

# Richtlijnen

## Inrichting Canonical Scheme PostNL

Versie 3.0:  
Aanpassingen/ aanvullingen per 2020.

1 oktober 2011

|         |                                                |
|---------|------------------------------------------------|
| Auteur  | Heleen de Boer, Nelleke van der Meer           |
| Tel     | +31 (0)651519319, +31 (0)620 82 86 37          |
| Rapport | Richtlijnen Inrichting Canonical Scheme PostNL |
| Versie  | 1.0                                            |

## Inhoudsopgave

|    |                                                                     |
|----|---------------------------------------------------------------------|
| 1  | Inleiding                                                           |
| 2  | Doel                                                                |
| 3  | Positionering                                                       |
| 4  | Inrichtingskeuzes: Namespaces                                       |
| 5  | Inrichtingskeuzes: Metamodel                                        |
| 6  | Inrichtingskeuzes: Versiebeheer                                     |
| 7  | Inrichtingskeuzes: Database en Mappenstructuur                      |
| 8  | Inrichtingskeuzes: Koppeling CS met ARIS en ESR.                    |
| 9  | OAGIS: Gebruik OAGIS Business Object binnen Canonical Scheme PostNL |
| 10 | Gerelateerde documenten                                             |

## Versie Informatie

| Versie              | Datum                      | Bijzonderheden                                            | Auteur                                                   |
|---------------------|----------------------------|-----------------------------------------------------------|----------------------------------------------------------|
| 0.1                 | 11-04-2011                 | Versie besproken met Linda Terlouw (CS expert)            | Nelleke van der Meer/<br>Heleen de Boer                  |
| 0.2                 | 13-04-2011                 | Versie ter besluitvorming aan project/stuurgroep          | Nelleke van der Meer/<br>Heleen de Boer                  |
| 0.3                 | 06-07-2011                 | Aangepast aan meest recente inzichten en besluitvorming.  | Nelleke van der Meer/<br>Heleen de Boer                  |
| 0.4                 | 22-07-2011                 | Opmerkingen op versie 0.4 verwerkt                        | Nelleke van der Meer/<br>Heleen de Boer                  |
| 0.9                 | 01-09-2011                 | Oplevering aan stuurgroep ten behoeve van finaal akkoord. | Nelleke van der Meer/<br>Heleen de Boer                  |
| 1.0                 | 01-10-2011                 | Akkoord stuurgroep op 23-09-2011.                         | Nelleke van der Meer/<br>Heleen de Boer                  |
| <a href="#">1.1</a> | <a href="#">30-12-2012</a> | <a href="#">Wijzigingen tijdens gebruik</a>               | <a href="#">Nelleke van der Meer/<br/>Heleen de Boer</a> |

## Verzendlijst

| Naam                | Bedrijf/Functie       |
|---------------------|-----------------------|
| Peter Vos           | Manager Architectuur  |
| Henk Willemse       | Informatie Architect  |
| Peter Schoneveld    | Informatie Architect  |
| Nelleke van de Meer | Architect             |
| Heleen de Boer      | Applicatie Consultant |

Richtlijnen Inrichting Canonical Scheme PostNL

# 1

## Inleiding

Dit document beschrijft de opzet en werking van het product Canonical Scheme PostNL (CS PostNL). Het CS PostNL is gepositioneerd als zijnde een van de producten vanuit Architectuur.

Het Canonical Scheme van PostNL bestaat uit een bibliotheek van XSD files met specificaties van de Fields, Components en Business Objecten die gezamenlijk het Canonical Scheme voor PostNL vormen.

Het Canonical Scheme is basis voor berichtuitwisseling tussen applicaties en/ of (op een hoger abstractie niveau)domeinen.

Door gebruik te maken van een Canonical Scheme is het mogelijk binnen SOA met business service te werken en zo een ontkoppelpunt te maken binnen het SOA landschap.

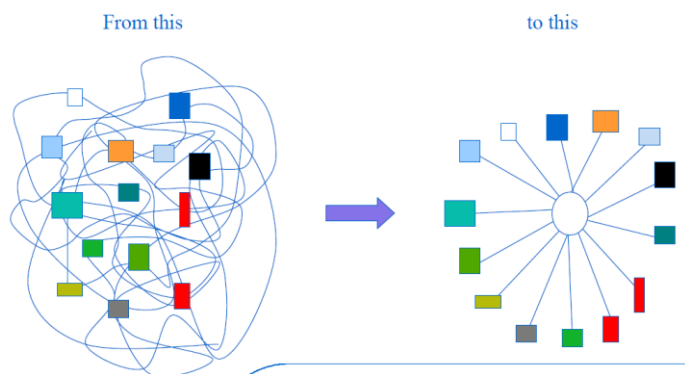
Daarnaast biedt het Canonical Scheme een gemeenschappelijk referentiekader van waaruit de aan de business te leveren services worden gedefinieerd. Door deze (business) services te baseren op het Canonical Scheme is maximale ontkoppeling mogelijk van de op het interne applicatie datamodel gerichte services.

In dit document wordt ingegaan op de richtlijnen voor de inrichting van het Canonical Scheme PostNL.

## 2

### Doel

Een Canonical Scheme is een gedeeld dataformaat voor uitwisseling van data tussen applicaties en/of business domeinen. Het doel van het Canonical Scheme is om te voorkomen dat er vertalingen tussen alle individuele formaten van objecten plaatsvindt. Door een Canonical Scheme vindt er vertaling tussen elk formaat naar het Canonical Scheme plaats.



### Doelgroepen

Het product Canonical Scheme wordt opgeleverd aan de manager van Business IT.

De eigenaar en daarmee eindverantwoordelijke voor het Canonical Scheme is de manager van de afdeling Architectuur.

Beheer en exploitatie van het Canonical Scheme wordt uitgevoerd door Informatie Architecten CS.

Ontwikkeling en beheer van Business Objecten wordt uitgevoerd door de Informatie Architect CS, in samenwerking met Informatie Architecten van de domeinen en de Informatie Analisten. Review wordt uitgevoerd door de afdeling Architectuur.

De initiële versie van de inhoud wordt ingevuld door de projectgroep; verdere vulling vindt plaats vanuit projecten en/ of vanuit de bedrijfsfunctie Informatie Architectuur.

Doelgroep voor het gebruik van het Canonical Scheme zijn externe leveranciers of interne afdelingen die services of applicatie maken.

