappen en het eindresultaat van de modellering in de (toekomstige) objectencatalogus geïllustreerd waarbij ook hier een formele koppeling wordt gelegd tussen de verschillende onderdelen.

Koppeling entiteit - concept

In deze stap wordt een object VISPASSAGE van het type 'EntiteitKlasse' (ENT001) gekoppeld aan een object Vispassage van het type 'concept' (CON0001). De bron voor de entiteit is het LM Aquo (ABR002). Het concept 'Vispassage' is weer een narrower term (subklasse) van het concept 'Kunstwerk' (CON0005).

Sub-superklasse relaties worden in de objectencatalogus dus niet vastgelegd op entiteit niveau maar op concept niveau. De objectencatalogus wordt daarmee niet afgeleid van de informatiemodellen, maar omgekeerd!

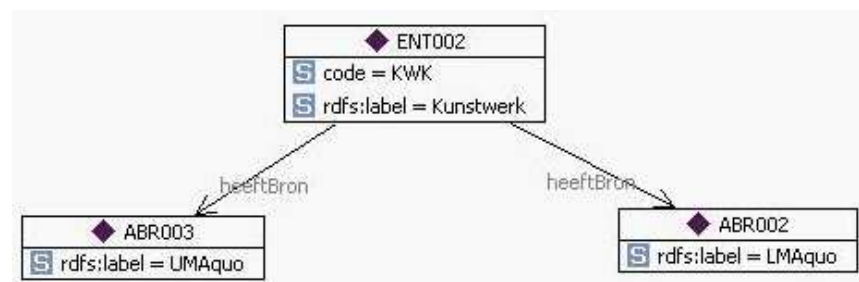
Verder is uit het voorbeeld te zien dat het concept weer een eigen bron, kennisgebied en werkproces kan hebben.



Figuur 1 De koppeling van LM Aquo entiteit Vispassage aan begrip Vispassage in Aquo-lex

Meerdere bronnen

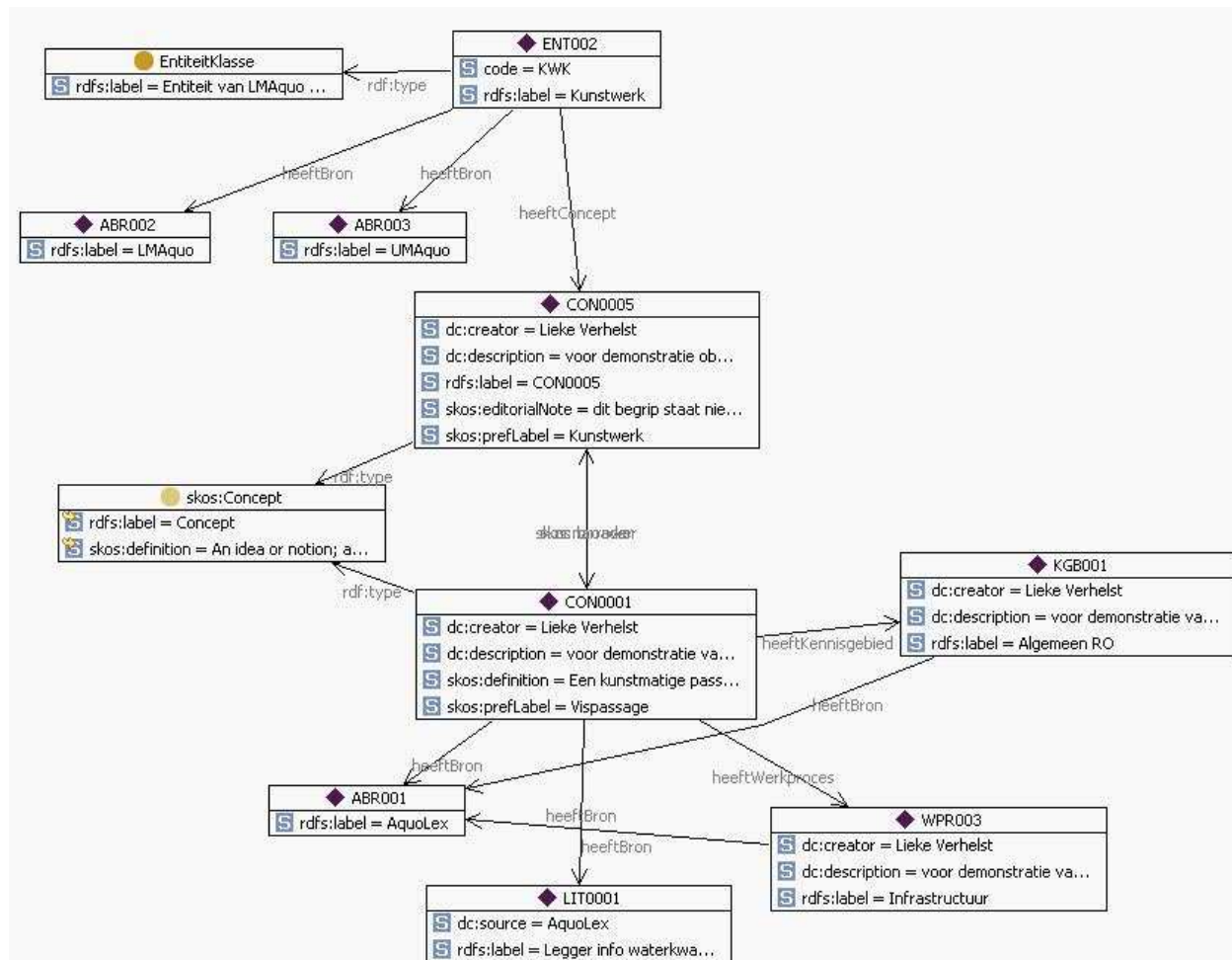
Van het object 'Kunstwerk' bestaat een entiteit in LM Aquo en een klasse in IMWA / UM Aquo. Dit kan in de objectencatalogus worden aangegeven door het toekennen van twee 'Aquo-bronnen'.



