

## Introduceren van een koppelingenregister

Bert Dingemans

**Veel organisaties introduceren verschillende vormen van systeem integratie zoals webservices en database-koppelingen. Hiermee wordt hergebruik van gegevens en applicatiefunctieiteit mogelijk. Echter op enig moment zal het aantal en de diversiteit van deze koppelingen om governance vragen. Het introduceren van een service- of koppelingenregister is veelal de gekozen oplossingsrichting. Het project dat de afgelopen periode binnen het ministerie van LNV is uitgevoerd geldt hierbij als uitgangspunt. Dit artikel beschrijft hoe op basis van interactieve workshops de applicatie-architectuur van dit register gemodelleerd is. De inbedding van het register in de werkorganisatie is daarbij door deze workshops met direct betrokkenen vereenvoudigd. Als laatste wordt ingegaan op de do's en don'ts bij het introduceren van dit type governance tools.**

## Inleiding

Het ontsluiten van applicaties op basis van koppelvlakken is een werkwijze die reeds lang wordt toegepast bij systeemintegratie. Deze ontsluitingswijze heeft zich ontwikkeld van batchgewijze bestandsuitwisseling naar realtime berichtuitwisseling. Door deze ontwikkeling is de afhankelijkheid van applicaties voor gegevens vanuit andere toepassingen steeds groter geworden, met name omdat de actualiteit van de uitgewisselde gegevens gewenst is.

Bij (grote) organisaties ziet men vanwege deze behoefte aan actuele gegevens een groot aantal koppelingen ontstaan tussen de gebruikte applicaties. Daar komt bij dat voor de afzonderlijke applicaties veelal verschillende technische oplossingen gezocht worden. Denk hierbij aan koppelingen tussen databases (databaselinks), bestandsuitwisseling, ETL inrichtingen en webservices.

Ten behoeve van de beheersbaarheid en het stimuleren van hergebruik van koppelingen [Erl] is het gebruik van een register van deze koppelingen een hulpmiddel. Dit register kan bijdragen aan een applicatielandschap dat op gestructureerde wijze de gegevensuitwisseling ingericht heeft en dat mogelijkheden biedt om dit applicatielandschap beheersbaar, koppelvlakken te hergebruiken en optimaal gebruik te maken van de reeds aanwezige koppelvlakken.

Dit artikel beschrijft een praktijkcase toegepast bij het Ministerie van LNV (Dictu) om te komen tot een koppelingenregister. Op basis van interactieve ontwerp sessies is een prototype ontwikkeld van een koppelingenregister. Door deze ontwerp sessies is een duidelijk beeld ontstaan van de requirements van het register, de gebruikersparticipatie verhoogd en is het werkproces beschreven dat de basis vormt voor de toekomstige inrichting van een koppelingenregister.

## Webservices en andere koppelingen

Bij koppelingen tussen applicaties worden momenteel veelal webservices bedoeld, men spreekt veelal over een serviceregister. Toch zijn webservices zeker niet de enige koppelwijze toegepast tussen applicaties, vandaar dat hier gesproken wordt over een koppelingenregister. In onderstaande opsomming vindt u een aantal voorbeelden van integratiewijzen:

- Bestandsuitwisseling
- FTP en SFTP
- Databasekoppelingen en databaselinks
- ETL scripts
- Webservices (SOAP en REST)
- EDIFACT etc

Alle soorten van koppelingen kenmerken zich door een bepaalde structurering van de gegevens. Deze structurering kan eenvoudig zijn, denk hierbij bijvoorbeeld aan bestandsuitwisseling op basis van SDF en comma delimited formaten. Echter het andere uiterste zijn berichten op basis van XML waarbij de inhoud en de metadata op vergaande wijze met elkaar verweven zijn.

Door deze structurering en de daadwerkelijke inhoud van de gegevens kenmerken koppelingen zich ook door de mogelijkheid dat de gegevens die via een koppeling uitgewisseld worden niet noodzakelijkerwijs overeenkomen met de interne structuur van de gekoppelde applicaties. Veelal worden in de koppelvlakken transformaties en selecties uitgevoerd op de gegevens die uitgewisseld worden. Deze ontkoppeling van de interne applicatiestructuur en de uitwisselstructuur introduceert de mogelijkheid om koppelvlakken te gaan hergebruiken.

Door deze ontkoppeling kan applicatieonafhankelijkheid ontstaan. Zodra de koppelvlakken gestandaardiseerde gegevensuitwisseling realiseren en er een ontkoppelpunt geïntroduceerd

wordt is het voor afnemende applicaties niet noodzakelijk kennis te hebben over de aanleverende applicaties. Enkel kennis over de berichtindeling is voldoende. Met name de inzet van webservices en producten als servicebussen maken de realisatie van deze ontkoppelde systeemintegratie mogelijk.

Echter door deze werkwijze zal het aantal en de diversiteit van de koppelingen en koppelvlakken sterk toenemen. Gevolg is dat wil men hergebruik van koppelvlakken stimuleren er een voorziening aanwezig moet zijn om op eenvoudige wijze in beeld te krijgen welke koppelvlakken bruikbaar zijn voor hergebruik [Erl].