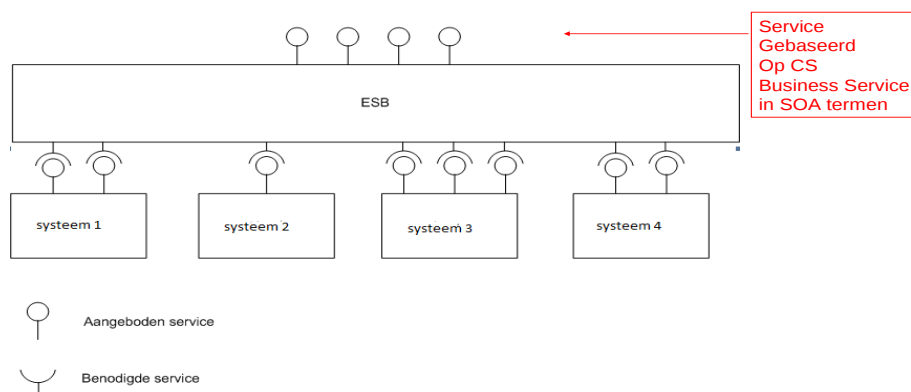
## **Uitgangspunten**

Uitgangspunten bij de inrichting en onderhoud van het Canonical Scheme PostNL:

- Het Canonical Scheme PostNL (CS PostNL) is gebaseerd op OAGIS 9\_5. Daarom zijn de richtlijnen voor OAGIS ook van toepassing op het CS PostNL. Deze richtlijnen zijn vastgelegd in de NDR: Naming and Design Rules Standard (30 sept 2005)
- De laatste versie van OAGIS 9\_5 dient als basis voor het CS PostNL. Eventuele versies die volgen zullen wel dienen als bron van inspiratie maar niet worden geïmporteerd in het CS van PostNL. Dit komt met name door maatwerk aanpassingen ten behoeve van de functionele eisen vanuit de business en omdat het CS PostNL de mogelijkheid biedt tot het verrijken van Fields met constraints in de oorspronkelijk XSD file van OAGIS. Objectdefinities gebaseerd op de marktstandaard OAGIS sec worden niet geaccepteerd.
- BOD (Business Object Documenten) uit OAGIS worden niet overgenomen en ook niet gewijzigd. BOD zijn de volledige bericht definities (incl. technische info over bericht verwerking) op basis van Nouns die wel worden overgenomen in het CS NL. De volledige berichtdefinities worden echter binnen PostNL vastgelegd in de ESR van SAP PI in de vorm van MessageTypes
- Nouns binnen OAGIS worden Business Objecten binnen CS PostNL genoemd. Nouns worden pas in CS PostNL gekopieerd naar Business Object op het moment dat ze onder architectuur verder worden gedefinieerd. Zo is het duidelijk voor welke objecten PostNL een CS PostNL heeft
- In het CS PostNL moet zichtbaar cq duidelijk zijn welke Components en Fields inmiddels aangepast zijn ten opzichte van de oorspronkelijke OAGIS definitie. Nb. Op moment van oplevering CSPostNL is dit report nog niet gerealiseerd.
- Nieuwe Business Objecten ontstaan in ARIS vanuit proces- en informatiearchitectuur en worden vervolgens onder architectuur gedefinieerd in het CS PostNL.
- De definities (structuur/ inhoud) van de Business Objecten moeten zo ver mogelijk functioneel afgestemd zijn met vertegenwoordigers van de gebruikers. Informatie over deze afstemming (met wie, wanneer, voor welk doel) moet voor ontwerpers beschikbaar zijn.
- De inrichting van Aris – ESR – CS moet op elkaar aansluiten zodat een totaalbeeld ontstaat in de samenhang van processen, Business Objecten, services, applicaties en data.
- Het Canonical Scheme bestaat niet alleen uit de structuur van objecten maar ook uit begrippen en definities.
- Het Canonieke Datamodel van de ESR is al in gebruik terwijl het CS PostNL nieuw wordt opgebouwd (obv OAGIS). De Business Objecten die al in de ESR aanwezig zijn, zijn onderdeel van de definities in het CS PostNL. Nb. Op moment van oplevering van het CSPostNL is dit nog niet gerealiseerd.
- Publicatie van het CS PostNL aan externe partners vindt plaats op basis van need-to-know.
- Bij het publiceren van het CS extern (bv tbv externe partners) is het wellicht noodzakelijk een "echte" URL te maken van de gebruikte namespaces. Het maken van deze URL valt buiten het project. Als work-around zullen de onderliggende Componenten en Fields in een bestand worden meegeleverd. Een goed ingericht versie beheer is hiervoor wel van belang.

- Eisen ten aanzien van koppeling CS –ESR:
  - Business Objecten in de ESR moeten overeenkomen met Business Objecten in het CS PostNL
  - De structuur van BON's in de ESR moet overeenkomen met de structuur van Components in het CS PostNL.
  - Naamgeving en definities van gegevens in de ESR moet overeenkomen met die in het CS
  - Afwijkingen/ aanvullingen in de ESR zijn toegestaan vanwege:
    - releasemanagement (in het CS staat de nieuwe gewenste versie)
    - het ten behoeve van detail ontwerp verder verfijnen en concretiseren van data
    - eisen vanuit vertaling naar executie

De berichten die services onderling uitwisselen mogen alleen gebruik maken van definities uit het Canonical Scheme. Binnen SOA worden hiervoor zgn. business service gebruikt, deze zijn applicatie onafhankelijk.



## 3

### Positionering

Het Canonical Scheme wordt gevormd door het geheel van bij elkaar behorende sets van business informatie, bestaande uit herbruikbare componenten, fields en data typen

Het speelt een rol bij organisatieverandering, procesinrichting en systeemontwikkeling.

Het Canonical Scheme heeft raakvlakken met:

#### Verandermanagement

- het betrokken organisatieonderdeel van Post NL en eventueel in/ externe betrokken partijen hebben in een vroegtijdig stadium een gemeenschappelijk beeld van de beschikbare en vereiste informatie;
- het vroegtijdig identificeren van eventuele nog op te lossen knelpunten en vraagstukken ten aanzien van de vereiste informatievoorziening;
- een Soll situatie met betrekking tot de vereiste informatievoorziening waarop het Veranderplan zich kan richten;

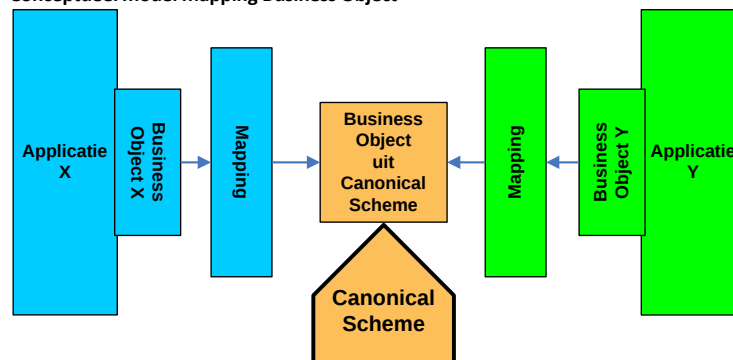
#### Procesinrichting

- het Canonical Scheme waarborgt bedrijfsbrede (lees Mail in NL) inrichting en toegang tot een bibliotheek van gestandaardiseerde informatie sets (=object definities) waarop de procesinrichting zich kan baseren