Figuur 2: 'Kunstwerk' komt voor in IMWA / UMAquo en LMAquo

Sub / superklasse

De klasse / entiteit Kunstwerk uit het LM Aquo en IMWA / UM Aquo wordt ook weer gekoppeld aan het concept 'Kunstwerk' zodat de definitie etc eenduidig vastligt. 'Concept' is namelijk het centrale punt in het semantische model. Op deze manier kan 'Kunstwerk' via een 'broader' concept gekoppeld worden aan 'Vispassage'. Dit is weergegeven in Figuur 3.



Figuur 3: De entiteit 'Kunstwerk' uit LM Aquo met de gekoppelde gegevens uit Aquo-lex (inclusief narrower relatie 'Vispassage')

3.4 Van objectencatalogus naar een informatiemodel

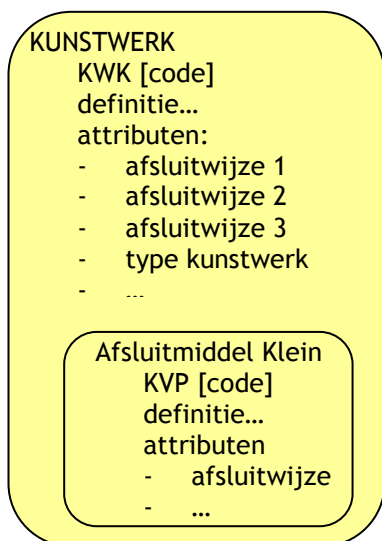
Naast de hiervoor geschetste voordelen van de integratie van de diverse onderdelen binnen de Aquo-standaard is het ook mogelijk om (nieuwe) informatie modellen af te leiden uit de objecten catalogus. Het voordeel van gebruik van de objecten catalogus is dat de gegevensblokken (objecten) eenduidig gedefinieerd worden terwijl de manier van modelleren vrij is en wordt overgelaten aan de informatiearchitect.

Wel gelden er zekere modelleerregels waar een architect zich aan zal moeten houden om de informatie uitwisselbaar en vergelijkbaar te houden. Door de ontwikkeling en gebruik van de objecten catalogus zal de noodzaak blijven om specifieke, gezamenlijke en gestandaardiseerde, informatie modellen zoals IMWA en UM Aquo te ontwikkelen voor specifieke processen. Dit omdat in de objecten catalogus geen relaties tussen objecten worden beschreven.

Vaak zijn het de relaties (en de keuze van uit te wisselen kenmerken) die van gegevens informatie maken binnen een specifieke context.

3.4.1 Voorbeeld: Kunstwerk - afsluitmiddel / wijze

In het huidige LM Aquo is de volgende modellering terug te vinden:



In Aquo-lex zijn daarnaast nog definities voor afsluitmiddel klein en kunstwerk opgenomen. Daarnaast verwijzen zowel de afsluitwijze 1,2 en 3 (Kunstwerk) als de afsluitwijze (Afsluitmiddel klein) naar dezelfde domeintabel met daarin een aantal mogelijke waarden. Het attribuut 'type kunstwerk' verwijst naar een domeintabel waarin onder andere de waarde 'afsluitmiddel klein' is opgenomen.

Modelleer keuzen

Ten aanzien van de huidige modellering kunnen een aantal opmerkingen gemaakt worden. Ten eerste is het drie keer voorkomen van de afsluitwijze bij Kunstwerk een 'workaround' om in ERD / databases een attribuut meer dan één keer te laten voorkomen. In UML / XML kan dit eenvoudig weg voorkomen worden door het attribuut 'afsluitwijze' een cardinaliteit van 0/1..3 te geven en daarmee het attribuut tot drie maal in de uitwisseling te laten terugkomen.

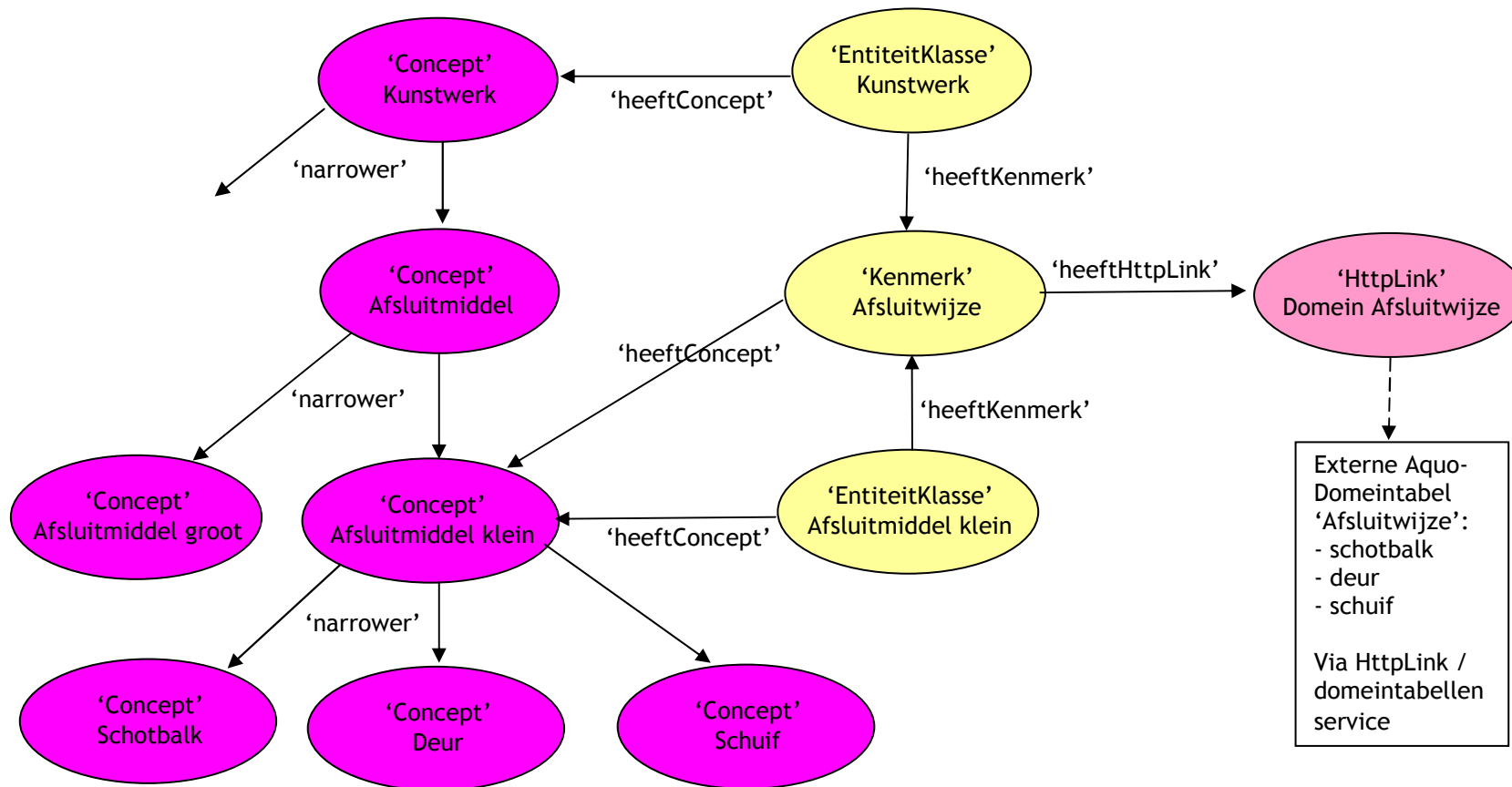
Wat verder opvalt aan de modelleerwijze is dat een afsluitmiddel klein op twee manieren gemodelleerd is; één maal als generiek kunstwerk van het type 'afsluitmiddel klein' met als attribuut de afsluitwijze, en één maal als subtype van kunstwerk met als attribuut de afsluitwijze (maar ook andere, aanvullende attributen).