## Implementatietraject

Bij het zoeken naar een oplossing voor bovengenoemd probleem zijn meerdere oplossingsrichtingen mogelijk. Hieronder zijn er een aantal uitgewerkt die binnen het ministerie van LNV in de afgelopen periode zijn onderzocht.

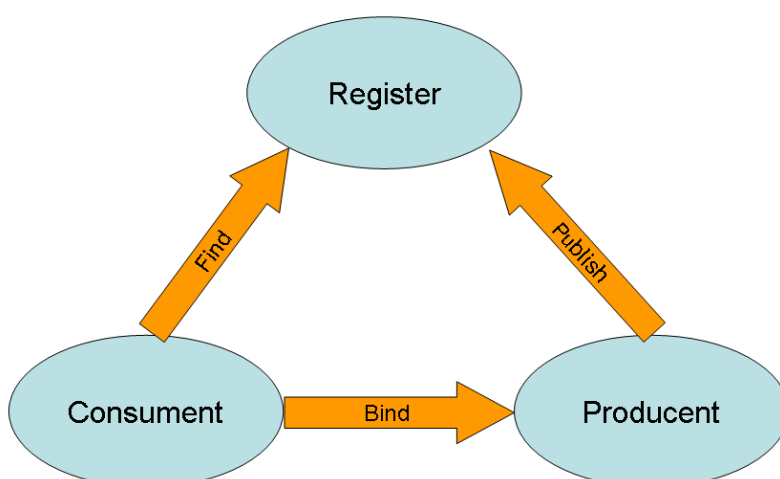
### **Onderzoek standaard producten**

Niet als koppelingenregister maar als service register of service registry zijn een aantal standaard producten op de markt beschikbaar, ook als Open Source Software. Bijvoorbeeld:

- <http://www.oracle.com/us/technologies/soa/soa-governance/index.html>
- <http://www.membrane-soa.org/soa-registry/>
- <http://juddi.apache.org/>

Veel van deze producten zijn gebaseerd op het publish – find – bind principe. Dit principe werkt als volgt:

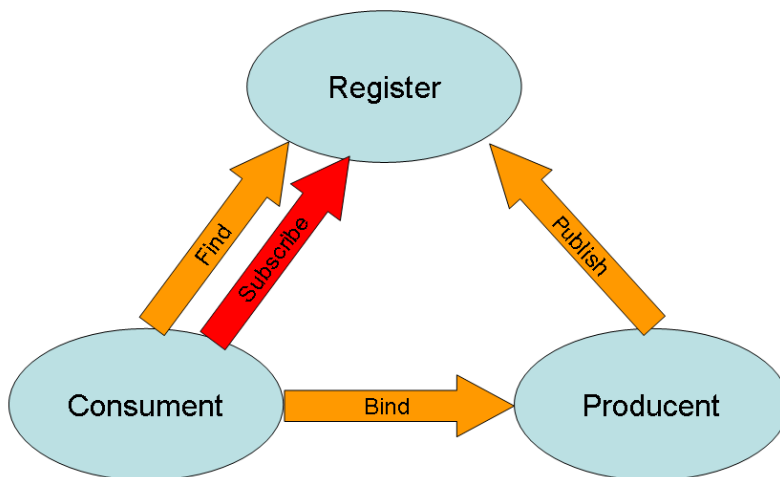
- Een koppeling producent (bronsysteem) maakt bekend dat er een koppeling is en wat de structuur van de koppeling is in het register (publish).
- Wil een consument (doelsysteem) gebruik maken van gegevens dan wordt in het register gekeken of er reeds een producent is (find).
- Is er een koppeling is die gebruikt kan worden dan wordt vervolgens een koppeling gelegd tussen consument en producent en kan uitwisseling van gegevens plaats gaan vinden (bind)



Afbeelding 1: Publish – Find – Bind principe

Nadeel van deze principe opzet is dat er niet geregistreerd wordt wat de consumenten zijn van een producerend koppelvlak. Dit heeft mogelijk gevolgen voor de performance en de

beschikbaarheid van de producent. Belangrijker nog is dat niet bekend is of er consumenten zijn die gebruik maken van dit koppelvlak. Gevolg kan zijn dat deze producent beheerd wordt zonder dat er gebruik van gemaakt wordt. Bijvoorbeeld omdat er reeds een nieuw versie is van het producent koppelvlak dat beter aansluit bij de behoeften van de consumenten. Daarom kan beter gekozen worden voor het Publish – Find – Subscribe – Bind principe. Hierbij wordt een extra stap toegevoegd waarbij een consument voordat er een bind met de producent aan het register kenbaar maakt dat er een koppeling gelegd wordt (subscribe).



Afbeelding 2: Publish – Find – Subscribe – Bind principe

### ***Andere initiatieven***

Met name in grote organisaties zijn meestal al initiatieven op andere vlakken van configuratie beheer ontplooid. Denk hierbij bijvoorbeeld aan applicatie registers, configuration management databases (CMDB) maar ook aan tooling ten behoeve van ITIL implementaties. Deze tools bieden vaak ook de mogelijkheid om koppelingen en services te registreren en te beheren. Soms bieden bovengenoemde producten ook de mogelijkheid om enerzijds de onderliggende infrastructuur van deze koppelingen in kaart te brengen. Hierdoor is het monitoren maar ook het inrichtingsbeheer te vereenvoudigen. Anderzijds zijn er producten die het mogelijk maken op geautomatiseerde wijze op zoek te gaan naar koppelingen en services. Dit biedt als voordeel dat het beheer van de gegevens sterk vereenvoudigd kan worden. Daarnaast kan bewaakt worden dat er geen koppelingen gelegd worden die niet onder de architectuur vallen.