#### Systeemontwikkeling

- het Canonical Scheme geeft richting aan het gegevensgebruik van de door applicaties geboden services
- mede op basis hiervan wordt invulling gegeven aan het service compositiemodel en het detail ontwerp van services
- het Canonical Scheme regelt het formaat waarin data tussen systemen/ services wordt uitgewisseld

Conceptueel model mapping Business Object



### **Omgevingen**

- Er worden twee omgevingen ingericht tbv het CS PostNL: een development omgeving en een productie omgeving.
- De productie resp. ontwikkelomgeving is beschikbaar voor alle daartoe geautoriseerde gebruikers.
- De development omgeving zal worden gebruikt tbv ontwikkelen nieuwe Business Objecten en aanpassen bestaande Business Objecten, Componenten en Fields.

## 4

### Inrichtingskeuzes: Namespaces

#### Inleiding

Binnen een XSD is het gebruik van namespaces van belang:

*Een **naamruimte** (Engels namespace ) is een mechanisme dat in veel computertalen de mogelijkheid biedt om namen van variabelen, constanten en functies (identifiers) zodanig te groeperen dat ze binnen verschillende contexten een (volstrekt) afzonderlijke betekenis en werking krijgen. Bron Wikipedia*

Binnen het OAGIS model zijn de BOD's, Nouns, Components en Fields opgenomen binnen dezelfde namespace: <http://www.openapplications.org/oagis/9>. OAGIS zal dit principe in de volgende release loslaten en dan staan de verschillende files in een eigen namespace.

#### Inrichting CS PostNL

Voor PostNL zullen de Business Objecten, de Components en Fields aangepast worden en in een eigen namespace gezet worden.

- Business Objecten (uitsterren van gedeeltes + Business erbij),
- Components (uitsterren van gedeeltes + Components erbij) en
- Fields (toevoegen van details en Fields erbij)

De business onafhankelijk elementen zijn opgenomen in de desbetreffende standaard namespace. Dit zal niet wijzigen.

Ieder Business Object, Component en de Fields file worden in een specifieke namespace gezet.

Ten behoeve van versiebeheer is het major versienummer onderdeel van de namespace.

De namespaces van Business Objecten hebben geen prefix.

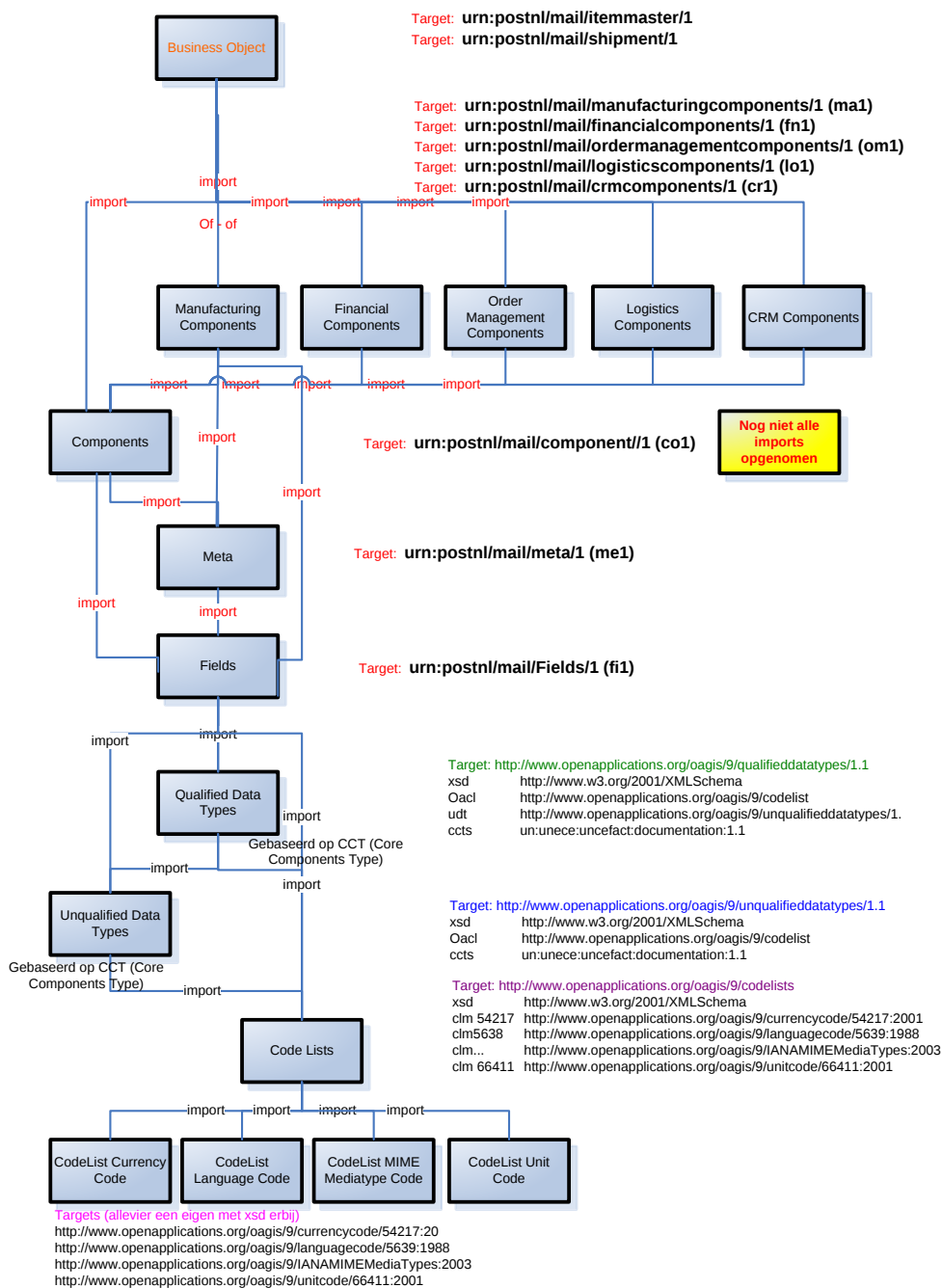
De namespaces van Components en Fields hebben wel een prefix. De prefix bestaat uit de eerste 2 letters van de naam van het XSD-file + het major versienummer.