Een andere vraag is natuurlijk wat een kunstwerk nu precies is; is een afsluitmiddel een kunstwerk op zich of een deel daarvan? Vaak zal het beide zijn; ook weer afhankelijk van wie het vraagt. Geredeneerd vanuit asset management is het wellicht een eigen object met een eigen beheer en bijbehorende eigenschappen. Voor het maken van een kaartbeeld is een andere keuze wellicht relevanter (er is alleen een ander symbool nodig en verder geen informatie). Een asset management systeem zal dan ook wellicht voor een meer gedetailleerde modellering kiezen dan een geografisch informatie systeem in deze situatie.

Aquo-catalogus

Bovenstaand voorbeeld is in de Aquo-catalogus op de volgende manier gemodelleerd (zonder bronnen, historie, kennisgebieden en werkprocessen):

- 'Kunstwerk' is als concept én EntiteitKlasse gemodelleerd (zie ook eerdere voorbeelden).
- 'Afsluitmiddel klein' is als concept gemodelleerd én als entiteit.
- 'Afsluitmiddel groot' is in dit voorbeeld als concept gemodelleerd (is ook een entiteit in het LM Aquo).
- Tussen 'kunstwerk' en 'afsluitmiddel klein' / 'afsluitmiddel groot' is een nieuw concept opgenomen: 'afsluitmiddel' (dus niet specifiek groot of klein). Deze laag is nieuw; er is geen overeenkomstige entiteit in het LM Aquo.
- 'Afsluitwijze' is als kenmerk gemodelleerd met een verwijzing naar het concept 'afsluitmiddel klein'.
- Alle subtypen die nu in de domeintabel 'afsluitwijze' zijn opgenomen komen terug als individuele concepten die subtypen zijn van het concept 'afsluitmiddel klein'



Figuur: Voorbeeld modellering kunstwerk - afsluitwijze in objecten catalogus

Als we nu zouden redeneren vanuit het oorspronkelijke model dan zijn we informatie kwijt geraakt; het is nl. niet meer direct te zien in de objectencatalogus (anders dan via de externe httpLink) dat er bij het attribuut afsluitwijze een domeintabel hoort; dit had evengoed een tekstveld kunnen zijn oid. Dit is ook niet de opzet van de objectencatalogus.

Omgekeerd kunnen we de bestaande modellering wel afleiden uit het voorbeeld van de objectencatalogus. Stel we hebben een applicatie die 'alleen' kunstwerken op de kaart wil zetten; in dat geval zal de applicatie kiezen voor het implementeren van het concept 'Kunstwerk'. Als leidraad kan daarbij gekeken worden naar de entiteitKlasse 'Kunstwerk' om daar te zien welke kenmerken mogelijk zijn voor die entiteit.

Als het relevant is voor die applicatie om onderscheid te maken tussen de verschillende type kunstwerken dan kunnen deze ófwel als subentiteiten gemodelleerd worden, ofwel kunnen deze middels een domeintabel 'typekunstwerk' gemodelleerd worden.

Hebben we een applicatie die zich 'alleen' bezig houdt met asset management voor 'kleine afsluitmiddelen', dan zal deze applicatie niet geïnteresseerd zijn in kunstwerken in het algemeen en deze ook niet modelleren. Dat is ook geen probleem. Als de applicatie wel onderscheid wil maken tussen de verschillende typen kleine afsluitmiddelen dan zijn ook hier weer twee mogelijkheden, namelijk als domeintabel of als subentiteiten.

De situatie met subentiteiten is (nog) niet gemodelleerd in het LM Aquo, maar kan wel worden toegepast in een eigen informatiemodel / database. Het enige wat dan nog door de informatie architect gedaan zal moeten worden is het toekennen van kenmerken aan die verschillende entiteiten.