### ***Maatwerk***

Het is mogelijk om een maatwerk toepassing te ontwikkelen waarin de koppelingen en de bijbehorende entiteiten beheerd kunnen worden. Voordeel van deze opzet is dat het nauw aansluit bij de behoefte van de beheerorganisatie. Echter een waarschuwing is hier noodzakelijk. Houdt er rekening mee dat hiermee het beheer van de inhoud een omvangrijke taak kan zijn. Veelal zijn er een groot aantal koppelvlakken en koppelingen aanwezig en is het ondoenlijk om deze administratie sluitend te houden.

Een geautomatiseerde implementatie van het publish – find – subscribe – bind principe kan hierbij een bijdrage leveren. Maar ook het gebruik van andere initiatieven zoals hierboven genoemd kunnen een bijdrage leveren.

## ***Combinatie van bovenstaande oplossingsrichtingen***

In een aantal gevallen kan een combinatie van bovengenoemde oplossingsrichtingen uitkomst bieden. Denk hierbij aan het uitbreiden van maatwerk oplossingen voor andere deelgebieden (zoals hierboven beschreven) die uitgebreid worden met een klein deel maatwerk voor bijvoorbeeld de koppelingen. Maar ook het aansluiten op generieke voorzieningen als Document Management Systemen, Wiki's en samenwerkingsplatformen kunnen gecombineerd worden met standaardpakketten en of maatwerk oplossingen. Binnen het ministerie van LNV is gekozen voor deze oplossing waarbij koppelingen met een applicatieregister en een CMDB voorzien zijn.

## **Werkwijze prototyping**

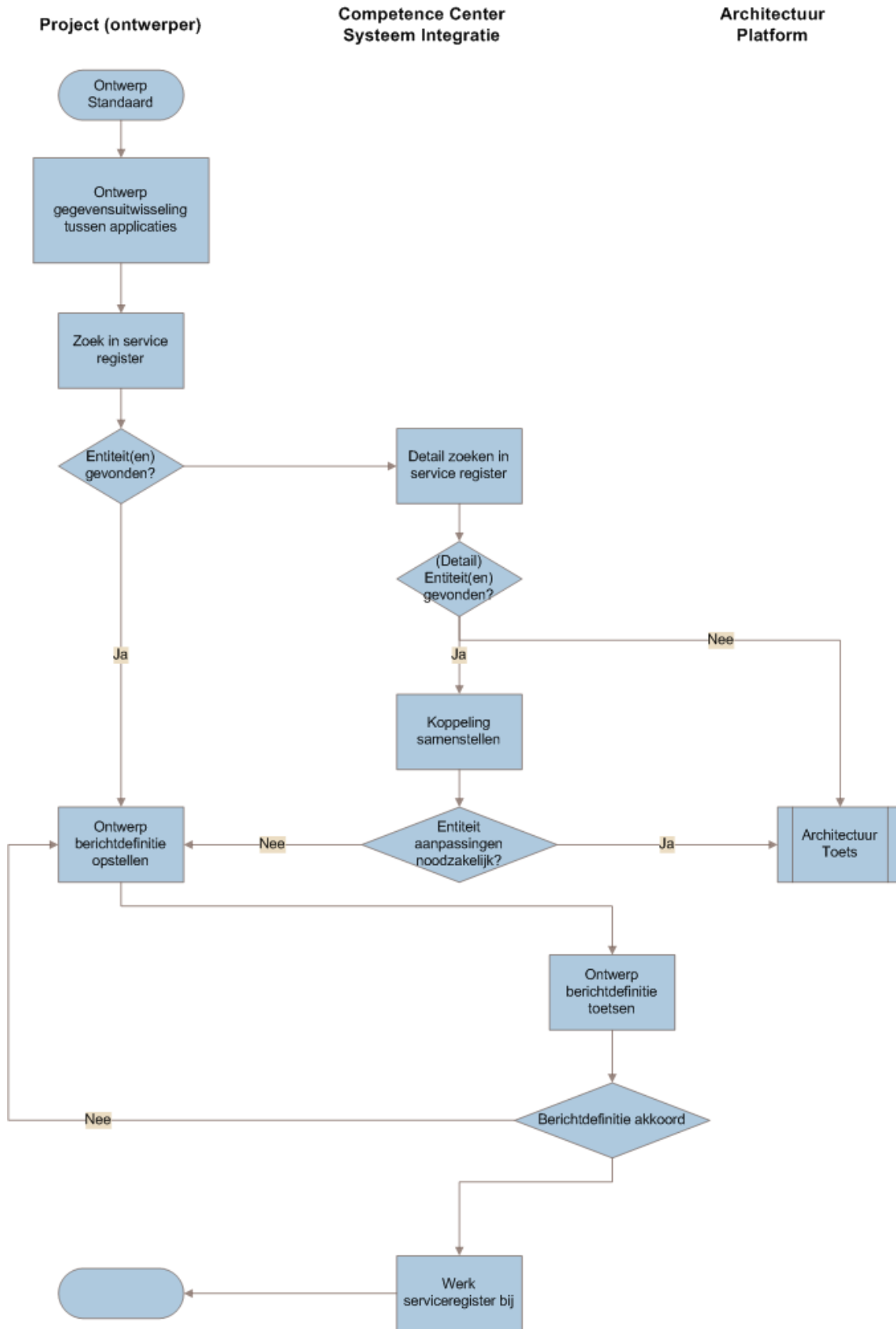
Bij het project binnen het ministerie van LNV ben ik betrokken geweest bij een project waarbij de hierboven beschreven problematiek speelde. Na de inventarisatie van bestaande producten en andere alternatieven ontstond de behoefte om nader te onderzoeken wat de behoeften van de organisatie waren en op basis hiervan uit de oplossingsrichtingen een uitwerking te kiezen. Hiertoe zijn onderstaande activiteiten uitgevoerd, deze worden vervolgens hieronder uitgewerkt:

- Opstellen bedrijfsproces
- Inventarisatie domeinmodel
- Interacties en services
- Prototypes evalueren

## ***Opstellen bedrijfsproces***

Als eerste stap is onderzocht hoe het bedrijfsproces bij het inzetten van een koppelingenregister eruit gaat zien. Op basis van de rollen in de organisatie en het bestaande ontwerpproces is een aanpassing gedaan in de bestaande procesinrichting. Hierbij is een getrapte opzet gemaakt om ervoor zorg te dragen dat de beheerders van het koppelingenregister een ondersteunende taak hebben bij het gebruik van de functionaliteit door gebruikers (ontwerpers e.d.)

In onderstaande afbeelding ziet u een deel van het bedrijfsproces. In dit model is de verdeling van de activiteiten op basis van swimming lanes voor de verschillende rollen uitgewerkt.



### Standaard ontwerptraject koppeling

Afbeelding 3: Bedrijfsproces

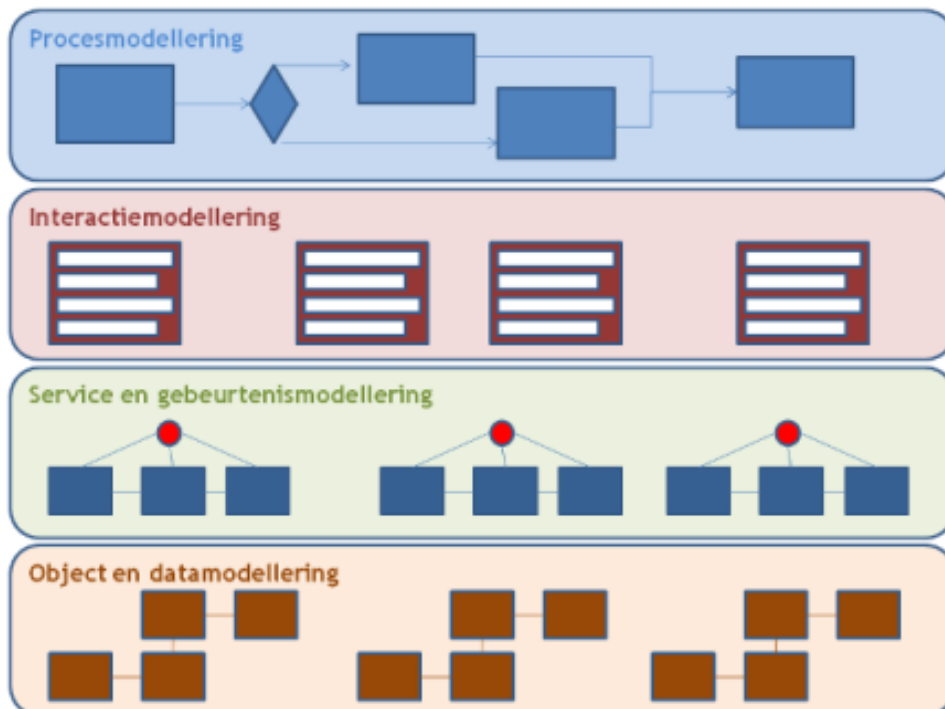
## ***Inventarisatie domeinmodel***

In deze tweede fase is gezocht naar de domeinentiteiten en hoe deze aan elkaar gerelateerd zijn. Hierbij is in eerste instantie een summier domeinmodel opgesteld. Binnen het domeinmodel kunnen entiteiten, associaties tussen entiteiten en eigenschappen van entiteiten gemodelleerd worden. De eerste uitwerking is hierbij summier van opzet, gedurende de interactieve sessies worden de entiteiten steeds verder ingevuld.

## ***Interacties en services***

Op basis van het domeinmodel wordt een voorstel voor interacties en basic services geïmplementeerd. Deze interacties en services zorgen voor de koppeling tussen het bedrijfsproces en het domeinmodel. De interacties worden hierbij gevoed met gegevens door gegevensverstreckende services eventueel mutaties in de interacties worden op basis van gegevensbewerkende services doorgevoerd in het domeinmodel [Snoeck].