Namespaces worden als volgt opgebouwd: urn:postnl/mail/<naam XSDfile>/<major versienummer>

Hierbij wordt er geen rekening gehouden met naamgeving in ESR SAP PI. Op dit moment is niet duidelijk of dit vereist is.

## Structuur CS MAIL NL XSD imports / includes incl namespaces

Heleen de Boer, 29 augustus 2011



## 5

### Inrichtingskeuzes: Metamodel

#### Inleiding

Het logisch metamodel van het Canonical Scheme van PostNL toont de logische structuur en samenhang van de objecten waaruit het Canonical Scheme is opgebouwd.

Het metamodel maakt zichtbaar:

- de keuzes die gemaakt zijn in de wijze waarop de verschillende objecten met elkaar samenhangen
- de elementen waaruit het object is samengesteld

#### Afwijking Metamodel OAGIS versus Metamodel CS PostNL

Het metamodel van het Canonical Scheme wijkt op sommige punten af van het door OAGIS gehanteerde metamodel (het model dat is afgeleid van de door OAGIS geïmplementeerde filestructuur).

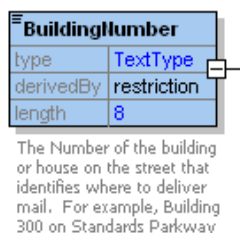
De verschillen zijn:

1. Structuur van het metamodel. Deze is in vergelijking met OAGIS eenvoudiger door het weglaten van de objecten Compound en Group. Dit betekent niet dat we bij hergebruik van OAGIS files het element Group niet hanteren, het betekent wel dat het object Group niet wordt onderkend als een logisch object in het metamodel.  
Compounds zijn in het geïmplementeerde model van OAGIS niet herkenbaar als een apart object.
2. Naamgeving objecten:  
Business Object in plaats van Noun
3. Het toekennen van details en constraints aan Fields.  
De Fields van OAGIS zijn niet voorzien van details en constraints. Dit is bijv. de lengte van het veld of andere kenmerken van een veld, afhankelijk van het type veld.  
bijv. straatnaam = string 10

Het Canonical Scheme kent in principe ook geen details en constraints toe aan Fields. Deze requirements moeten geregeld worden door de onderliggende applicaties.

Uitzonderingen zijn wel mogelijk, bijvoorbeeld als dit vanwege (nog niet concreet te benoemen redenen) wordt vereist door PostNL of als gekozen wordt voor implementatie van een marktstandaard die details en constraints voorschrijft (bijv. NEN-normering voor adressen).

Het toekennen van details en constraints gebeurt door het aanpassen oorspronkelijk OAGIS Field



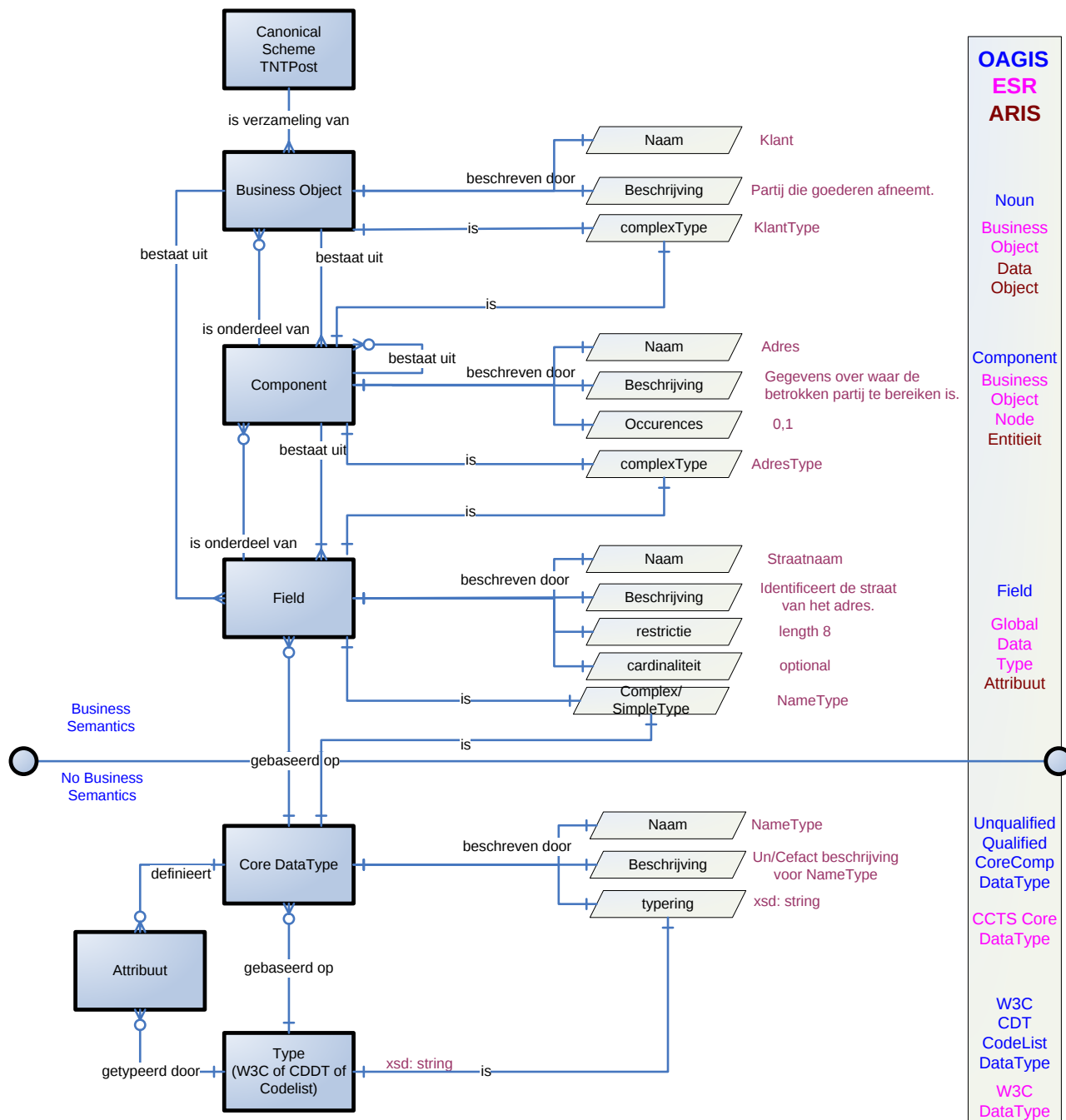
Bij het Field wordt de constraint/ details ingevuld.

