# De RCE kwaliteitsstandaard voor ontologieën

Versie 0.3

Patrick Mout

Joop Vanderheiden

Frans van der Zande

# Inhoudsopgave

[**De RCE kwaliteitsstandaard voor ontologieën**](#_qv9s0l8vqjpj) **0**

[**Inhoudsopgave**](#_vxfamr77v319) **1**

[**Inleiding**](#_ilq58xk30sap) **2**

[Afwijkingen en openstaande issues](#_umork6q5j3xw) 2

[**Architectuur uitgangspunten voor de Linked Data Voorziening (LDV)**](#_neat69eukobl) **5**

[**Uitgangspunten en ontwerpbeslissingen voor de RCE ontologieën**](#_rolir69s4v39) **6**

[Algemene RCE uitgangspunten voor het modelleren van ontologieën](#_fpjkbaoii3cv) 6

[Afspraken over RCE werkwijze voor het modelleren van de ontologieën](#_plrwju1yxnk5) 6

[Aanpak ontologie validatie](#_93vwkysduklh) 7

[De RCE ontwerpbeslissingen](#_9knoczu26apb) 7

[De RCE ontologie documentatiestandaard](#_vp6yiuwok8xb) 9

[De visualisatie standaard](#_klsxn71vfead) 10

[Uitgangspunten m.b.t. het modelleren van kardinaliteit](#_5kymaxwmb2hx) 11

[**De betekenis van concepten**](#_yro3ly194j3q) **12**

[**Annotatie van de status van de gehele ontologie**](#_m72qcllqnvy4) **14**

[Aanzet beschrijving versiebeheer irt named graph](#_qtn0gmm1ottu) 15

[**Glosarium**](#_vqra35kbt2fq) **16**

[**Bijlagen**](#_m6wmy81n8pej) **17**

[Sjabloon voor classes en properties](#_b73bbbbzqxx8) 17

[Versie planning en ToDo](#_te3p4t63f5gc) 17

[Cultureel Erfgoed Ontologie (CEO)](#_gbwc2zzggco1) 17

[De ZAAK ontologie (ZO)](#_soaa1c5g8vsb) 17

[Rest materiaal ontologie modellering](#_5puojbvbm1sg) 17

# 

# Inleiding

Dit document bevat de algemene uitgangspunten, afspraken voor het modelleren, visualiseren en documenteren van de RCE eigen ontologieën,

De volgende RCE ontologieën worden voorzien:

* ceo: Cultureel Erfgoed Ontologie (CHO en KENNIS)
* zo: zaakinformatie Ontologie (ZAAK)
* ztco: ZaakTypeCatalogus Ontologie (ZAAK)
* ztceo ZaakTypeCatalogus Eigenschappen Ontologie (ZAAK)
* rec: records (documenten)
* bib: bibliografische gegevens
* media: diverse media bestanden (bv afbeeldingen uit de beeldbank)
* lxr: Ontologie voor het verrijken van RCE data. Linked data Usability Ontology RCE (LuXoR)

In de huidige fase (2018) zal Linked Data Voorziening (LDV) de CEO en ZAAK-ontologie (ZO) bevatten. In de CEO zullen uitsluitend de classes, attributen en relaties uit de logische datamodellen CHO en KENNIS worden opgenomen. De ZO zal gebaseerd worden op het logische model ZAAK en de eigenschapstypen die in de verschillende zaaktypen zijn gedefinieerd.

## Afwijkingen en openstaande issues

Op een aantal punten kan beargumenteerd worden afgeweken van de in dit document vermelde uitgangspunten. Generieke afwijkingen en openstaande issues worden hieronder opgenomen. Specifieke afwijkingen en openstaande issues worden ook beschreven bij de documentatie van de ontologie zelf.