In onderstaande afbeelding is te zien hoe de interacties en services gerelateerd zijn aan het domeinmodel en het bedrijfsproces.



Afbeelding 4: Architectuurmodel prototype omgeving

Hiermee zijn alle elementen in het architectuurmodel voor een applicatiesimulatie ingevuld en kan het model gesimuleerd worden.

## ***Prototypes evalueren***

Op basis van interactieve ontwerpsessies is het bedrijfsmodel in een aantal stappen ingevuld. Door het huidige prototype steeds als uitgangspunt voor de discussie te nemen en tijdens de discussie direct aanpassingen door te voeren in het model kunnen wijzigingen direct geëvalueerd en eventueel aangepast worden. Het ontwikkelen en evalueren van de prototypes wordt in een viertal stappen uitgevoerd, te weten:

- Interacteer, bediscussieer het architectuurmodel met de betrokken deskundigen
- Modelleer, stel een model op of pas het bestaande model aan
- Registreer, registreer afspraken gemaakt over het model, dit ter voorkoming van ping-pong modelleren.

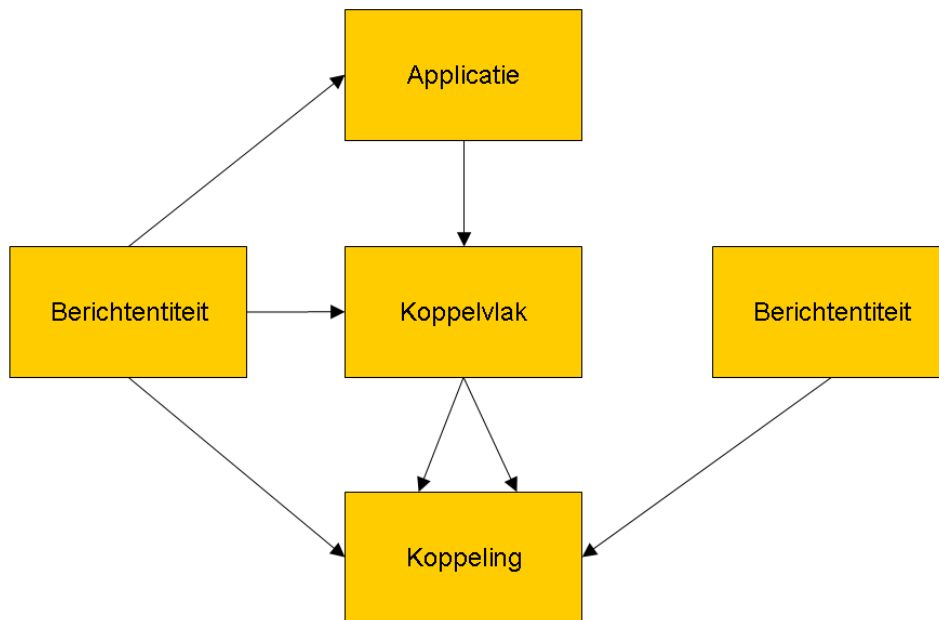
- Evalueer, evalueer de aanpassingen met de betrokken deskundigen

Deze vier stappen worden in een aantal iteraties herhaald. Vanzelfsprekend wordt dit voor grote domeinmodellen op basis van modules uitgevoerd. Uitkomst van de werkwijze is een applicatiemodel en een werkend prototype van het koppelingenregister. Het uiteindelijke model van dit koppelingenregister wordt in het volgende hoofdstuk beschreven

## Koppelingen of serviceregister

### **Register Entiteiten**

In voorgaande hoofdstukken zijn reeds een aantal requirements aan de orde gekomen. Ten eerst het bedrijfsproces dat ondersteund dient te worden door deze toepassing. Daarnaast is de wens dat niet alleen webservices geregistreerd moeten worden een bepalend criterium. Meest bepalend is echter de wens om niet alleen de koppelvlakken maar ook de daadwerkelijke koppelingen te registreren in het systeem. In onderstaande afbeelding wordt een vereenvoudigd entiteit-relatiemodel gegeven van de register entiteiten



Afbeelding 5: entiteit-relatiemodel

Onderstaande paragrafen geven een korte toelichting over de desbetreffende entiteit

### Applicatie

Applicaties omvatten de informatiesystemen waarin gegevens verwerkt worden ten behoeve van de bedrijfsprocessen in de organisatie. Hierbij wordt veelal gedacht aan systemen die gegevens in bestanden manipuleren. Echter denk hierbij ook aan registers die op basis van koppelingen ontsloten worden maar waarvan het beheer buiten de scope van de eigen organisatie ligt. Ook belang is dat in deze een service bus of een broker ook geïnterpreteerd wordt als een applicatie binnen het koppelingenregister.

### Koppelvlak

Omvatten een, veelal technische, implementatie, van de transformatie van de interne indeling van de gegevens in de betrokken applicatie naar een generieke structuur waarmee de



gegevens interpreteerbaar zijn voor andere toepassingen. Belangrijke extra uitwerking is dat in het register de historie en releases van koppelvlakken beheerd wordt.