Als blijkt dat hetzelfde Field als onderdeel van een ander Component of Business Object een andere restrictie heeft, dan is hergebruik niet mogelijk (= illusie van hergebruik) en moet een nieuw Field gemaakt worden.

#### Definities Metamodel

|                        |                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Business Object</b> | het Business Object is een eenheid van informatie die vanuit het perspectief van de business van PostNL relevant is.<br>Def Object in ARIS: Een object dat de gegevens bevat die nodig zijn om een business functie uit te voeren (bv Vrachtbrief) |
| <b>Component</b>       | Het Component beschrijft een groep van bij elkaar behorende gegevens die dienen als bouwsteen binnen een Business Object of binnen andere Componenten.                                                                                             |
| <b>Field</b>           | Een field is een kenmerk van een Component of een Business Object.                                                                                                                                                                                 |
| <b>Core Data Type</b>  | Een Data Type definieert de inhoud van gegevens (Field) zoals de weergave van de inhoud en toegestane waarden.                                                                                                                                     |
| <b>Attribuut</b>       | Een kenmerk dat het Core Data Type definieert.                                                                                                                                                                                                     |
| <b>Type</b>            | Het (wereldwijd) gestandaardiseerde gegevenstype waarop het Core Data Type is gebaseerd.                                                                                                                                                           |

Metamodel Canonical Scheme PostNL



#### Inzicht in gebruik van Components en Fields

Een van de uitgangspunten is:

*In het CS PostNL moet zichtbaar cq duidelijk zijn welke Components en Fields inmiddels aangepast zijn door PostNL.*

Tevens bestaat de wens om inzichtelijk te maken welke Components en Fields gebruikt worden door welke Business Objecten.

De mogelijkheid om dit te rapporteren is Xquery.

Nb. Beide punten staan nog open op moment van oplevering van CSPostNL.

## 6

### Inrichtingskeuzes: Versiebeheer

#### Inleiding

Versiebeheer is een noodzakelijke voorwaarde om veranderingen te verwerken in het Canonical Scheme en de daardoor ontstane versies op elkaar aan te sluiten.

Versiebeheer maakt onderscheid tussen major versies en minor versies. Wijziging van een major versie is nodig als de verandering niet compatible is met de vorige versie. Als de wijziging wel compatible is met de vorige versie, is sprake van een minor versie wijziging.

Bestaande applicaties en services mogen gebruik maken van vorige versies, nieuwe applicaties en services moeten hun object definities altijd baseren op de meest recente versie.

Er bestaan 2 typen versie compatibiliteit: backward en foreward.

Forward compatibiliteit betekent dat een oude versie gebruikt kan worden naast de nieuwe versie. Bijv.: toevoeging optionele elementen.

Backward compatibiliteit betekent dat de nieuwe versie van een object kan worden uitgerold zonder dat de bestaande object versies daardoor in de problemen komen.

Versie locatie:

Identificatie van de schema versie kan op verschillende wijzen:

1. gebruik van het interne schema versie attribuut
2. gebruik van een schema versie attribuut, gedefinieerd door de auteur in het root element
3. gebruik van het XML Schema target namespace
4. gebruik van de XML Schema Location in het schemaLocation attribute

Versienummering van WSDL's en XSD's is niet per definitie gerelateerd. Het is mogelijk dat een Canonical XSD door meerdere WSDL's gebruikt wordt en in een ander tempo wijzigt dan de WSDL's die deze XSD gebruiken.

## Versiebeheer Canonical Scheme PostNL

Het Canonical Scheme van PostNL is gebaseerd op OAGIS en volgt de wijze van versiebeheer van OAGIS.

### **SchemaLocation**

SchemaLocation wordt door OAGIS gebruikt voor versiebeheer van BOD's. BOD's vormen geen onderdeel van het Canonical Scheme, deze wijze van versiebeheer is dus niet relevant.

### **Attribuut in Root Element**

OAGIS gebruikt in het rootelement de attributen ReleaseID en VerionID voor versiebeheer van BOD's. BOD's vormen geen onderdeel van het Canonical Scheme, deze wijze van versiebeheer is dus niet relevant.

### **Namespace**

OAGIS legt in de namespace van Nouns, Components en DataTypes de major versie vast.

In de comment van het XSD Scheme staat de major en het minor versienummer genoemd.

Het Canonical Scheme van PostNL volgt deze wijze van versiebeheer.

In de Target namespace van Business Objecten, Components en Fields wordt de major versie vastgelegd.

Het major.minor versienummer van het XSD Scheme wordt vastgelegd in de comment van het schema.

Het major.minor versienummer van OAGIS dat vermeld staat in de comment kan ongewijzigd blijven en dient als referentie naar de oorspronkelijke OAGIS versie die diende ter inspiratie.

Daarnaast wordt in het Canonical Scheme het major versienummer ook toegevoegd aan de filenaam van het XSD.

Als de namespace van een XSD schema is voorzien van een prefix, is het major versienummer ook onderdeel van de prefix.

De XSD Schema's voor de Non Business Semantic data types worden overgenomen zoals ze zijn, inclusief de huidige namespaces en versienummers. Deze wereldwijde standaard is zeer stabiel en versiebeheer op deze XSD Schema's is niet nodig.

c. Werkwijze

**Versienummering bij oplevering Canonical Scheme**

Bij oplevering van het Canonical Scheme van PostNL door het project hebben alle XSD schema's het major versienummer 1 en het minor versienummer 0.

De namespace wordt gebruikt voor vastlegging van het major versienummer.

Het major en het minor versienummer staan in de tekst van het XSD schema.

Aan de naam van het XSD file is het major versienummer toegevoegd.

Als de namespace van een XSD schema is voorzien van een prefix, is aan de prefix het versienummer 1 toegevoegd.

Hieronder een voorbeeld van een OAGIS XSD met een major versienummer in de namespace en een minor en major versienummer in de comment.

*Voorbeeld*

Het XSD schema van het Business Object Product/ ItemMaster wordt opgeleverd met major.minor versie 1.0

De major versie 1 staat in de namespace. de namespace heeft geen prefix.