**Afwijkingen en** **issues:**

1. De ZAAK documentering afronden. O.a. Zaak-Eigenschapstype documenteren en thesaurus aanvullen met lijsten van eigenschapstype. Controle juiste gebruik in namespace in Visio en TopBraid**. (Todo)**
2. De CEO bevat nog geen inverse properties (uitzonderingen: zie samenhang en heeftKennisregistratie) terwijl dit voor het browsen door de LD wel wenselijk is. **(versie 2)**
3. Als ontoloog wil ik in de ontologie de teksten van rdfs:comment controleren of de classes en predicates goed beschreven zijn.  **(Todo)**
4. Brondata van de ontologieën in turtle doorlopen.
5. Modelleren metadata van de ontologie en de metadata van de data. (Zie NDE register, NGR en Bv KB), zodat we de dataset kunnen publiceren in het NDE-register en in onze eigen P&D-catalogus. Hierdoor wordt de Linked Data vindbaar.
6. De materiële en formele historie is in de huidige versie nog niet meegenomen. Dit is wel nodig voor versiebeheer (persistentie) van de data en voor tijdreizen. (zie ook visio tabblad “Ax MFHistorie”) Duidelijk moet zijn wat de lifecycle is van de data. M.a.w. wanneer leidt een wijziging in de data tot andere versie van het record en/of MF historie. Moeten we in de ontologie onderscheid gaan maken tussen het ding en de informatie (met historie) over het ding (zie foaf:document bij Kadaster). (versie 2)
7. Uitgangspunt is om inferred triples te maken tijdens ETL, zodat we in control zijn over de linked data. Bepalen welke triples dit moeten zijn.
8. Het onderzoeken van de noodzaak om logische eigenschappen aan de properties (RDF-plus). Symmetrie, transitiviteit etc.toe te voegen. Hierdoor zijn betekenis vollere afleidingen te maken, waardoor de linked data waardevoller wordt. (versie 2) (Lieke vanwege gebruik van SHACL wordt dit overbodig)
9. Onderzoek of het handig is om ontologie-verrijkingen onder te brengen in een zogenaamde ceo-plus ontologie. (bv door het gebruik van een ceo:kadasterDeelEnNummer naast de reeds bestaande twee properties). Hierdoor blijft de hoofd-ontologie stabiel en is er meer ruimte om beter te voldoen aan de vraag. Dit impliceert een inferencing/ruling slag na ETL. De vraag is of dit dan ‘automatisch’ zichtbaar wordt in API en/of dotnetwebstack. Hierdoor wordt de capability van het ontologie-team vergroot. (versie 2)
10. Het consequent wel opnemen in de ETL van attributen waarvoor nog geen data in het model zit in relatie tot de ETL, zodat er geen onverwachte triples ontstaan op het moment dat er wel data in de datalaag komt. Breng huidige situatie in kaart. En neem een besluit over gepubliceerde ontologie.
11. De ceo:GBKN en ceo:BRT zijn niet opgenomen in de ontologie. Het is niet bekend of hier ook data over beschikbaar is. (versie 2)
12. In de CEO worden classes en properties soms wel en soms niet gedefinieerd als een subClass of subProperty uit een extern vocabulair. Dit lijkt inconsequent de vraag is of dit ook zo is. Vergelijk LOCN (wel) en BRK (niet). Zie ook het betreffende visio-tabblad. Daarnaast zou dit ook als een equivalent class of property gemodelleerd kunnen worden. M.a.w. We hebben een Best Practise nodig voor het gebruik (koppelen) met een externe vocabulair. Zodat er consequente afleidingen gemaakt kunnen worden dit heeft als gevolg dat de ontologie als betrouwbaar wordt ervaren. (versie 2)
13. De ‘nummer’ properties voor identificatie vervangen door UUID? (bv ceo:BAGRelatienummer) Dan ook naam van de property vervangen door identificatienummer? Dit komt de persistentie ten goede waardoor we beter voldoen aan de eisen uit de DERA. (versie 2)
14. Analyseren wat wij kunnen of moeten met ceo:featureIdentificatie, zodat beter bekend wordt hoe een relatie gelegd kan worden met de data uit de basisregistraties (m.n. BAG). (versie 2)
15. De ontologie documentatie in de ontologie zelf opnemen. (bv wanneer wordt een triple gemaakt als de range een boolean is zie bv de property formeelstandpunt). Hierdoor kan de Linked Data beter begrepen worden door gebruikers.
16. Kritisch onderzoek naar het consequent gebruik van tekst ‘naam’ in de naam van een property. Hierdoor kan de Linked Data beter begrepen worden door gebruikers. Nader beschrijven de keuze voor heeftXNaam alleen voor constructie naar range skos:Concept. (zie Visio plaat Kennisregistratiepatroon) (versie 2)
17. De naamgeving van de properties zou mogelijk explicieter kunnen (bv ceo:aantal). Hierdoor kan de Linked Data beter begrepen worden door gebruikers. (versie 2)
18. Een constrain is: ceo:heeftTijdvak en ceo:heeftPeriode kunnen niet gelijktijdig worden gebruikt bij dezelfde instance ceo:Datering. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.g1aowiokc6rn)) Hierdoor kan de data beter worden gechecked op juistheid de volledigheid waardoor betere garanties worden gegeven over de datakwaliteit.
19. Als ontoloog wil ik onderzoek doen naar de gebruik van de importeerde standaard vocabulaires, zodat er geen ongebruikte constructen wordt opgenomen in de ontologie. (versie 2)
20. De mogelijkheid voor het opnemen van meerdere srs modelleren dan RD. (versie 2)
21. Er zijn properties in het logische model die een expliciete kardinaliteit kennen, dit is nog niet gemodelleerd. Onderzoek naar de modellering van de kardinaliteit (1, 1-n, 0-1 etc.) per construct. M.a.w. Het bepalen van verplichte en optionele onderdelen Dit kunnen we modelleren mbv owl, SHACL, we kunnen ook tekstueel documenteren met aanvullende visualiseren in visio. Hierdoor kan de data beter worden gechecked op juistheid de volledigheid waardoor betere garanties worden gegeven over de datakwaliteit. (?)
22. Het gebruik van afkortingen (bv BAG) in de naam van classes en properties onder de loep nemen. Zodat de Linked data beter leesbaar wordt voor de gebruikers en we consequent voldoen aan de documentatiestandaard. Eventueel gebruiken equivalent property: best of both worlds. (versie 2)
23. Ontologie vertalen naar het Engels. Hierdoor kan de Linked Data beter worden gebruikt en begrepen kan worden door internationale gebruikers.Bepalen of de strings in de data voorzien moeten worden van ‘@nl’. Hierdoor kan de Linked Data beter begrepen worden door gebruikers. Daardoor is er in de data te filteren op taal. (versie 2)
24. Onderzoeken het wel of niet opnemen van domain/range in de RCE ontologieën. (versie 2)
25. Duidelijk moet worden hoe het versiebeheer van de ontologie moet worden ingericht, zodat verbeteringen en uitbreidingen op een gecontroleerde wijze worden doorgevoerd. (voor versie 2)
26. Aanpak voor Betrokkene modellering bedenken en onderbouwen in het licht van de AVG.
27. Omzetten eigenschapstype instanties van ztco naar ztceo.
28. De kardinaliteiten en specialisatie-relatie zijn in het conceptuele model ZAAK niet goed getekend de ZO wijkt hier dus bewust van af. Betreft dit zaakdocumenten?