## Koppeling

Dit omvat de daadwerkelijke verbinding tussen twee koppelvlakken. Dit omvat de registratie van de technische implementatie op basis van de daadwerkelijke integratiewijze. Daarnaast wordt de mogelijkheid geboden om gegevens over omvang en snelheid van de gegevensverwerking in de koppeling te registreren

## Bericht entiteit

Binnen de koppelingen en koppelvlakken wordt gewerkt met gegevensdefinities. Deze definities hebben allemaal een eigen structuur en kenmerken. Om hergebruik van deze structuren (en de transformerende componenten) mogelijk te maken is het wenselijk deze entiteiten vast te leggen. In het volgende hoofdstuk wordt nader ingegaan op deze bericht-entiteiten.

## Document

Bij een organisatie waar het ontwikkelproces wordt ondersteund door een documentenstroom, bijvoorbeeld PSA's, functionele- en technische ontwerpen en testrapportages, zijn deze documenten een belangrijke bron van informatie. Binnen een ontwikkel- of implementatietraject kan een koppeling tussen deze documenten en de entiteiten in het register bijdragen aan een eenvoudige interpretatie van de bestaande koppelingen, wat het hergebruik vereenvoudigt.

## ***Generieke (bericht) entiteiten***

Het werken met een koppelingenregister ondersteunt het hergebruik van berichten. Om dit hergebruik optimaal mogelijk te maken is het opnemen van de berichten onvoldoende. Een bericht kan namelijk uit een groot aantal afzonderlijke elementen bestaan die ook geschikt zijn voor hergebruik. Een administratieve ondersteuning hiervan in het koppelingen register is hiervoor noodzakelijk.

Onderkennen van generieke berichten en entiteiten kan op een aantal niveaus plaatsvinden:

- Gelaagde berichtdefinities
- Generieke entiteiten
- Berichtstandaardisatie voor generieke en specifieke berichtentiteiten

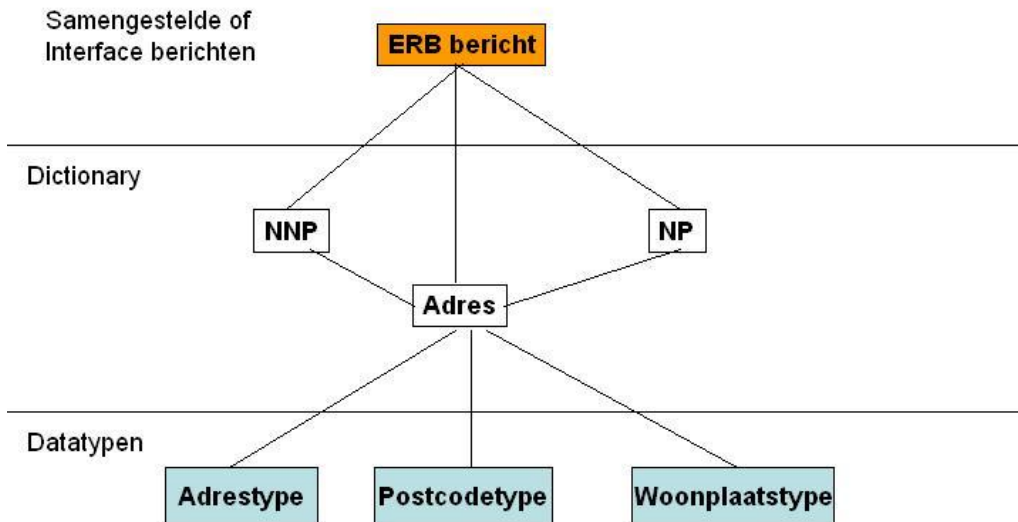
In onderstaande paragrafen wordt nader ingegaan op deze niveaus

### Gelaagde berichtdefinities

Bij gelaagde berichtdefinities beschrijft de onderste laag de gegevenstype of datatype en beschrijft de kenmerken van een enkele eigenschap van één of meerdere entiteiten. Bijvoorbeeld binnen de entiteit adres wordt een postcodetype geïntroduceerd waarbij de eis wordt gesteld 6 posities waarbij de eerste vier een cijfer en de laatste twee een hoofdletter. De middelste laag omvat een dictionary van entiteiten waarbij een entiteit een verzameling is van een aantal eigenschappen (gebaseerd op datatypen). Denk bijvoorbeeld aan de entiteit Adres of Persoon.

De bovenste laag omvat de berichtinterface. Een bericht bestaat uit de clustering van één of meerdere entiteiten vanuit de entiteit dictionary. Daarnaast kan een deel van het bericht bestaan uit specifieke entiteiten (niet afkomstig vanuit de dictionary) waarmee een bericht gegevens kan bevatten voor één specifieke koppeling.