**Schema settings**

Default Element form: ☒ Qualified ☐ Unqualified  
 Default Attribute form: ☐ Qualified ☒ Unqualified

Block default:   
 Final default:   
 Version:   
 xml:lang:  ID:

☐ No target namespace  
☒ Target namespace:

| Prefix | Namespace                                 |
|--------|-------------------------------------------|
| xsd    | http://www.w3.org/2001/XMLSchema          |
|        | urn:postnl/mail/itemmaster/1              |
| ma1    | urn:postnl/mail/manufacturingcomponents/1 |
| co1    | urn:postnl/mail/components/1              |
| me1    | urn:postnl/mail/meta/1                    |
| fi1    | urn:postnl/mail/fields1                   |

OK Cancel



Het major.minor versienummere 1.0 staat in de comment.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <!-- edited with XMLSpy v2010 rel. 3 (http://www.a
3  <!--
4
5      CS PostNL versie 1.0
6
7
8      Schema agency:   Open Applications Group
9
10
11     OAGIS® Revision:  OAGIS 9_5
12     Date:              05 May 2011
13     Copyright 1997-2011, All Rights Reserved

```

Het XSD Schema van de Component files en de Fields file heeft bij oplevering door het project:

Major versie 1 in de namespace

Major versie 1 in de prefix.

**Schema settings**

Default Element form: ☒ Qualified ☐ Unqualified  
 Default Attribute form: ☐ Qualified ☒ Unqualified

Block default:   
 Final default:   
 Version:   
 xml:lang:  ID:

☐ No target namespace  
☒ Target namespace:

| Prefix | Namespace                               |
|--------|-----------------------------------------|
| xsd    | http://www.w3.org/2001/XMLSchema        |
| co1    | urn:postnl/mail/components/1            |
| ccts   | urn:un:unece:uncefact:documentation:1.1 |
| fi1    | urn:postnl/mail/fields/1                |
| me1    | urn:postnl/mail/meta/1                  |

OK Cancel

## Versienummering bij aanpassing Canonical Scheme

Bij het aanpassen van het Canonical Scheme wordt de strategie van major en minor versienummers gehanteerd, waarbij minor versies de backward en forward comptabiliteit niet aantasten en major versies dit wel doen.

De volgende typen aanpassingen leiden tot aanpassing major en/of minor versie:

### Business Object

Ieder Business Object wordt beschreven in een eigen XSD Schema.

Ieder Business Object heeft een eigen namespace waarin het major versienummer is opgenomen.

In de comment van het XSD schema van het Business Object staat het major en het minor versienummer.

De namespace van een Business Object heeft geen prefix; dus geen versienummer in de prefix.

- nieuw Business Object
  - major versienummer = 1
  - minor versienummer = 0
  
- aanpassing Business Object
  - toevoeging optioneel element: minor versienummer = +1
  - toevoeging verplicht element: major versienummer = +1
  - wijziging constraint major versienummer = +1
  - wijziging beschrijving minor versienummer = +1
  
- aanpassing in component dat door Business Object wordt gebruikt
  - major versiewijziging component: major versienummer = +1
  - minor versiewijziging component: geen aanpassing versienummer
  
- aanpassing in een field dat door een Business Object wordt gebruikt
  - major versiewijziging field: major versienummer = +1
  - minor versiewijziging field: geen versie aanpassing
  
- verwijderen Business Object: is niet toegestaan

## Component

Alle door de Business Objecten Components staan verdeeld over 6 XSD Schema's (cf. OAGIS filestructuur).

Components worden door meerdere Business Objecten gebruikt.

Bij oplevering van het Canonical Scheme wordt vanuit de opgeleverde Business Objecten gebruik gemaakt van een deel van de Components.

Alle Component files staan in een eigen namespace.

In de namespace staat het major versienummer.

In de comment van het XSD schema van iedere Component file staat het major en het minor versienummer.

In de prefix van iedere Component file staat het major versienummer.

Components kunnen op de volgende wijze worden aangepast

- nieuw Component
  - optioneel, nog niet in OAGIS minor versienummer = + 1
  - optioneel, reeds in OAGIS geen aanpassing versie
  - nieuw verplicht component major versienummer = + 1
- aanpassing Component
  - toevoeging optionele elementen minor versienummer = + 1
  - toevoeging verplichte elementen major versienummer = + 1
  - wijziging constraint:
    - component gebruikt door BO major versienummer = +1
    - component nog niet gebruikt: minor versienummer = +1
  - wijziging beschrijving: minor versienummer = +1
  - verwijderen tekstmarkering zodat component weer gebruikt kan worden:
    - optioneel component minor versienummer = +1
    - verplicht component major versienummer = +1
- verwijderen component (het verwijderen van een component is niet fysiek verwijderen maar het markeren van de component als tekst)
  - component gebruikt door BO: major versienummer = +1
  - component niet in gebruik: geen aanpassing versie

## Fields

Alle Fields staan in 1 XSD Scheme

Fields worden door meerdere Components en Business Objecten gebruikt.