3.4.2 Aanzet tot afleidingsregels voor de objecten catalogus

Bij het maken van een 'eigen', Aquo-conform informatie model op basis van de objectencatalogus zal de informatie architect zich aan een aantal 'spelregels' moeten houden. Gebeurt dit niet dan is uitwisseling van informatie met andere partijen niet meer mogelijk. Deze spelregels moeten nog worden opgesteld, maar er kan aan het volgende gedacht worden:

- Een datamodel / informatie model begint met een specifiek concept XXX uit de objectencatalogus en neemt deze qua eigenschappen van het concept volledig over. Wijzigingen zijn niet toegestaan; bij voorkeur wordt verwezen naar de url van het concept wat is overgenomen.
- Indien er onder het gekozen concept een 'narrower' laag aanwezig is dan kan deze ófwel als domeintabel (typeXXX) worden gemodelleerd, ofwel als individuele entiteiten. De waarden uit de domeintabel of de nieuwe entiteiten nemen deze qua eigenschappen van het concept volledig over. Wijzigingen zijn niet toegestaan; bij voorkeur wordt verwezen naar de url van het concept wat is overgenomen.
- Samenvoegingen van bestaande concepten zijn niet toegestaan. Ook het samenvoegen van concepten met een andere 'narrower' relatie (andere super/subklasse) tot één domeintabel is niet toegestaan.
- Kenmerken die aangemaakt worden met een eenheid en/of hoedanigheid hebben altijd een eenheid en/of hoedanigheid uit de Aquo-domeintabel met de bijbehorende definitie etc.
- Het toepassen van synoniemen (of andere talen) van concepten is toegestaan zolang de verdere eigenschappen van het concept niet aangepast worden.

4. Het model van de objectencatalogus

Het ontwerp van (het datamodel van) de objectencatalogus is ontworpen uitgevoerd in OWL als uitbreiding op het SKOS datamodel. Met andere woorden, voor het deel waarin de begrippen zijn beschreven is gebruik gemaakt van SKOS; de overige klassen (die niet in SKOS zijn opgenomen) zijn toegevoegd als een OWL uitbreiding op het SKOS datamodel. Dit is binnen SKOS toegestaan. Daarmee is bereikt dat een uniforme, uitwisselbare kern ontstaat voor de begrippen die nu in Aquo-lex zijn opgenomen (SKOS datamodel).

In het semantische ontwerp van de objectencatalogus zijn de attributen in datatype properties. De relaties tussen klassen zijn gemaakt met object properties (voorbeeld: Kenmerk heeftEenheid AquoDomeinEenheid).

Voor meer informatie over SKOS en OWL wordt verwezen naar Bijlage A.

4.1.1 Objectklassen in de objectencatalogus

Objectklasse	Functie in objectencatalogus
<i>Thing</i>	- <i>abstracte klasse in RDF; hiervan zijn alle overige objecten afgeleid.</i>
Concept	Begrip uit Aquo-lex; bevat o.a. definitie, toelichting, afkorting, taal, etc. Verwijst ook naar hogere orde / lagere orde begrippen (sub / superklassen).
EntiteitKlasse	Naam en code uit LM Aquo; UM Aquo en IMWA. Verwijst naar kenmerken die toegestaan zijn en naar het begrip (concept) wat erbij hoort.
Kenmerk	Attribuut met naam en code uit LM Aquo, UM Aquo en IMWA. Verwijst naar de bijbehorende eenheid en hoedanigheid (indien van toepassing) en naar het bijbehorende begrip (concept).
AquoDomeinEenheid	Bevat de waarden uit de overeenkomstige Aquo domeintabel met een verwijzing naar het begrip (concept) (indien van toepassing). Dit object is opgenomen om bij kenmerken de mogelijkheid te hebben de eenheid (bv voor de lengte / breedte) aan te geven.
AquoDomeinHoedanigheid	Bevat de waarden uit de overeenkomstige Aquo domeintabel met een verwijzing naar het begrip (concept) (indien van toepassing). Dit object is opgenomen om bij de kenmerken / eenheden een nadere aanduiding van de eenheid (bv t.o.v. NAP) te kunnen geven.
Graphic	Bevat een verwijzing (url) naar een figuur met meer informatie / toelichting.
HttpLink	Bevat een verwijzing (url) naar een bron op het internet met meer informatie / toelichting.
LitRef	Bevat een verwijzing naar de literatuur waar het concept in gedefinieerd wordt of vanuit ontstaan is.
AquoBron	De oorspronkelijke herkomst van de EntiteitKlassen, Kenmerken, Domeinen en Concepten wordt vastgelegd in de klasse AquoBron. De individuals van AquoBron zijn bijvoorbeeld LMAquo, IMWA, UMAquo, AquoLex etcetera.
Kennisgebied	Beschrijft tot welk kennisgebied een concept behoort (bv 'Hydrologie' of 'Civiele techniek').
Werkproces	Beschrijft tot welk werkproces een concept of entiteit behoort (bv 'Keringenbeheer' of 'Monitoring').
Historie	Houdt bij welke wijzigingen er op een object zijn uitgevoerd en wat er gewijzigd is en wat de status van een object is, etc.
Autoriteit	Beschrijft door welke autoriteit een begrip wordt gebruikt (bv

Objectklasse	Functie in objectencatalogus
	'KRW' of 'NHV')

Naast de Concept klasse zijn door het gebruiken van het SKOS schema nog de klassen Collection, OrderedCollection en ConceptSchema aanwezig. Deze worden niet gebruikt. De Thing klasse is altijd aanwezig in een ontologie.

4.1.2 Properties van de objectencatalogus

In Tabel 1 zijn de properties weergegeven die zijn gebruikt. Verschillende properties komen uit bestaande namespaces (dublin core - dc, rdf en rdfs, skos). Voor de objectencatalogus zijn eigen properties gedefinieerd in de 'catalogus' namespace.