# 

# Architectuur uitgangspunten voor de Linked Data Voorziening (LDV)

Deze paragraaf bevat uitsluitend die uitgangspunten die van belang zijn voor de modellering en de ETL.

Met behulp van de RCE eigen ontologieën wordt de data uit de RCE datalaag in de Linked Data Voorziening (LDV) gemodelleerd.

**De data in de LDV beoogd uiteindelijk een 100% volledige, juiste en actuele representatie te zijn van de data in de datalaag van de RCE.** Dit is een belangrijke vereiste. Zonder deze overeenkomst tussen LDV en datalaag zouden er bij de RCE twee verschillende werkelijkheden gaan bestaan. Dat zou dan consequenties hebben voor het gebruik van deze informatie in de de werkprocessen van de RCE en derden.

De data in de datalaag van de RCE is de administratieve werkelijkheid van de de RCE. Deze kan en hoeft ook niet noodzakelijkerwijs overeen te komen met de administratieve werkelijkheid van andere organisaties of de actuele ‘materiële’ werkelijkheid.

Hiervoor bestaan een aantal redenen:

1. datakwaliteit;
2. verschil in tijd;
3. verschil in betekenis.

Ad A. De ontstaansgeschiedenis van gegevens van de RCE gaat terug tot het analoge tijdperk. Gebruik en semantiek van de gegevens zijn door de tijd gewijzigd en niet altijd actueel gehouden.