Het Field file staat in een eigen namespace.

In de namespace staat het major versienummer.

In de prefix staat het major versienummer.

Dit betekent voor versiebeheer:

- nieuw Field
  - optioneel, nog niet in OAGIS minor versienummer = + 1
  - optioneel, reeds in OAGIS geen aanpassing versie
  - nieuw verplicht field major versienummer = + 1
  
- aanpassing Field
  - toevoeging optionele elementen minor versienummer = + 1
  - toevoeging verplichte elementen major versienummer = + 1
  - wijziging constraint:
    - field in gebruik door BO/Comp major versienummer = +1
    - field nog niet gebruikt: minor versienummer = +1
  - wijziging beschrijving: minor versienummer = +1
  - verwijderen tekstmarkering zodat field weer gebruikt kan worden:
    - optioneel field minor versienummer = +1
    - verplicht field major versienummer = +1
  
- verwijderen field (het verwijderen van een field is niet fysiek verwijderen maar het markeren van de field als tekst)
  - field gebruikt door BO/Comp: major versienummer = +1
  - field niet in gebruik: geen aanpassing versie

## Non semantic data types

De door OAGIS gebruikte versienummers worden overgenomen.

Aanpassingen zullen in principe niet voorkomen, dus geen versiebeheer nodig.

## 7

### Inrichtingskeuzes: Database en Mappenstructuur

Het CSPostNL is vastgelegd in 2 databases.

In CSPostNL-Ontwikkel worden Business Objecten ontwikkeld, aangepast en getest.

Na akkoord worden ze opgeleverd naar CSPostNL-Productie. Vanuit CSPostNL-Productie kunnen Business Objecten door derden (ESR) worden gebruikt.

De file- en mappenstructuur van het Canonical Scheme PostNL (zowel Ontwikkel als Productie) is als volgt: CSPostNL bestaat uit 2 hoofdmappen: Common en Nouns.

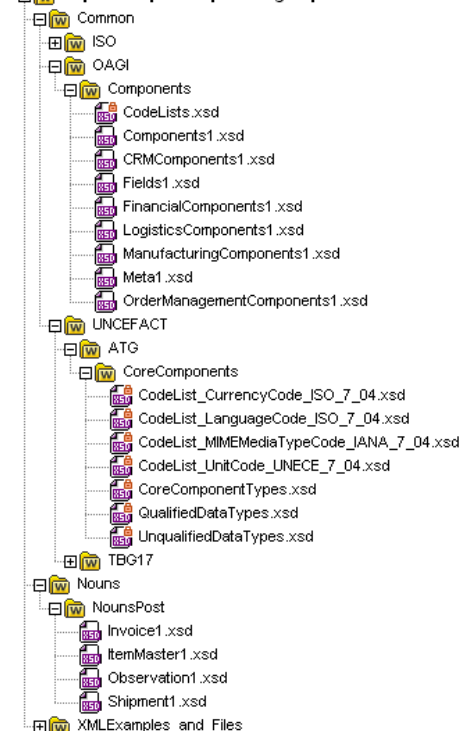
In de map Common zijn de herbruikbare elementen vastgelegd volgens de map en file indeling van OAGIS:

- In de map OAGIS/ Components staan de XSD files voor Componenten en Fields
- In de map UNCEFACT/CoreComponents staan de No business semantic XSD files van de Core Data Types

In de map Nouns staan de BusinessObjecten:

- In de map Nouns/ NounsPost staan de XSD files van de Business Objecten van het CSPostNL

<http://sharepoint.nl.post.tnt.groupit/bit/architectuur/CSP/CSPostNL>



Door major wijzigingen kunnen verschillende versies van hetzelfde XSD actueel zijn.

Alle versies worden in dezelfde map bewaard, het XSD met het hoogste versienummer is de meest actuele.

## 8

### Inrichtingskeuzes: Koppeling CS met ARIS en ESR.

#### Koppeling CS PostNL en ARIS

Nb. Op 26-08-2011 is door de beheerders van het handboek conventies Aris besloten dat:

- in Aris een repository wordt opgenomen van Business Objecten. Dit zijn Business Objecten volgens de definitie van Archimate; dus de informatie sets die voorkomen in het business domein
- als een Business Object voorkomt in het applicatie domein wordt dit weergegeven via een Data Object
- de structuur van een Data Object wordt gemodelleerd door middel van Entiteiten en de relaties ertussen
- de inhoud van een Entiteit wordt gemodelleerd door middel van Attributen

De koppeling tussen Aris en het CS ligt dan als volgt:

| ARIS        | Canonical Scheme |
|-------------|------------------|
| Data Object | Business Object  |
| Entiteit    | Component        |
| Attribuut   | Field            |

Door middel van naamgeving en een external link vanuit het Aris object naar de XSD file in het Canonical Scheme wordt de link geïmplementeerd.

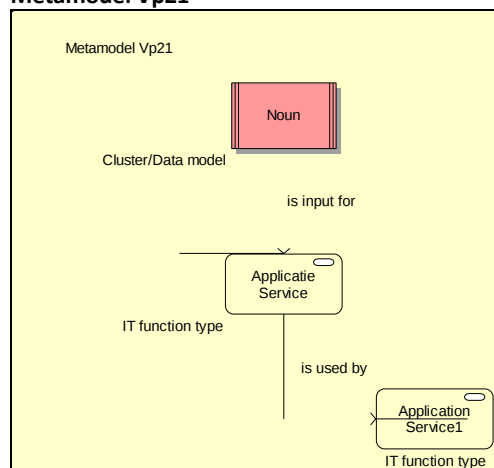
In de huidige situatie is de koppeling als volgt geregeld:

#### Koppeling Services – Noun

In Aris worden in het VP21 Service Compositie Viewpoint services gegroepeerd in serviceaggregaties en noun definities, beide ten behoeve van het canonical model.

*Nb. In de nieuw op te leveren versie van het handboek conventies zal Noun gewijzigd zijn in Data Object.*

#### Metamodel Vp21



De link tussen Aris en het Canonical Scheme wordt als volgt gelegd:

- naast een Nederlandse naam krijgt het Noun ook een Engelse naam
- deze Engelse naam moet gelijk zijn aan de naam van het Business Object in het Canonical Scheme
- in het veld link van het object Noun wordt een rechtstreekse link gelegd naar de XSD file in het Canonical Scheme

### Koppeling Business Object – Gegevensmodel

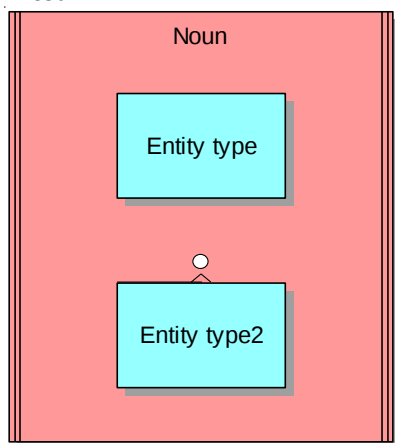
Nb. In de nieuw op te leveren versie van Aris zullen nieuwe viewpoints beschikbaar komen voor het vastleggen van de structuur en inhoud van Data Objecten. Voorlopig worden onderstaande viewpoints gebruikt.

Aris biedt voor gegevensmodellering oa. de viewpoints:

VP09a-Base Registration Structure en VP24 Entity with attributes.

Op het modeltype VP09a-Base Registration Structure wordt de structuur van het Noun door middel van entiteiten zichtbaar gemaakt. Entiteiten in Aris zijn vergelijkbaar met Componenten in het Canonical Scheme.

#### VP09a



Via het attribuut Link van het Aris object Noun of Entiteit kan rechtstreeks een link worden gelegd met het CSPostNL op Sharepoint.

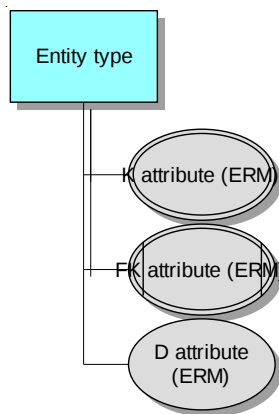