# Gelaagde berichtdefinitie



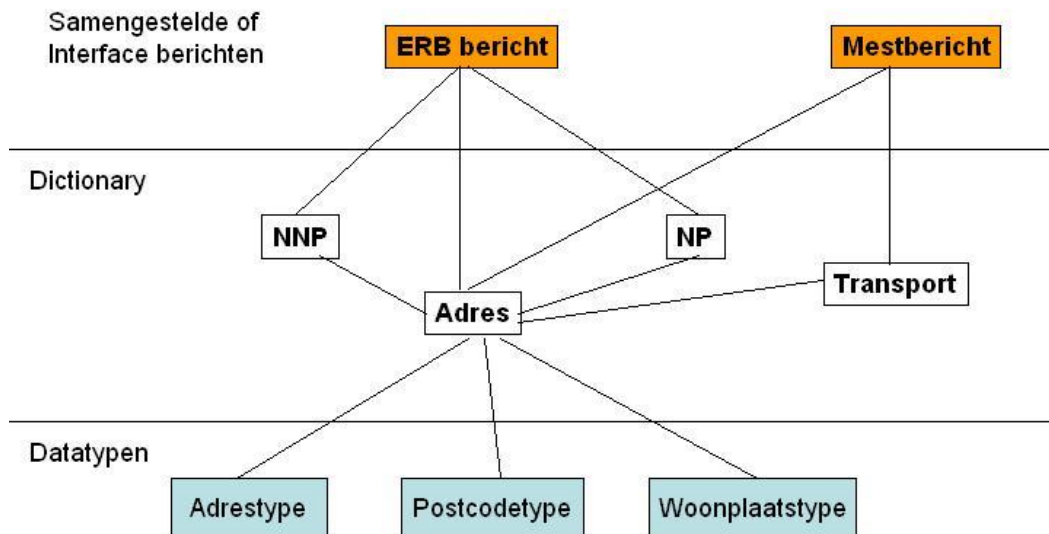
Afbeelding 6: Gelaagde berichtdefinities

## Generieke entiteiten

Door te werken met een dictionary van berichtentiteiten en een interface van berichtdefinities wordt het realiseren van hergebruik eenvoudiger. Omdat entiteiten in de dictionary onderkend worden is bij het definiëren van een nieuw interfacebericht het onderkennen van entiteiten die overeenkomen met de entiteiten in de dictionary eenvoudig.

Bijvoorbeeld in afbeelding 6 is aangegeven dat een relatiebericht (interfacebericht) bestaat uit drie entiteiten in de dictionary. Één hiervan is de entiteit adres. Vervolgens wordt er een nieuw bericht gedefinieerd bijvoorbeeld voor het doen van een mesttransport (zie afbeelding 7). Een mesttransport bericht bestaat enerzijds uit gegevens over het transport (een nieuwe dictionary entiteit. Anderzijds uit een tweetal adressen (het vertrekpunt en het aankomstpunt. Hiervoor kan op dat moment de reeds bestaande entiteit adres worden gebruikt waarmee hergebruik wordt geïntroduceerd.

# Toepassen Dictionary



Afbeelding 7: Uitbreiding en hergebruik generieke entiteiten

Door het werken met interactie berichten en een dictionary worden er andere eisen gesteld aan het koppelingenregister. Binnen dit register zullen ook de dictionary entiteiten en de datatypes een plaats dienen te krijgen.

## Generieke en specifieke gegevens

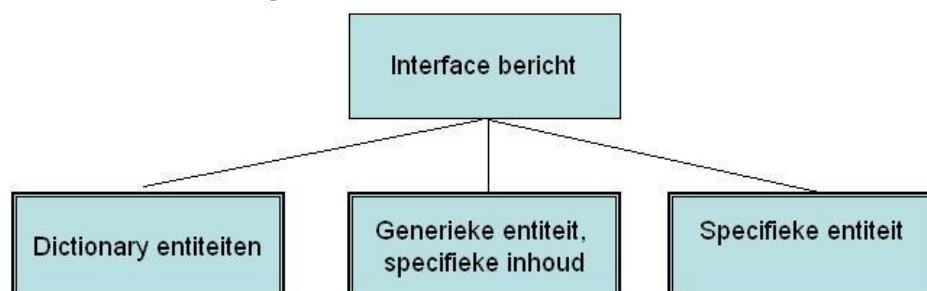
In de vorige paragraaf is aan de orde gekomen hoe generieke entiteiten vormgegeven kunnen worden door het toepassen van een dictionary. Echter in een aantal gevallen zullen generieke berichten niet volstaan. Bijvoorbeeld in de situatie waarbij een bericht gedefinieerd wordt dat zeer specifiek is voor een koppeling tussen twee applicaties.

Hierbij is het wenselijk dat er een mogelijkheid is om specifieke gegevens op te nemen in het interfacebericht, eventueel in combinatie met generieke delen. Een bericht zal daarom bestaan uit maximaal drie hoofdelementen:

- Een samenvoeging van alle generieke entiteiten (vanuit de dictionary)
- Specifieke gegevens echter omvat door een generieke entiteit.
- Specifieke gegevens als specifieke entiteit(en).

In afbeelding 8 wordt dit uitgewerkt, enerzijds in een diagram anderzijds in een eenvoudige XML structuur.

# Voorbeeld generieke oplossing specifieke berichten



```
<interface-bericht>  
  <dictionary-entiteiten>  
  </dictionary-entiteiten>  
  <specifieke-data-entiteiten>  
  </specifieke-data-entiteiten>  
  <specifieke-entiteiten>  
  </specifieke-entiteiten>  
</interface-bericht>
```

Afbeelding 8: Generieke en specifieke scheiding

Het bericht bestaat uit maximaal drie elementen waarbij ten minste één van deze entiteiten ingevuld dient te zijn met een nadere specificatie van de onderliggende elementen.