Tabel 1: Alle gebruikte properties en hun namespace

namespace	property	Type	Triple
dc	creator	String	Datatype property bij Thing
	description	String	Datatype property bij Thing
	source	String	Datatype property bij Thing
rdfs	label	String	Datatype property bij rdfs:Resource
rdf	type (de functie hiervan is het weergeven van de klasse waartoe het object behoort)	rdf:Property	Datatype property bij rdfs:Resource
skos	skos:prefLabel	String	Datatype property bij Concept
	skos:altLabel	String	Datatype property bij Concept
	skos:editorialNote	String	Datatype property bij Concept
	skos:definition	String	Datatype property bij Concept
	skos:broader	Object	Concept skos:broader Concept
	skos:narrower	Object	Concept skos:narrower Concept
catalogus	bepaalt	Object	Graphic; HttpLink; LitRef; AquoBron bepaalt Concept
	beveeltAan	Object	Autoriteit beveeltAan Concept (inverse van isAanbevolenDoor)
	heeftBron	Object	Concept heeftBron Graphic; HttpLink; LitRef; AquoBron
	heeftConcept	Object	AquoDomeinEenheid; AquoDomeinHoedanigheid; Kenmerk; EntiteitKlasse heeftConcept Concept
	heeftEenheid	Object	Kenmerk heeftEenheid AquoDomeinEenheid
	heeftHistorie	Object	Thing heeft Historie Historie
	heeftHoedanigheid	Object	Kenmerk heeftHoedanigheid AquoDomeinHoedanigheid
	heeftHttpLink	Object	Thing heeftHttpLink HttpLink
	heeftKenmerk	Object	EntiteitKlasse heeftKenmerk Kenmerk
	heeftKennisgebied	Object	Concept heeftKennisgebied Kennisgebied
	heeftWerkproces	Object	Concept heeftWerkproces Werkproces
	isAanbevolenDoor	Object	Concept isAanbevolenDoor Autoriteit (inverse van beveeltAan)
	sterkGerelateerd	Object	Concept sterkGerelateerd Concept
	zwakGerelateerd	Object	Concept zwakGerelateerd Concept
	actienemer	string	Datatype property bij Historie
	attribuutVeranderd	String	Datatype property bij Historie
code	String	Datatype property bij EntiteitKlasse; Kenmerk; AquoDomeinEenheid; AquoDomeinHoedanigheid	
datumEindeStatus	Date	Datatype property bij Historie	
datumIngangStatus	Date	Datatype property bij Historie	
objectVeranderd	String	Datatype property bij Historie	

status	String	Datatype property bij Historie
visualisatie	String	Datatype property bij EntiteitKlasse; Kenmerk
synoniem	String	Datatype property bij Concept
waardeWas	String	Datatype property bij Historie
waardeWerd	String	Datatype property bij Historie

De relatie ‘sterk gerelateerd’ en ‘zwak gerelateerd’ zijn niet in SKOS gedefinieerd en daarom apart in de objectencatalogus opgenomen.

Alle klassen zijn door middel van de property heeftConcept gekoppeld aan de Concept klasse. Dat betekent dat van de EntiteitKlassen, Kenmerken, en domeintabellen ‘individuals’ in de vorm van Concepten kunnen bestaan. Op deze manier is de koppeling gemaakt tussen Aquo-lex (Concept klasse) enerzijds en LM Aquo (Entiteiten in de EntiteitKlasse klasse), IMWA / UM Aquo (Klasse in de EntiteitKlasse klasse) en de domeintabellen Eenheid en Hoedanigheid (AquoDomeinEenheid en AquoDomeinHoedanigheid) anderzijds.

De domeintabellen AquoDomeinEenheid en AquoDomeinHoedanigheid zijn met de properties heeftEenheid en heeftHoedanigheid gekoppeld aan Kenmerk.

4.1.3 Meertaligheid

In SKOS kan de taal worden opgenomen door meerdere waarden bij prefLabel te specificeren. In de objectencatalogus zijn de volgende SKOS properties actief die ook in Aquo-lex voorkomen:

SKOS property	Attribuut uit Aquo-lex
skos:prefLabel	Naamgeving + taal
skos:altLabel	Afkorting
skos:editorialNote	Toelichting
skos:definition	Definitie

Het gebruik van verschillende talen wordt als volgt genoteerd (vetgedrukte tekst):

```
<skos:Concept rdf:ID="CON0006">
  <skos:prefLabel xml:lang="en">wave frequency</skos:prefLabel>
  <skos:prefLabel xml:lang="nl">Golffrequentie</skos:prefLabel>
  <dc:source rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >AquoLex</dc:source>
  <skos:definition
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >de reciproke waarde van de golfperiode</skos:definition>
  <dc:description
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >voor demonstratie objectencatalogus</dc:description>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >CON0006</rdfs:label>
</skos:Concept>
```

4.1.4 Propertes per objectklasse

Tabel 2 is een weergave van alle mogelijke properties (object properties - relaties zijn **vetgedrukt**) per Object. Te zien valt dat alle klassen de eigenschappen van de topklasse 'Thing' overerven vanwege de subklasse relatie die er per definitie in elke ontologie is met 'Thing'.

Daarnaast is de historie van alle individuals vastgelegd door alle klassen de eigenschap 'heeftHistorie' te geven via de toekenning van het domein 'Thing' aan 'heeftHistorie'. Hetzelfde geldt voor 'heeftHttpLink' en 'heeftBron'. Dit levert onlogische combinaties op waarmee bij invullen van de data rekening gehouden moet worden (HttpLink heeftHttpLink, Historie heeftHistorie etc.). Bij het invoeren van gegevens in de objectencatalogus zullen dit soort 'onmogelijke' relaties door een goede beheertool afgevangen moeten worden. Als onlogisch vastgestelde relaties zijn in de tabel hieronder in het **rood** weergegeven.

Tabel 2: Properties per klasse (object properties zijn **vetgedrukt**; onlogische relaties **rood**)

Aquo catalogus velden	UniqueKey (URI)	Eigenschappen Objectencatalogus
Thing	nvt	rdfs:label dc:description heeftHttpLink dc:creator heeftBron dc:source heeftHistorie
Concept	CON[volgnr]	skos:prefLabel + language skos:definition skos:altLabel skos:editorialNote synoniem rdfs:label dc:description heeftHttpLink dc:creator dc:source heeftHistorie isAanbevolenDoor heeftBron heeftWerkproces heeftKennisgebied skos:broader skos:narrower alle andere skos properties zwakGerelateerd sterkGerelateerd
EntiteitKlasse	ENT[volgnr]	rdfs:label dc:description dc:creator

Aquo catalogus velden	UniqueKey (URI)	Eigenschappen Objectencatalogus
		heeftHttpLink heeftBron dc:source heeftConcept code visualisatie heeftKenmerk heeftHistorie
Kenmerk	KEN[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink dc:source heeftBron code visualisatie heeftEenheid heeftHoedanigheid heeftConcept heeftHistorie
AquoDomeinHoedanigheid	HOE[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink dc:source heeftConcept heeftBron code heeftHistorie
AquoDomeinEenheid	EEN[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink dc:source heeftBron code heeftConcept heeftHistorie
Graphic	GRA[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink heeftBron dc:source bepaalt heeftHistorie
HttpLink	HTL[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink

Aquo catalogus velden	UniqueKey (URI)	Eigenschappen Objectencatalogus
		heeftBron bepaalt dc:source heeftHistorie
LitRef	LIT[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink heeftBron bepaalt dc:source heeftHistorie
AquoBron	ABR[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink heeftBron bepaalt dc:source heeftHistorie
Kennisgebied	KGB[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink heeftBron dc:source heeftHistorie
Werkproces	WER[<i>volgnr</i>]	rdfs:label dc:description dc:creator heeftHttpLink heeftBron heeftHistorie dc:source
Historie	HIS[<i>volgnr</i>]	rdfs:label dc:description heeftHttpLink dc:creator heeftBron dc:source heeftHistorie DatumIngangStatus DatumEindeStatus Status ActieNemer ObjectVeranderd AttribuutVeranderd WaardeWas WaardeWerd
Autoriteit	AUT[<i>volgnr</i>]	rdfs:label

Aquo catalogus velden	UniqueKey (URI)	Eigenschappen Objectencatalogus
		dc:description dc:creator heeftHttpLink heeftBron dc:source heeftHistorie beveeltAan

Bijlage A Theorie en concepten achter de objectencatalogus

Deze bijlage is gebaseerd op de Alterra rapporten 'Bron theseaurus' en 'Onderzoeksrapport Aquo Objectencatalogus'

A.1 "Representation" en "Reasoning" systeem

In dit project wordt de basis gelegd voor inpassing van gegevens over water in het toekomstige Web 3.0. Door de opzet van de Objectencatalogus wordt gewerkt aan wat in de literatuur vaak met een "representation and reasoning" systeem (RRS) wordt aangeduid. In dit geval zal een RRS toegepast worden voor de verdere ontwikkeling van een Semantische Web oplossing voor het intelligent ontsluiten van waterbeheer informatie.

Het semantisch web is volgens de W3C: "... a common framework that allows data to be shared and reused across application, enterprise, and community boundaries" [2]. Het specifieke kenmerk van het semantisch web is dat het een web is van open data en niet van (door een applicatie) gepresenteerde data. Hierdoor ontstaat de mogelijkheid voor derden om de gereserveerde databronnen als data te gebruiken in eigen applicaties en te koppelen aan andere open databronnen.

In z'n algemeenheid bestaat een RSS bestaat uit de volgende componenten:

1. Een taal voor communicatie met een computer (syntax)
2. Een methode om betekenis aan de taal te koppeling (semantiek)
3. Procedures om antwoorden te genereren (berekenen) op vragen die gesteld worden aan de computer via de taal ("reasoning" theorie)

De ontwikkeling van een RSS omvat in z'n algemeenheid de volgende stappen (Poole 1998):

1. Definiëren en karakteriseren van het domein. Met andere woorden het domein waarover het voor een computer mogelijk moet zijn vragen te beantwoorden.
2. Het onderscheiden van de afzonderlijke "dingen" (entiteiten, begrippen) die vastgelegd moeten worden (de ontologie)
3. De "dingen" vastleggen in de computer met behulp van symbolen.
4. Toevoegen van kennis over relaties tussen "dingen". Om vragen te kunnen beantwoorden moet de computer kennis hebben over hoe de verschillende "dingen" in de wereld aan elkaar zijn gerelateerd.
5. Stel vragen aan het RSS. Dit zet het RSS aan het redeneren (reasoning) om, gebaseerd op de "dingen" en de kennis over hoe de "dingen" aan elkaar zijn gerelateerd, een probleem op te lossen.

A.2 RDF en OWL

Zoals al aangestipt in de inleiding wordt de basis van het Semantisch Web gevormd door data die zodanig wordt opgeslagen en ontsloten dat deze via het internet op een relatieve manier eenvoudig herbruikbaar is. Een belangrijk aspect hierbij een algemeen geaccepteerd standaard voor de integratie en combinatie van kennis die opgeslagen ligt op verschillende plekken in het internet. De belangrijkste standaard is het Resource Description Framework (RDF) een "W3C recommendation". RDF is gebaseerd op de XML syntax en bouwt verder op de standaarden die deel uitmaken van XML.

In dit framework is vastgelegd hoe data kan worden weergegeven op het web. De specificatie van RDF Schema (RDFS) legt de syntax vast van objecten (klassen) en hun relaties. De basisgedachte achter RDF is de triple relatie Subject, Predicate, Object. Predicate is de relatie weergave tussen het object en subject, bv een synoniem. Deze drie-eenheid is de bouwsteen van alle datamodellen in het semantisch web.



Figuur: Grafische weergave van de RDF triplet.

Ieder triplet is een representatie van een relatie tussen twee dingen uit de echte wereld (entiteiten). De pijl wijst altijd van subject naar object. Het is van groot belang dat subjects, predicates en objects in de RDF graph altijd identificeerbaar zijn op het internet. Elk onderdeel van een triple (object, predicate en subject) heeft een URI (Uniform Resource Identifier) in de vorm van een http-link. Dit is de unieke sleutel waarmee het kan worden geïdentificeerd en benaderd. Een instantie van een object of subject heet in RDF een individual.

De figuur hieronder geeft hiervan een voorbeeld voor een persoon geïdentificeerd via <http://www.w3.org/People/EM/contact#me> met de naam Eric Miller en het emailadres em@w3.org en de titel DR . De objecten die hier gebruikt worden komen van verschillende ontologieën (meestal) te herkennen aan de verschillende URI's



Figuur: Voorbeeld van het gebruik van URI voor identificatie van RDF entiteiten (bron: <http://www.w3.org/TR/rdf-primer/>)

Wordt bovenstaande uitgeschreven in de XML-syntaxis (RDF/XML) dan ontstaat er iets zoals in box 1 is weergegeven. Het belangrijkste uitgangspunt van RDF is dat de entiteit die beschreven worden eigenschappen hebben die kunnen worden uitgedrukt in een waarde (via tekst of nummers)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person
rdf:about="http://www.w3.org/People/EM/contact#me">
  <contact:fullName>Huibert-Jan
Lekkerkerk</contact:fullName>
  <contact:mailbox
rdf:resource="mailto:h.lekkerkerk@idsw.nl"/>
  <contact:personalTitle>Ing.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

OWL

RDF is met name bedoeld voor het creëren en ontsluiten van databases via het internet. Om daarnaast ook deze informatie door computers te kunnen laten verwerken is OWL ontwikkeld. OWL is een uitbreiding op RDF en maakt het mogelijk om rijkere kennisconstructies te maken. OWL is specifiek ontwikkeld voor het definiëren van ontologieën voor "machine-based

reasoning". OWL gebruikt RDF en RDF-schema als basis voor het definiëren van de vocabulaire (classes, properties en relaties).