Ad B. Het moment waarop een wijziging t.o.v. een andere administratie of werkelijkheid bekend wordt en verwerkt is bij de RCE kan verschillen.

Ad C. Het kan zijn dat de RCE bewust een andere betekenis geeft aan informatie in de datalaag, zodat deze (beter) aansluit bij de praktijk en zienswijze van de RCE.

# Uitgangspunten en ontwerpbeslissingen voor de RCE ontologieën

## Algemene RCE uitgangspunten voor het modelleren van ontologieën

1. We modelleren zodanig dat de gegevens in de LDV een juiste volledige representatie is van de data in de datalaag, Dit noemen we aanbod gedreven modellering.(Ten behoeve van de verschillende gebruikers(groepen) en toepassingen voorzien we in de toekomst ook een vraag gedreven modellering. Deze aanpak draagt bij aan flexibiliteit en snelle realisatie van nieuwe informatiebehoefte van gebruikers.)
2. De ontologie moet leesbaar zijn voor mensen
3. De ontologie is logisch (statements zijn met elkaar in overeenstemming)
4. De ontologie is consistent (zie documentatiestandaard en ontwerpbeslissingen)
5. Tijdelijke modellering van andere datasets gebeurt in afzonderlijke andere ontologieën. Zie bijvoorbeeld de ‘kerken-sheet’ t.b.v. de PoC (kpo) hierin zou dan de RCE ontologieën geïmporteerd kunnen worden.
6. Logische modellen zijn leidend voor de modellering van de ontologie en niet de technische modellen, zodat er geen afhankelijkheid ontstaat met de huidige implementaties en de ontologie en daarmee de Linked Data duurzaam is. Dit impliceert dat in de ontologie ook classes worden opgenomen die (nog) geen technische implementatie kennen.
7. We maken gebruik van bestaande vocabulaire waar nuttig.
8. De URI’s van de Linked Data moeten voldoen aan URI-strategie en worden aldaar beschreven.
9. Voor de leesbaarheid van de ontologie worden de domein en range van properties aangegeven. De wenselijkheid van dit uitgangspunt moeten we later nogmaals beoordelen. Indien een property meerder classes heeft in het domein of range, dan moet de juiste afleiding worden gecontroleerd! ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.g1aowiokc6rn)) (N.b. Het oorspronkelijk standpunt was dat we geen *domein* en *range* modelleren om ongewenste afleidingen te voorkomen, tenzij onomstotelijk vast staat dat dit noodzakelijk is.

## Afspraken over RCE werkwijze voor het modelleren van de ontologieën

1. De visuele weergave van de ontologie is leidend, omdat dit de betekenis van het model het beste communiceert. Wijzigingen en aanvullingen worden dus hierin als eerste aangebracht. Dit impliceert dat alle andere versies (bv tekstuele weergave in een editor) getoetst moeten worden aan de visuele weergave. Constructen worden eerst visueel gemodelleerd in VoWL en na een review, indien nodig met een (externe) expert, nemen we de constructen op in RDF.
2. De ontologieën maken gebruik van geïmporteerde controlled vocabularies. Als een externe controlled vocabulary wordt gebruikt dan wordt die in zijn geheel overgenomen, zodat altijd duidelijk is dat de overgenomen controlled vocabulary volledig is. Niet gebruikte constructen uit de externe controlled vocabulary blijven in de ontologie behouden vanuit het perspectief van volledigheid tijdens het modelleren van de ontologie.

### Aanpak ontologie validatie

Controleer overeenkomst tussen:

1. Logische model - Visio
2. Visio - Logische model
3. Visio -TopBraid
4. TopBraid - Visio
5. Documentatie

Aandacht heeft:

1. Controleer de generieke benaming van attributen en relaties in het logische model omzetten naar een specifieke semantiek benaming in de ontologie! Omdat we een range aangeven tbv leesbaarheid en gebruik.
2. Controleer de kardinaliteit per construct.