Het modeltype VP24 wordt (voorlopig) gebruikt voor het functioneel ontwerp van de inhoud van de Componenten.

Per entiteit worden de attributen waaruit de entiteit is samengesteld gemodelleerd op het modeltype VP24-Entity with attributes.

Een entiteit in Aris is een Component in het CSPostNL; een attribuut in Aris is een Field in het CSPostNL.

Dit modeltype wordt alleen gebruikt ten behoeve van de analyse van de inhoud van Componenten die nog niet het CSPostNL voorkomen.

#### VP24



Deze modellen in Aris vormen het logisch/ functioneel ontwerp van het Business Object en zijn de basis waarop de definitie van het canonieke Business Object wordt gebaseerd.

#### Koppeling CS PostNL en ESR SAP PI

Helaas is het (nog) mogelijk een extern Canonical Scheme binnen de ESR van SAP PI automatisch te importeren.

Business Objecten, Components en Fields gedefinieerd in het Canonical Scheme worden handmatig overgezet naar Business Objects, Business Objects Nodes en Global Data Types in de ESR van SAP PI. Daarbij wordt in de ESR zoveel mogelijk de namespaces van het CSPostNL overgenomen.

Richtlijnen voor de inrichting van de ESR zijn dan ook niet van toepassing.

(zie Gebruik CS in ESR V03, Rolf van der Tang/ Heleen de Boer, 28 maart 2011)

## 9

### OAGIS: Gebruik OAGIS Business Object binnen Canonical Scheme PostNL

Dit hoofdstuk geeft de stappen aan de die nodig zijn om een Noun uit OAGIS over te zetten naar CPostNL.

#### Stap 1: Back Up CS PostNL

Maak op stick een back up van het CPostNL:

- open via verkenner CPostNL
- selecteer hoogste niveau CPostNL
- maak kopie van CPostNL naar stick

Group IT Net > Business IT > Architectuur > Canonical Schema PostNL

### Canonical Schema PostNL

Canonical Schema PostNL

Acties ▾ Weergave: Verkenner-weergave ▾

**Bestands- en maptaken** ⌵

- Deze map een andere naam geven
- Deze map verplaatsen
- Deze map kopiëren
- Deze map op het web publiceren
- De bestanden in deze map per e-mail verzenden
- Deze map verwijderen

**CPostNL**

**OAGIS\_9\_5**

Methode van mappen van XML Spy op Sharepoint.doc  
Microsoft Word-document

Procedure maken Backup CPostNL.doc  
Microsoft Word-document

Status BO en overzicht gebruik Components.xls  
Microsoft Excel-werkblad

- op stick:
- rename CPostNL opstick naar: <datum>CPostNLBackup
- via winzip: add to <datum>CPostNLBackup.ZIP
- vervolgens op sharepoint:
- via Uploaden zip file plaatsen in Backup Map CPostNL (voorlopig nog in projectmap)

Group IT Net > Business IT > Architectuur > Canonical Schema > Projectinhoudelijke Documenten > 99 Backup CPostNL

### Projectinhoudelijke Documenten

Nieuw ▾ | Uploaden ▾ | Acties ▾ Weergave: Alle documenten ▾

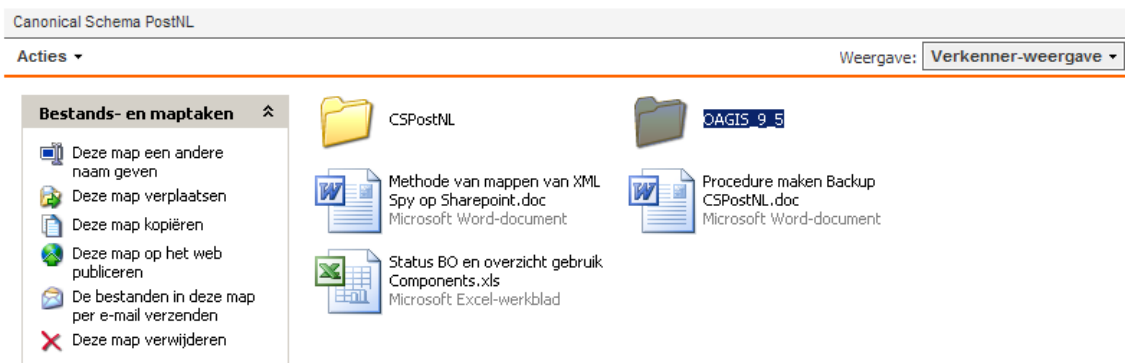
| Type | Naam                    | Gewijzigd       | Gewijzigd door                      |
|------|-------------------------|-----------------|-------------------------------------|
| 📁    | 20110620 CPostNLBackup  | 28-6-2011 8:43  | Boer - van de Velde, Heleen (H.) de |
| 📁    | 20110627 CPostNLBackup  | 28-6-2011 8:44  | Boer - van de Velde, Heleen (H.) de |
| 📁    | 20110704 CPostNL Backup | 4-7-2011 16:07  | Boer - van de Velde, Heleen (H.) de |
| 📁    | 20110815 CPostNLBackup  | 15-8-2011 10:39 | Boer - van de Velde, Heleen (H.) de |
| 📁    | 20110829 CPostNL Backup | 29-8-2011 9:17  | Boer - van de Velde, Heleen (H.) de |
| 📁    | 20110912CPostNLBackup   | 12-9-2011 12:15 | Boer - van de Velde, Heleen (H.) de |

## Stap2: Toevoegen OAGIS Noun aan Back Up CS PostNL

- open OAGIS op Sharepoint

Group IT Net > Business IT > Architectuur > Canonical Schema PostNL

### Canonical Schema PostNL



- aan CSPostNL toe te voegen Noun openen met verkenner
- kopiëren Noun naar back up op stick, in subdirectory Nouns
- maak nieuw Altova project met link naar backup op stick

## Stap 3: Aanpassen OAGIS NOUN

### 1) namespace aanpassen

- open Noun,
- in text aanpassen target namespace naar: urn:postnl/mail/<naamBusinessObject>/1

### 2) includes vervangen

- import toevoegen via browse + make path relative
- verwijder includes
- overige imports toevoegen voor components, meta en fields + make path relative

### 3) prefix aanpassen

- wegwerken alle rode gemarkeerde prefixen; deze verwijzen naar onjuiste prefix
- de meeste prefixen verwijzen naar de Fields file. Handig kan zijn een replace van ref=" naar ref="fi1
- de overige prefixen on line aanpassen
- **let op:** de prefix op het hoogste niveau is niet in het schema te zien: deze aanpassen in text

### 4) filenaam

- remane het Noun naar <naam>/1

#### Stap 4: Aangepast OAGIS Noun van Back Up naar Sharepoint CS PostNL

- maak kopie Noun op stick
- open Sharepoint CSPostNL in verkenner
- open subdirectory Nouns
- plakken Noun

#### Stap 5: Check

- open CSPostNL
- check nieuw Business Object door deze te openen en te save = ok

#### Stap 6: Bijwerken documentatie

- bijwerken excel met status Business Object

## 10 Gerelateerde documenten

Verdere uitwerking van de richtlijnen staat in:

- Procedure Beheren Business Objecten
- Procedure Consistentie ARIS – CS - ESR