Zoals al gezegd vormt OWL de basis voor het vastleggen van een ontologie. In de klassieke AI is een ontologie een formele specificatie van een concept. Als we het hebben over het representeren van kennis kan een ontologie specifiekere gedefinieerd worden als de beschrijving van de concepten en relaties tussen deze concepten in een specifiek domein.

Een absolute vereiste voor een ontologie is dat deze eenduidig moet zijn zowel vanuit een syntactische als een semantisch perspectief. Syntactische eenduidig betekend dat er een geformaliseerde set of statements en argumenten (of kortweg de "grammatica") nodig is waarmee de concepten beschreven kunnen worden. Semantische eenduidigheid betekend dat de beschrijving van de entiteiten via de symbolen uit de taal een uniek bekende relatie hebben met het domein waarvoor de ontologie is opgesteld.

In een praktische taal worden concepten gedefinieerd als "classes" en de individuele entiteiten als objecten die tot een bepaalde klasse behoren (met andere woorden instantiaties van een klasse);zo ook in OWL. Het basis element van RDF en dus van OWL is de zgn triplet bestaande uit een object (een representatie van een ding uit de werkelijke wereld), een relatie (en een ander object (een ander ding).

SKOS

Specifiek voor de ontsluiting van kennis systemen (Knowledge Organization Systems of 'KOS') is SKOS (Simple Knowledge Organization System) ontwikkeld. Typische voorbeelden van een SKOS toepassing is een thesaurus, taxonomie of classificatie schema.

SKOS is zeer geschikt om beschrijvingen van en verbanden tussen begrippen weer te geven. Wat betreft beschrijvingen zijn er mogelijkheden om naamgeving, definitie, voorbeeld, notes en taal op te slaan. Wat betreft relaties zijn naast hogere orde/lagere orde ook exacte relatie en het algemene 'gerelateerd' aanwezig. Door het beschrijven van de verschillende manieren waarop begrippen gerelateerd zijn kan een semantische laag aan de objectencatalogus worden toegevoegd. Deze laag stelt gebruikers in staat meer contextinformatie rond het begrip te verzamelen.

SKOS biedt ook de mogelijkheid om meerdere KOSsen te koppelen door middel van de klasse 'ConceptScheme' en de relatie 'inScheme'

Omdat OWL en SKOS gebaseerd zijn op RDF kunnen ze als uitbreidingen binnen een RDF framework worden opgenomen. Op die manier kan een combinatie framework van RDF, OWL en SKOS worden gemaakt. De aparte OWL en SKOS syntax zijn aan hun eigen namespace binnen het framework te herkennen.

Omdat SKOS en OWL eigenschappen hebben die van toepassing zijn op de doelstellingen van de objectencatalogus is het ontwerp van de objectencatalogus in deze standaard beschreven. Waar het SKOS datamodel onvoldoende mogelijkheden bood zijn aanvullingen gedaan in OWL.

A.3 RDF als taal

Een belangrijk kenmerk van RDF waardoor het zich van andere modelleer talen laat onderscheiden is de relatie oriëntatie (in tegenstelling tot klasse oriëntatie). In RDF kan een predicate (=relatie, property) bestaan zonder dat er een klasse bij hoort. Klassen worden juist gekoppeld door predicates (properties).

Er bestaan verschillende soorten properties. De belangrijkste zijn:

- Datatype Property. Een datatype property wordt gebruikt om individuals aan data values te koppelen. Een grove vertaling van datatype property naar andere modelleertalen zoals UML en ERD zou zijn 'attribuut'.
- Object Property. Een object property wordt gebruikt om individuals aan individuals te koppelen. Een grove vertaling van datatype property naar andere modelleertalen zoals UML en ERD zou zijn 'relatie' of 'associatie'.
- Annotation Property. Deze is bedoeld om informatief commentaar te verwerken. Het verschil tussen een annotation property en een string datatype property is onder meer dat een annotation property niet gebruikt mag worden in een constraintregel.

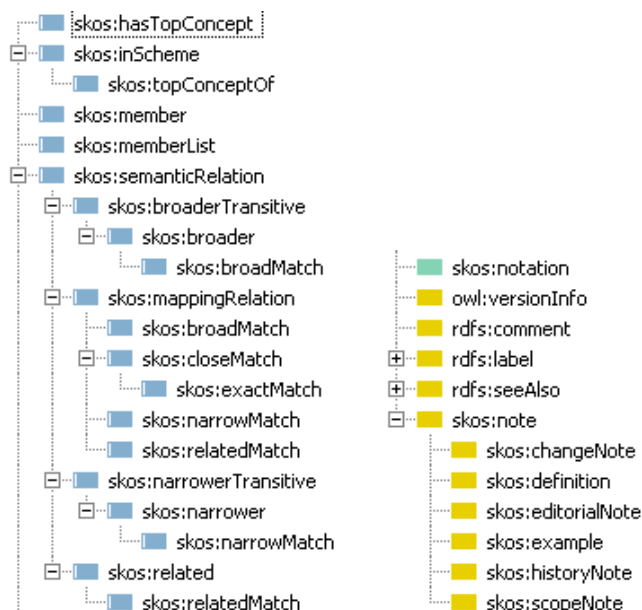
Het bereik van een property wordt door het *domain* en de *range* van de property bepaald. Voor een object property is dat: *domain* property *range* (bijvoorbeeld *Kind* heeftOuder *Moeder*).