## De RCE ontwerpbeslissingen

1. In de definitieve gepubliceerde ontologie staan alleen de gebruikte constructen uit de gebruikte controlled vocabularies.
2. Voor de menselijke leesbaarheid moet de ontologie beschikken over: rdfs:label, rdfs:comment en de afgesproken annotaties en daarnaast toelichting en visualisatie in de documentatie
3. In de ontologie worden geen blank nodes gebruikt, tenzij een constructie niet op een andere manier te modelleren is. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.g1aowiokc6rn)).
4. De ontologie is in het Nederlands, maar heeft nog geen expliciete taal aanduiding, totdat het vocabulaire meertaligheid krijgt. Uiteindelijk moeten de ontologieën meertalig worden. (Iedere datatypeProperty met als range een string krijgt “@nl”, zolang er nog geen meertaligheid in de data voorkomt. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.ej4chn8hksx6))
5. Alle classes en properties met RCE data worden onder de eigen namespace geplaatst, vanwege de semantische herkenbaarheid en de mogelijkheid om externe properties in het eigen class domain te herdefiniëring. Dit staat in contrast met het linked data principe om standaard vocabulaires her te gebruiken. Om hierin te voorzien is het uitgangspunt gedefinieerd: Om in de CEO en ZAAK standaard custom classes en properties te modelleren, ook wanneer er sprake is van een geschikte externe vocabulaire. In dit geval wordt er vanuit de eigen class of property een relatie aangelegd met het equivalent uit het standaard vocabulaire d.m.v. rdfs:subClassOf ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.g1aowiokc6rn)) en rdfs:subPropertryOf ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.g1aowiokc6rn)).
6. We modelleren voor een class alleen constructen die van toepassing zijn op alle instanties van die class. (De houdbaarheid en wenselijkheid van deze afspraak moeten we nog vaststellen.) Bijvoorbeeld: We modelleren ceo:CultuurhistorischObject niet als gsp:Feature omdat niet ieder CHO object een geometrie heeft. Indien dit wel het geval is dan volgt dit automatisch uit de aanwezigheid van de property gsp:hasGeometry.
7. Materiële en formele historie: Standpunt 10-2018 geen historie in ontologie wegens onduidelijkheid over de lifecycle van de data. (In de ontologie wordt de materiële en formele historie van de data (voorlopig) gemodelleerd als vastgelegd in de Kimomo database. Deze constructie wordt gezien (27-3) als een tussenstap, want: het is ons onbekend wanneer history wijzigt in de actuele data en ook niet hoe dit zou moeten zijn in de gewenste situatie. We gaan in deze constructie nog niet uit van “graafversies”. Daarom blijven we dicht bij het conceptuele model.)
8. Als de range van een dataproperty een Boolean is ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.mu5lf07yw0kg)) dan wordt alleen een triple gemaakt als de waarde van de Boolean “true” is.
9. In de ontologie hebben alle classes en properties een rdfs:label, als leesbare resource-naam ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.fc3xdarxvkpn)).
10. In de data hebben instanties een property, die het mogelijk maakt om een persistent nummer te geven aan de betreffende resource:
    1. Voor de subclasses die vallen onder de classes ceo:CultuurhistorischObject en ceo:Kennisregistratie is dat respectievelijk de property: ceo:cultuurhistorischObjectnummer ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.fc3xdarxvkpn)) en ceo:kennisregistratienummer ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.fc3xdarxvkpn)) ;
    2. Voor de andere classes in de database hebben de tabellen een eigennummer-property, bijvoorbeeld bij ceo:Samenhang is dat de property ceo:samenhangnummer. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.fc3xdarxvkpn))
11. Inferred en inserted triples: Uitgangspunt (10-2018): de inferred triples worden gemaakt in de ETL, zodat de RCE in control is over de LD. (Het moet nog worden bepaald welke triples dit zijn (10-2018). Dit wordt verder beschreven in een afzonderlijk document: <https://docs.google.com/document/d/10HvFkuGl1RzYq52XMT-MktqMTnQv178yC9zHc9kUzGY/>
12. Ieder (data)instantie wordt voorzien van een rdfs:label, die bestaat uit: de rdfs:label van de klasse waaronder de instantie valt, gevolgd door een dubbele punt, spatie en het refentienummer. Bijvoorbeeld: https://linkeddata.cultureelerfgoed.nl/cho-kennis/id/actorenrol/55605>

rdfs:label “Actor en rol: 55605";

Dit is de minimale varia, eventueel uitbreiden naar een meer betekenisvolle label.