Het middelste element de generieke entiteit met de specifieke data kan gezien worden als een parametertabel met daarin een lijst van namen en waarden behorend bij deze waarde. Hiermee kan een eenvoudige lijst van specifieke waarden ondergebracht worden in een interfacebericht. Eventueel kan deze naam/waarde lijst uitgebreid worden met XML datatypen en reguliere expressies om controle op de ingevulde data mogelijk te maken.

## Lessons learned

Bij de introductie van het koppelingenregister in een grote organisatie komen veel aspecten van de ICT organisatie in scope van de activiteiten. Gevolg is dat de ene activiteit meer inzet vraagt om tot een positief einde te komen dan de andere. In onderstaande opsomming een aantal aspecten waar wij in het LNV project tegenaan zijn gelopen en de ondernomen acties.

- Bepaal vooraf de scope van het register. In de loop van het project zijn overal kansen om extra gegevens op te slaan over koppelingen en koppelvlakken. In dit project wilde we bijvoorbeeld in een later stadium gegevens over leveringsovereenkomsten en eigenaarschap gaan opnemen. Gezien de omvang van het team dat het register gaat beheren is hier in een vroeg stadium nee tegen gezegd.
- Zoek naar bronsystemen voor ondersteunende gegevens. In onze situatie bleek er een geautomatiseerd applicatieregister en een CMDB toepassing aanwezig. Deze registers kunnen gegevens leveren die voor het koppelingenregister ondersteunend zijn maar voor deze registers primair. In dat geval zal de kwaliteit in de aanleverende registers beter zijn.
- Introduceer de activiteiten vroeg bij andere onderdelen van de organisatie. Het is van belang dat de projecten gebruik gaan maken van de gegevens in het register, zodat hergebruik gestimuleerd wordt. Door al in een vroeg stadium informatie te geven over het eindresultaat wordt de bekendheid van het product vergroot en wordt de introductie in een later stadium vereenvoudigd.

- Betrek belangrijke stakeholders bij het ontwerp. Omdat het ontwerptraject gebaseerd is op interactieve sessies wordt het mogelijk om de stakeholders te betrekken bij het ontwerptraject en zodoende kunnen zij bijdragen aan het eindresultaat. Gevolg is dat de direct betrokkenen als ambassadeur van het register gaan fungeren.
- Houdt rekening met problemen in de financiering. Introduceren van een koppelingenregister kost geld terwijl het directe nut in een bezuinigende overheid gauw uit het zicht verdwijnt. In dit project is hierdoor een behoorlijke vertraging ontstaan. In eerste instantie is gekozen voor financiering uit generieke middelen, daarna is gezocht naar een subsidiërend project. Beiden zijn mislukt, oplossing is gevonden door het ambitieniveau naar beneden bij te stellen en te zoeken naar een uitwerking door een HTS stagiaire met in eerste instantie een interne applicatie.
- De werkwijze met een prototype dat bijgesteld wordt op basis van interactieve workshops biedt mogelijkheden om op goedkope en efficiënte wijze tot een resultaat te komen. Gezien de problemen met de financiering heeft deze werkwijze bijgedragen aan het bereiken van het uiteindelijke resultaat.
- Ontwikkel voorlichtingsmateriaal bij de introductie. In dit project is een factsheet ontwikkeld een document van twee A4tjes waarin uitgelegd wordt wat het koppelingenregister is en hoe en wanneer het gebruikt zal worden. Dit wordt als Pdf naar technisch projectleiders gestuurd ter informatie. Hiermee wordt de bekendheid van het register op goedkope manier vergroot.

## Koppelingenregister webapplicatie

In dit artikel is het prototype van het koppelingregister reeds aan de orde gekomen. Het uiteindelijke prototype is gepubliceerd op internet en kan door iedereen bekeken worden. Het geeft een beeld van het eindproduct van de prototyping fase binnen het hierboven beschreven project. Dit is te vinden op <http://www.serviceregister.dla-os.nl>. U kunt inloggen met gebruikersnaam developer en wachtwoord developer om de inhoud te kunnen muteren.

## Literatuur

Erl, T., *SOA, Principles of service design*, Prentice Hall International, Englewood Cliffs NJ, 2008.

Rumbaugh, J. et al, *Object-Oriented Modeling and Design*, Prentice Hall International, Englewood Cliffs NJ, 1991.

Snoeck, M. et al, *Object-Oriented Enterprise Modelling with MERODE*, Leuven University Press, Leuven BE, 1999.

## Over de auteur

Bert is onafhankelijk software architect voornamelijk werkzaam in de zorg en bij overheidsinstellingen. Zijn werkzaamheden liggen veelal op het raakvlak van software ontwikkeling en werkproces modellering. Bert heeft al vanaf 1990 een passie voor object oriëntatie, dictionaires, software architectuur en het genereren van software. Informatie over deze onderwerpen in de vorm van artikelen, tools en een weblog vindt u op [www.dla-os.nl](http://www.dla-os.nl)