Voor een datatype property is de *range* gelijk aan de waarde die de datatype property heeft (string, integer, float, boolean etc). Het domein van een datatype property is de klasse waarvoor het datatype geldt (bijvoorbeeld: 'naam' heeft als domein 'Kind' en als range 'string').

Een inverse property is een property die een omgekeerde domain-range relatie heeft met een andere property. Een voorbeeld is: 'heeftOuder' en 'isOuderVan'. Door ook de inverse relatie van een property vast te leggen wordt een regel in het model opgeslagen die gebruikt kan worden om nieuwe informatie af te leiden. Bijvoorbeeld: stel dat in de data is vastgelegd dat het kind een individual is met een heeftOuder relatie, dan kan uit deze informatie worden afgeleid dat er ook een individual is met een isOuderVan relatie.

Een subklasse beschrijft de 'is-a' relatie. Hieruit volgt dat als iets een subklasse is, dat het dan vanzelfsprekend ook tot de bovenliggende klasse behoort. De properties van de superklasse worden overgeërfd door de subklasse.

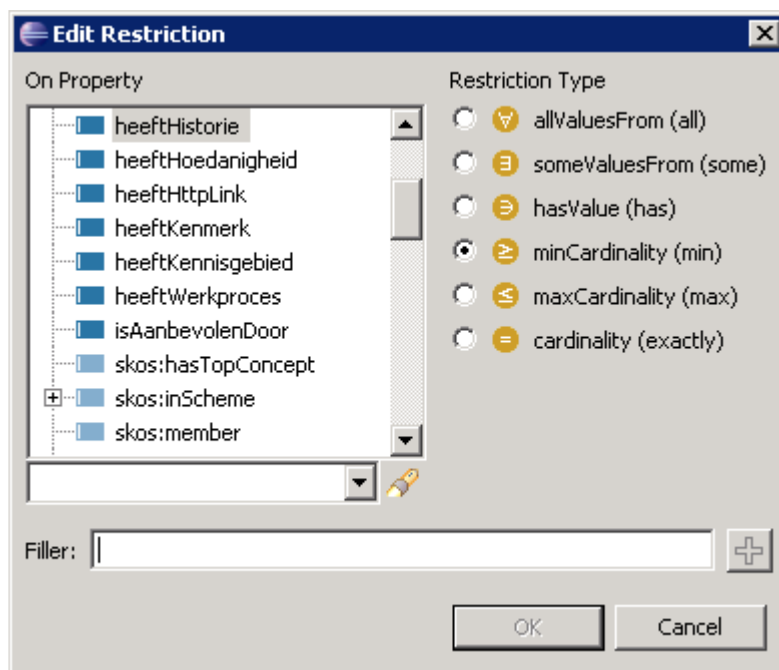
Het SKOS datamodel heeft de volgende properties (Figuur 4):



Figuur 4: de SKOS properties (links de datatype properties, rechts de object properties)

Cardinaliteit en constraints

Een OWL ontologie kan cardinaliteit en constraints bevatten. Deze regels worden in het ontwerp opgenomen als restrictie (Figuur 5)



Figuur 5: Het toekennen van restricties

Ook andere restricties zoals bijvoorbeeld de union tussen klassen (AND) en de intersectie (OR) kunnen worden toegekend. Niet alle restricties worden bij invullen in een OWL tool gecontroleerd. Bij een zogenaamde consistency check wordt gekeken of de logische constraints in de ontologie op elkaar zijn afgestemd. Voorbeeld: stel dat is gespecificeerd dat een AquoDomeinEenheid een eenheid is van enkel en alleen een Kenmerk dan kan een AquoDomeinEenheid geen eenheid zijn van een Concept:

EntiteitKlasse heeftEenheid AquoDomeinEenheid is geldig

Concept heeft Eenheid AquoDomein Eenheid is ongeldig

Een consistency check gebeurt na het invullen van de data door het runnen van een reasoner.

Queryen

Een RDF/OWL bestand bevat data die bevroegd kan worden door het uitvoeren van een query. Dit werkt analoog aan query technieken bij een database, alleen de query engine en de query syntax zijn anders. De query taal voor RDF/OWL is SPARQL.

Een SPARQL query is bijvoorbeeld:

```
SELECT ?x ?y
WHERE {x skos:prefLabel ?y .
FILTER (?y = "Vispassage")}
```

Het resultaat van deze query is de individual waarvan de property skos:prefLabel de waarde "Vispassage" heeft.

Reasoning

Reasoning is het evalueren van de gevolgen van in de RDF/OWL opgeslagen regels en constraints.

Reasoning is een actie die op een RDF/OWL bestand met data wordt uitgevoerd. Niet alle constraints in een RDF/OWL bestand worden bij data invoer door een ontologie editor gevalideerd. Dit moet ter controle achteraf gebeuren door het aanroepen van een reasoner. In sommige ontologie editors is een reasoner ingebouwd (Topbraid Composer) bij andere editors moet hij extern worden gestart en vanuit de editor worden aangeroepen (vroegere Protege versies).

Het resultaat van reasoning kan zijn:

1. een classificatie. Een reasoner kan ook individuals classificeren. Hierbij worden lege klassen gemaakt die voorzien zijn van een logische omschrijving. Als voorbeeld nemen we de wijn ontologie [3]: stel dat voor de klasse 'PinotNoir' nog geen individuals zijn gedefinieerd en de klasse zelf volgens de logische omschrijving is gedefinieerd:

Wine, madeFromGrape has PinotNoirGrape, madeFromGrape max 1 and hasColor has Red

De reasoner zal alle individuals die voldoen aan deze regels plaatsen (classificeren) in de klasse 'PinotNoir'.

2. consistency checking. Een reasoner kan de consistency van een RDF/OWL bestand checken. Hierbij wordt de ingevulde data op de constraints getest en bij onjuiste invoer wordt dit gemeld.
3. concept satisfiability. Het checken van concept satisfiability wil zeggen dat de reasoner test of een klasse wel individuals kan hebben.
4. realisatie (inference). Realisatie of inference wil zeggen dat een reasoner uit regels die zijn opgeslagen in de ontologie een conclusie kan afleiden. Het bekendste voorbeeld is:

Alle mensen zijn sterfelijk

Socrates is een mens

Dus Socrates is sterfelijk