1. Een property wordt gespecificeerd c.q. opgesplitst in meerdere properties. Wanneer er sprake is van meer dan 1 domain en er verschil in betekenis is tussen de oorspronkelijke property en zijn verschillende domains. Bijvoorbeeld: Zowel bij het domain ceo:Vondsten als bij ceo:Grondsporen kwam de property ceo:aantal voor, omdat het in het ene geval gaat om aantal vondsten en in andere geval om aantal grondsporen zijn er twee aparte properties aangemaakt: ceo:aantalVondsten en ceo:aantalGrondsporen.

#### 

## De RCE ontologie documentatiestandaard

1. Er wordt zo weinig mogelijk gebruik maakt van afkortingen. Bijvoorbeeld: niet CHO, maar CultuurhistorischObject.
2. Constructen (classes en properties) in de ontologie krijgen een ‘eigen’ omschrijving, waarbij zoveel mogelijk geparafraseerd wordt uit logische datamodellen. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n))
3. De property skos:editorialNote wordt gebruikt om opmerkingen over gebruik en nog te nemen beslissingen over classes en properties te plaatsen.
4. De annotatie property dcterms:created ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n)) geeft de datum weer waarop het construct is opgenomen in de ontologie. Met dcterms:modifiied wordt de datum van een wijziging aangegeven.
5. We modelleren zo specifiek mogelijk w.b.t. het gebruik van semantische constructen uit RDF, OWL e.a. Bijvoorbeeld: We geven de voorkeur aan datatype string boven het datatype literal[[1]](#footnote-0)
6. We geven een owl:ObjectProperty de prefix ‘heeft’ of ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n))
7. en een owl:DatatypeProperty geen prefix.([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n))
8. In de naamgeving van een owl:DatatypeProperty blijven we zo dicht mogelijk bij het datamodel.
9. Opbouw van de rdfs:comment:
   1. Classes worden omschreven als: De class ... is de representatie van … {@nl} ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n))
   2. Een owl:ObjectProperty worden omschreven als: De property ... legt de relatie tussen … {@nl} ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n))
   3. Een owl:DatatypeProperty worden omschreven als: De property ... wordt gebruikt om aan te geven…{@nl} ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n))

Typografie gebruik bij naamgeving van classes en properties:

1. Een class schrijven we in UpperCamelCase. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n))Voorbeeld ceo:CultuurhistorischObject.
2. Een dataproperty schrijven we met een kleine letter aan het begin. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n)) Voorbeeld ceo:huisnummer. Uitzondering zijn gangbare afkortingen zoals BRK.
3. Een objectproperty laten we met ‘heeft’ beginnen en wordt geschreven in lowerCamelCase. ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.rc6hbtlcwv2n)) Voorbeeld ceo:heeftAdres.
4. Een instance van een Class schrijven we met kleine letters.
5. De namen van classes en properties bestaat alleen uit letters ([testquery](https://docs.google.com/document/d/1n9xO50E7O2W0wviyLbIcbo4yF2ZNeAkRh_G0zeqaIW8/edit#bookmark=id.tg38bspn58r6)).

### De visualisatie standaard

De ontologieën worden gevisualiseerd met behulp van de VOWL-standaard. Deze paragraaf beschrijft de aanvullende afspraken die moet zorgen voor een consequente visualisering in MsVisio. Dit heeft betrekking op zaken als: kleur, lettertype, uitlijning, harde return, schrijfwijze, lettergrootte, richting van de pijl, naamgeving tabbladen, etc

1. De tabbladen hebben een prefix als volgt:

A = Algemeen -> ontologie onafhankelijk

C = Worden alleen classes, attributen en relaties uit het CHOI en relaties met KENNIS of ZAAK indien van toepassing.

K = Worden alleen classes, attributen en relaties uit het KENNIS en relaties met CHOI of ZAAK indien van toepassing.

Z = Worden alleen classes, attributen en relaties uit het ZAAK en relaties met KENNIS of CHOI indien van toepassing.

X = Nog in ontwikkeling

1. We tonen een constructie eenmaal, tenzij een volledigheid gewenst is.
2. Voor iedere class is (maximaal) één tabblad opgenomen die alle DatatypeProperties en ObjectProperties van die class weergeeft.

# 

## Uitgangspunten m.b.t. het modelleren van kardinaliteit

Deze paragraaf beschrijft de modellering van de kardinaliteiten van de properties in de RCE ontologieën.

Het gaat hierbij om de voorkomens in de Linked Data en niet in het bronsysteem, omdat niet altijd voor alle items in de database ook triples worden gemaakt. Denk bijvoorbeeld aan triples met als object een boolean deze is alleen aanwezig als de waarde ‘true’ is.

Logisch volgt hieruit dat, indien de range een boolean dat dan de kardinaliteit altijd 0-1 is.

De volgende kardinaliteiten worden onderscheiden.

Range is een literal

0..1 functional propertie?

0..\* geen restrictie

1..1

Range is een resource

0..1 functional propertie?

0..\* geen restrictie

1..1

2..\* een complex kent altijd minimaal 2 rijksmonumenten.

We overwegen een property ceo:kardinaliteitComment als subproperty van rdfs comment voor het administratief vastleggen van de kardinaliteit. Naast de semantische modellering.

Naast kardinaliteit van de property moet ook gekeken worden naar:

* De logische OR bijvoorbeeld ceo:Datering heeft altijd een periode of tijdvak.
* Constraint:Een (subclass van) Kennisregistratie behoort altijd tot een CHO. zie visio K -class.

# 

# De betekenis van concepten

In de ontologie willen we de relatie tussen en de betekenis van de concepten vastleggen en beschrijven.

We onderscheiden de volgende voorkomens van concepten in verschillende omgevingen:

1. Concept in ontologie
2. Concept in de logische datamodel
3. Concept in thesaurus
4. Betekenis van het concept volgens de (erfgoed)wet
5. Concept in uitwisselmodellen

Het zijn dezelfde termen, maar kennen hun eigen definities. Voorbeeld: het concept “rijksmonument”.

In de ontologie wordt een class aangemaakt die “rijksmonument” heet. Binnen de ontologie wordt deze class en het gebruik uitgelegd. Een omschrijving zou kunnen zijn: Een subclass van de class CHO die de subclass rijksmonument representeert. Voor een semantisch expert die met de ceo aan de slag gaat, is dit voldoende informatie om te weten dat hier het rijksmonument gevonden kan worden.

In het logische datamodel CHOI wordt “rijksmonument” uitgelegd als: “Onroerende cultuurhistorische objecten die als Rijksmonument zijn aangewezen. Dit kunnen zowel onroerende cultuurhistorische objecten als archeologische terreinen zijn.” Voor bijvoorbeeld een developer die met de database aan de slag wil is dit voldoende informatie om met de data te kunnen werken.

In de CultureelErfgoed Thesaurus van de RCE staat “rijksmonument” omschreven als: *‘Een rijksmonument is in Nederland een zaak (een bouwwerk of object, of het restant daarvan) die van algemeen belang is wegens de schoonheid, de betekenis voor de wetenschap of de cultuurhistorische waarde.'* Hier is voor deze omschrijving gekozen omdat de thesaurus door iedereen gebruikt kan worden en een duidelijke uitleg moet geven over de betekenis van een concept.

Verder worden concepten gebruikt in uitwisselmodellen. In dit stadium is nog niet bekend wat de best practice daar zal zijn.

Tenslotte is er nog de (erfgoed)wet. De RCE is een overheidsdienst en voor de RCE is de wettekst de betekenis van dit concept.

Rijksmonument: *monument* of *archeologisch monument* dat is ingeschreven in het *rijksmonumentenregister*;

*monument*: onroerende zaak die deel uitmaakt van cultureel erfgoed

*archeologisch monument*: terrein dat deel uitmaakt van cultureel erfgoed vanwege de daar aanwezige overblijfselen, voorwerpen of andere sporen van menselijke aanwezigheid in het verleden, met inbegrip van die overblijfselen, voorwerpen en sporen;

*rijksmonumentenregister*: register als bedoeld in [artikel 3.3](http://wetten.overheid.nl/BWBR0037521/2017-09-01#Hoofdstuk3_Paragraaf3.1_Artikel3.3);

NB niet ieder concept heeft deze “wet component”.

Conclusie

De aanname is dat deze betekenissen náást elkaar kunnen en moeten bestaan, als ze maar uitleg krijgen van de context.

# Annotatie van de status van de gehele ontologie

dcterms:created te gebruiken als aanmaakdatum

dcterms:modified te gebruiken als wijzigingsdatum en

dcterms:valid te gebruiken als gelding sinds datum.

dcterms:creator te gebruiken als aanmaker

dcterms:hasVersion te gebruiken als versienummer, voorbeel 0.1, 0.2, 0.21, etc.

dcterms:license te gebruiken als onder licentie van.

dcterms:publisher te gebruiken als uitgever van.

dcterms:title te gebruiken als titel van.

Daarnaast is er nog het vocab “vs”. vs staat voor SemWeb Vocab Status Ontology.

In deze ontologie is er een property (totaal maar 3) ns:term\_status.

We zouden deze kunnen gebruiken voor de statussen van de verschillende classes / properties. Met enkele vastgestelde waarden als: candidate, unstable, testing, valid, depricated. (oid)

Nb dit is ook nodig voor de dataset bv met void. O.a. i.v.m. het NDE register.

Status (voorstel)

Welk annotatie property kunnen we hiervoor gebruiken?

Van ieder construct wordt de status in de PoC bijgehouden als volgt:

|  |  |
| --- | --- |
| **Status** | **Toelichting** |
| Proposed | candidate |
| Approved | unstable |
| Implemented | testing |
| Validated | valid |
| Deprecated | *deprecated* |

#### Aanzet beschrijving versiebeheer irt named graph

Wat is kardinaliteit tussen ontologie en data?

Casus

Start conditie:

* Geen wijzigingen in de ontologie.
* Wel wijzigingen in de data

En de ontologie en data in twee verschillende named graphs.

Ontologie: https://linkeddata.cultureelerfgoed.nl/def/ceo

Data: https://linkeddata.cultureelerfgoed.nl/cho-kennis

Start met een initiële load van de gehele dataset

Daarna een ‘delta’s met:’

* Nieuw
* Wijzigingen
* Verwijderingen

# 

# Glosarium

|  |  |
| --- | --- |
| *Term* | *Betekenis* |
| Construct | Verzamelterm voor classes en properties die zijn opgenomen in de ontologie. |
|  |  |
|  |  |

# Bijlagen

### Sjabloon voor classes en properties

<https://docs.google.com/document/d/1N9tSmxUMYYsIc_tl6RlpoSzmjBcqXcPXfDPcCtuYjm8/edit?usp=sharing>

### Versie planning en ToDo

<https://docs.google.com/document/d/1C_mD2-wuiICzTAkfVpf8739JOm7IZnSyaH7mPkqjirk/edit?usp=sharing>

### Cultureel Erfgoed Ontologie (CEO)

[https://docs.google.com/document/d/1-AMp7VcOO6axMYTSxwU6imv42dgiX2P-\_LLaCJfnqS8/edit?usp=sharing](https://docs.google.com/document/d/19y00TWb-fYBQrIIyanY8kBBIYRsrTOaZ4ZB8lo3N0IU/edit?usp=sharing)

### De ZAAK ontologie (ZO)

<https://docs.google.com/document/d/1snSnt2mlbRy6V0QowLNvnt4gBlqtpdU2tOj4Vx0DLPk/edit?usp=sharing>

### Rest materiaal ontologie modellering

<https://docs.google.com/document/d/1WBiK0ZJD0NdGi8z5sJwc8SzuVUfmTYJMf0uj3DZdo7E/edit?usp=sharing>

1. Het begrip Literal is een verzamelnaam en staat voor alles dat geen resource is. Het stamt uit de tijd dat owl:datatypeProperty nog niet bestond. Dus het is minder specifiek dan string, het kan ook een nummer of datum zijn. Je hebt plain literals, dat zijn tekst literals, en typed literals, die hebben een xsddatatype erbij gespecificeerd, dus bv integer. Zie <https://www.w3.org/TR/rdf-concepts/#section-Literal-Value> [↑](#footnote-ref-0)